PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

A UNIFIED EDUCATIONAL RESOURCE SEARCH ENGINE WITH INTELLIGENT RANKING FOR SMART EDUCATION

A PROJECT REPORT

Submitted by

NITHIN M S- 20221ISE0066

ADITYA V S – 20221ISE0032

CHIRANTHAN T C NADIG- 20221ISE0062

Under the guidance of,

Mr. JOHN BENNET JOHNSON

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2025

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013
Itgalpura, Rajankunte, Yelahanka, Bengaluru – 560064

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this report "A Unified Educational Resource Search Engine with Intelligent Ranking for Smart Education" is a Bonafide work of "Nithin M S (20221ISE0066), Aditya V S (20221ISE0032), Chiranthan T C Nadig (20221ISE0062)", who have successfully carried out the project work and submitted the report for partial fulfilment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in INFORMATION SCIENCE AND ENGINEERING, during 2025-26.

[illegible] 3/12

Mr. John Bennet J
Project Guide
PSCS
Presidency University

[illegible]

Ms. Suma N G
Program Project Coordinator
PSCS
Presidency University

[illegible] 3/12/25

Dr. Sampath A K
Dr. Geetha A
School Project Coordinators
PSCS
Presidency University

[illegible] 03/12/25

Dr. Zafar Ali Khan
Head of the Depart
PSCS
Presidency University

[illegible]

Dr. Shakkeera L
Associate Dean
PSCS
Presidency University

[illegible]

Dr. Duraipandian N
Dean
PSCS & PSIS
Presidency University

Examiners

Sl. no.	Name	Signature	Date
1	Deepthi. J	[illegible]	3/12/25
2	JAI KUMAR. B	[illegible]	3/12/25

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of final year B.Tech in INFORMATION SCIENCE AND ENGINEERING, at Presidency University, Bengaluru, named Nithin M S, Aditya V S, Chiranthan T C Nadig, hereby declare that the project work titled "A Unified Educational Resource Search Engine with Intelligent Ranking for Smart Education" has been independently carried out by us and submitted in partial fulfilment for the award of the degree of B.Tech in INFORMATION SCIENCE AND ENGINEERING, during the academic year of 2025-26. Further, the matter embodied in the project has not been submitted previously by anybody for the award of any Degree or Diploma to any other institution.

Nithin M S	USN: 20221ISE0066	[illegible]
Aditya V S	USN: 20221ISE0032	Aditya V.S
Chiranthan T C Nadig	USN: 20221ISE0062	[illegible]

PLACE: BENGALURU

DATE: 03/12/2025

ACKNOWLEDGEMENT

For completing this project work, we have received the support and the guidance from many people whom I would like to mention with deep sense of gratitude and indebtedness. We extend our gratitude to our beloved **Chancellor, Pro-Vice Chancellor, and Registrar** for their support and encouragement in completion of the project.
I would like to sincerely thank my internal guide **Mr. John Bennet J, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University, for his moral support, motivation, timely guidance and encouragement provided to us during the period of our project work.

I am also thankful to **Dr. Zafar Ali Khan, Professor, Head of the Department, Presidency School of Computer Science and Engineering** Presidency University, for his mentorship and encouragement.

We express our cordial thanks to **Dr. Duraipandian N**, Dean PSCS & PSIS, **Dr. Shakkeera L**, Associate Dean, Presidency School of computer Science and Engineering and the Management of Presidency University for providing the required facilities and intellectually stimulating environment that aided in the completion of my project work.

We are grateful to **Dr. Sampath A K, and Dr. Geetha A, PSCS** Project Coordinators**, Ms. Suma N G, Program Project Coordinator**, Presidency School of Computer Science and Engineering, or facilitating problem statements, coordinating reviews, monitoring progress, and providing their valuable support and guidance.

We are also grateful to Teaching and Non-Teaching staff of Presidency School of Computer Science and Engineering and also staff from other departments who have extended their valuable help and cooperation.

NITHIN M S

ADITYA V S

CHIRANTHAN T C NADIG

ABSTRACT

The growth of the digital learning methods and open access educational resources has changed the information retrieval methods that the learners, researchers, developers use to obtain the required information from various platforms. YouTube, GitHub, arXiv and Kaggle are the various important sources for finding video tutorials, open-source code, research papers and datasets. However, even with the existence of all these resources, the learning process is still lacking. It still requires the user to independently browse the information from different sites, glance through the relevant and irrelevant information and decide resources are trustworthy and appropriate for their needs. The absence of a unified resource system is lacking in efficient learning and discovery of knowledge.

The project titled **"A Unified Educational Resource Search Engine with Intelligent Ranking For Smart Education"** aims to propose complete unified full stack federated search and recommendation system to solve the issue of dispersed resources. The proposed system includes frontend and backend designed using HTML5, CSS3, JavaScript, PHP, MySQL and it is hosted locally on XAMPP. The system allows users to register, sign-in and search for the required educational materials. It also provides a personalized dashboard for its users where the user will have five different categories such as videos for YouTube, code repositories from GitHub, research papers for arXiv, datasets from Kaggle. The system accesses platform specific APIs with real time data and provides the ranked results based on the stars, likes, recency and their usability. It uses a weighted score function that ensures that the retrieved contents are relevant and high-quality resources are provided with their ranks while the results are displayed.

The backend systems use the user search history and user interaction with the search results and offers a personalized recommendations to the users, stored in MySQL database. The system uses content-based filtering and frequency analysis on the user queries to offer recommendations that matches with the user's interest. The feedback from is stored in the backend to assist the system improvement.

This integration of advanced learning technology solves several important problems such as resource fragmentation, personalization and ineffective retrieval system. The use of unified search engine with the ranking system and recommendation system allows the user to engage with the wider variety of educational resources at a single platform for seamless education.

Table of Content

List of Figures

List of Tables

Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
GDPR	General Data Protection Regulation
HTML	Hyper Text Markup Language
HTTPS	Hyper Text Transfer Protocol Secure
ID	Identifier
IR	Information Retrieval
NLP	Natural Language Processing
OA	Open Access
OJ	Online Judge
PHP	Hypertext Preprocessor
RDF	Resource Description Framework
SDG	Sustainable Development Goals
SEO	Search Engine Optimization
SQL	Structured Query Language
TF-IDF	Term Frequency – Inverse Document Frequency
UI	User Interface
URL	Uniform Resource Locator

Chapter 1
Introduction

1.1 Background of the Project

Digital education helps the learners, educators and researchers with a new and flexible way to access the required information. Over the past ten years, resources like YouTube, GitHub, arXiv and Kaggle have become the most commonly used resources for educational purposes. YouTube provides educational videos and tutorials, GitHub and Kaggle have collaborative projects and the datasets that are required for solving analytical and machine-learning challenges and arXiv has research papers that are required for the analysis for developing a project. Each platform provides accessibility, but in isolation. A learner who is interested in machine learning domain has to check for the required resources in each platform separately YouTube for video tutorials, GitHub for the project repositories, arXiv for the research papers and Kaggle for datasets. The gap in the process of finding the educational resources is a major challenge for the learners and it is time consuming. Moreover, Bing and Google which are used as search engine for these resources only prioritize popularity while returning the search results and the overlook the educational value.

Given these obstacles, the project aims to provide "Study Search: A Unified Educational Resource Platform" designed to be a federated search and recommendation system that consolidates disparate educational materials into a single smart interface. The system crawls different platforms, retrieves, and ranks educational materials, and actively learns from users to provide recommendations. The updated version, unlike the earlier client-side prototype, incorporates both frontend and backend modules. It allows secure user login, tailored recommendations and feedback, and administrator analytics, all from within the system built with XAMPP, PHP, and MySQL.

1.2 Statistics of the Project

The demand for a unified learning platform is justified by the growth of the global educational contents, as shown by the following statistics for different categories:

- YouTube's educational content related videos that are being uploaded exceed 500 hours every minute (Google Trends,2024).

- GitHub has over 100 million repositories where many are educational.
- arXiv uploads almost 16,000 new research papers every month which covers multiple fields such as artificial intelligence and physics.
- Kaggle has more than 50,000 public datasets for research purpose and analytics.

Study Search Pilot Results:

The average time saved per query: ~35-40% compared to the traditional information retrieval methods.

- User satisfaction rating based on the retrieved contents in pilot testing: 4.6/5 for providing the relevant results and the clear results.
- Database evaluation (local MySQL log entries): 300+ unique searches based on the required topic, 120+ registered user accounts, 50+ feedback entries received by the users during the evaluation of the database entries.

These statistics of the user rating and the database evaluation clearly justifies the need for a unified search engine that can address educational queries related to multiple platforms efficiently and provide a personalized experience to the users

1.3 Prior Existing Technologies.

There are many existing technologies that support the in process of finding educational resources, but they still remain the same and limited in integrating the resources and giving personalized content:

1. **Traditional Search Engines like Google, Bing.**

- Offers are broad categories of resources or search results but this searching is done based on the key word identified from the user query.
- They prioritize popularity and advertisement metrics rather than the academic values.
- They do not provide categorically specified results such as videos/code/papers/datasets.

2. **Domain-Specific Platforms like YouTube, GitHub, arXiv and Kaggle.**

- Each platform has its own database and API key, which performs independently.
- Users must browse on these different platforms separately and the manually aggregate the search results.

3. Semantic Search Engines and Recommendation Engines

- They use natural language processing to enhance the search query and provide relevant resources.
- Semantic Search engines are only restricted to only one type of resource format.
- These engines lack the idea of unified scoring across the different categories of resources.

4. Adaptive Learning Systems and MOOCs like Coursera, edX, Udemy

- They provide structed courses in different domain but not an open resource discovery.
- These systems have a closed ecosystems with only linited external API integration in information retrieval.

The limitations of these systems are the fragmented architecture of the system, the absence of the personalization of the resources and they have a weak cross platform compatibility. These limitations highlight the need for developing a unified resource platform such as the study search engine.

1.4 Proposed Approach

The proposed of this project combines the aspects like federated search, content-based recommendation based on the search history and a feedback driven improvement method under a full-stack framework.

- **Frontend Layer of the Project (HTML, CSS, JavaScript):**

The front end of the project provides an interactive interface where the users can create their accounts by registering and access the same by logging in. Then search for the required content and view the results in different categories like videos, code, datasets, papers. The smooth animations and the skeleton loaders enhance the responsive design of the frontend.

- **Backend Layer of the Project (PHP + MySQL):**

The Backend layer handles the authentication, search query processing and provides persistent data storage. Factors such as search query, user history and the feedback are analyzed for providing personalized recommendations.

- **API Integrations:**

Each external platform such as YouTube, Kaggle, GitHub and arXiv is queried separately using its public API to obtain the results. These retrieved responses are normalized into a unified schema and these results are ranked based on the resource specific algorithms.

- **Recommendation Module for the Project:**

This analyzes the search history of the user to identify the user's interest. Then the system generates the recommendations based on this search history.

- **Admin Dashboard:**

This enables the system administrator to monitor the user activity, review the feedback and make changes accordingly for the future updates of system.

This architecture of the proposed system ensures scalability, modularity and adaptability to future integration of the system with the machine-learning-based semantic models.

1.5 Objectives.

The main objectives of the project are as follows:

1.Unify All the Educational Resources:

To provide a single platform to retrieve code, videos, datasets and research papers at a single platform,

2.Develop a Full-Stack Functionality:

Implement the process of secure user login / registration, storage options at the database and the admin control panel using PHP and MySQL within XAMPP.

3.Designing of Personalized Recommendation System:

Generating intelligent recommendations based on user's previous search history and the interaction with the contents.

4.Implementation of Efficient Ranking Mechanisms:

Rank the resources using quality-specific measures such as likes, stars, citations and downloads.

5.Providing Feedback and Analytics:

Enabling the users to submit feedback and allowing the administrators to visualize user's insight using the backend dashboards.

6.Ensuring the User-Centric Experiences:

Creating a responsive and visually appreciative interface suitable for learners and educators of all backgrounds.

1.6 Alignment with the Sustainable Development Goals (SDGs).

Numerous United Nations Sustainable Development Goals (SDGs) are directly impacted by the project

Table 1.1 SDG Goal Mapping

SDG Goal	Description	Project Contribution
SDG 4 – Quality Education	Make sure that all students receive high-quality education that is inclusive and equitable, and encourage opportunities for lifelong learning.	Offers well-organized, easily accessible, and cost-free learning materials across several platforms.
SDG 9 – Industry, Innovation and Infrastructure	Encourage innovation, advance sustainable industrialization, and construct robust infrastructure.	Encourages open-source innovation by facilitating cross-disciplinary access and GitHub integration.
SDG 10 – Reduced Inequalities	Minimize disparities in access to technology and education.	Permits students to access global resources without any restrictions on subscriptions.
SDG 12 – Responsible Consumption and Production	Make sure patterns of production and consumption are sustainable.	Reduces the need for printed materials while promoting the use of digital learning.

By bridging the gap educational aspects and democratizing the access to quality-based content, **Study Search** aligns technology with sustainability, inclusivity and innovation.

1.7 Overview of the Project Report

This report is structured into nine comprehensive chapters, each detailing a specific component of the project lifecycle:

- **Chapter 1** – Introduction: Provides background, motivation, objectives of the project and the project overview.
- **Chapter 2** – Literature Review: Highlights the existing studies, methods and technologies related to federated search, recommendation systems, and educational resource discovery.
- **Chapter 3** – Methodology: Describes the design approach, workflow, and architecture of the system.
- **Chapter 4** – Project Timeline: Presents development milestones and review stages.
- **Chapter 5** – Analysis and Design: Details the system analysis, requirement specifications, and design diagrams.
- **Chapter 6** – Software and Implementation: Explains the front-end, back-end, and database modules with code structure and integration details.
- **Chapter 7** – Evaluation: Provides testing results, performance metrics, and usability assessments.
- **Chapter 8** – Social, Legal, Ethical, Sustainability, and Safety Aspects: Discusses compliance and societal implications of the project.
- **Chapter 9** – Conclusion and Future Work: Summary of the key findings, conclusions of the project and the future work that has been planned.

Every section explains how the evolution of the **Study Search Engine** from ideas to the implementation as a unified educational resource platform supporting the education of the learners, developers and researchers with personalized, secure and sustainable learning with ranked results.

Chapter 2

Literature Review

2.1 Information Retrieval: Recent Advances and Beyond

[1] K. A. Hambarde H. Proença 2023. Hambarde Proença document a current survey of information retrieval (IR) advancements including semantic embeddings, neural ranking models, and distributed retrieval frameworks. They distil trends demonstrating the distinct retrieval architectures and methods neural models plus embedding based semantics matching retrieval. The survey highlights important deficits resulting from domain specific quality ranking and multi-modal multi-source educational retrieval lacking standardized benchmarks. Constructing new benchmarks and coupling pedagogical evaluative metrics to retrieval systems is a priority. This supports the methodological reasoning semantic embeddings federated search for retrieval frame the search and evaluation pedagogy suggest new. This survey highlights evaluation issues that we need to cover during the experiment.

2.2 A Semantic Approach to Ranking Techniques: Improving Web Page Searches for Educational Purposes

[2] C. Limongelli et al. (2022). Limongelli et al. discuss outlines of ranking web pages for educational purposes using web pages semantic features coupled with relevance criteria defined by educators and focus on boosting instructional materials within search results. Their empirical assessment reflects better alignment with instructional objectives and pedagogical tiers (i.e., transparency and depth of concepts) as opposed to standard search results. They recommend future research on automated extraction of pedagogical features and broadened learning outcome evaluation. Some of the major concerns are the curated pedagogical criteria and focus on HTML/web page content; the developed method does not cover non-web page formats for instance repositories of code and other datasets. In the case of our project, Limongelli's primary focus on pedagogically relevant content supports the incorporation of domain-specific quality signals but also highlights the requirement for wide-scale automated feature filler extraction.

2.3 Automatic Adaptation of Open Educational Resources: An Approach From a Multilevel Methodology Based on Students' Preferences, Educational Special Needs, Artificial Intelligence and Accessibility Metadata.

[3] P.Ingavélez-Guerra et al. (2022) describe the intelligent adaptation framework proposed by Ingavélez-Guerra and members of the group that recommend the use of multi- tiered artificial intelligence approaches to completely adjust open educational resources (OERs) to each of the learner's needs, preferences, and accessibility requirements. Their research showcases the adaptation of content by employing metadata enrichment, learner modeling, and accessibility annotation. Results from the pilot studies show an increase in learner accessibility and satisfaction. They are, however, in favor of the addition of centering the learner feedback loops and more refined accessibility semantics for expanded reach. The approaches described in the literature seek to improve learner satisfaction.

2.4 Efficient Retrieval of Data Using Semantic Search Engine Based on NLP and RDF

[4] U. Yadav and N. Duhan (2021). Yadav and Duhan had presented a semantic retrieval approach that used NLP methods on RDF for the representation of educational structured documents to help query understanding and increase the precision of the information retrieval. The method maps queries and document metadata to semantic representations, and uses RDF relations to produce a ranked list of semantically relevant concepts in addition to lexical matching, resulting in improved precision and recall scores on RDF-backed test collections relative to a baseline keyword search. The findings confirm that explicit semantic structure supports disambiguating the user's informational needs. The authors suggest a potential extension to model larger and more heterogeneous datasets for educational resources and integrate domain-adaptive representations by including richer ontologies. The main limitations they acknowledge is the reliance on existing RDF annotations, which do not exist most multimedia and unstructured content, and reduced to purely unstructured sources such as video transcripts or code files. Scalability issues were also mentioned in converting raw data from the web into RDF. The potential to adapt similar semantic embeddings is an inspiration for our project but the paper also suggests that we have to address the challenge of unstructured and multimedia content without such handcrafted RDF graphs.

2.5 An Intelligent Video Tag Recommendation Method for Improving Video Popularity in Mobile

[5] R. Zhou et al. (2020). Zhou and coauthors develop an intelligent video tag recommendation system aimed at improving video discoverability and popularity in mobile environments. Their system analyzes video metadata, visual and audio features, and viewer interaction data to propose tags that better reflect content and user intent, which in turn increases visibility in platform search and recommendation pipelines. Evaluations show measurable increases in click-through and watch times after applying the tag recommendations. The authors recommend combining automated tagging with manual curation for higher-quality labels and exploring personalization of tags based on viewer segments. The primary weakness is the single-resource focus techniques rely on video- specific signals and do not translate directly to code or papers. For Quick Search, this paper suggests valuable feature extraction methods for video ranking, but it also points to the necessity of designing resource-specific extractors while maintaining a unified ranking strategy across types.

2.6 From Code to Ratings – Enhancing Collaborative Filtering in OJ Systems

[6] D. Muepu & Y. Watanobe (2019). Muepu and Watanobe use the data from a participant's programming activities and convert it to implicit ratings that can enhance collaborative-filtering recommendation systems. These systems typically recommend problems to solve in online judges (OJs). The procedure takes in logs of submissions and executions, as well as the time to solve, in order to produce a user-item affinity score. The user-item affinity scores are used to generate recommendations of problems for the user to attempt. The results show that the recommendations were more relevant and the learners were more engaged compared to the baseline settings. This was particularly the case for novice users who have a beneficial experience with targeting problem difficulty. This paper recommends enriching rating signals with contextual features such as code style and hint usage to improve personalization. Key limitations are the heavy coupling to OJ platforms and the domain-specificity of derived features; approaches and features do not generalize to resources such as research papers, videos, or datasets. The study shows how signals can be incorporated into recommendation systems. It also shows how the project's multi-metric ranking of resource pieces (which intersects popularity and semantics) can be achieved. There is therefore a lesson about feature designs for generalization across resource types.

2.7 Using Research Literature to Generate Datasets of Implicit Feedback (LIBRETTI)

[7] D. Spinellis et al. (2019). Spinellis and colleagues present a method to extract implicit feedback datasets from research literature to help recommend science articles The author of the paper mine citations, co-authorship networks and usage to create datasets reflecting how researchers implicitly endorse or relate to work. They subsequently use these to train and evaluate recommendation models. The outcome of our experimentation suggests that systems trained to rely on data derived from implicit feedback recommend papers of high topical relevance than simple keyword-based baselines. The authors suggest that providing richer feedback sources (e.g., citations, references, full-text similarity) would do better to capture the scholarly relevance of OA content. Furthermore, they propose that future attempts should perhaps try to include altmetrics and usage logs. The weaknesses targets are limited to text-based scholarly artifacts and limited handling of multimedia or code artifacts and quality implicit feedback generation requires access to a large bibliographic corpus. The contribution is useful for the "papers" component of our platform but highlights the need to merge different feedback and quality signals in ranking heterogeneous resources.

2.8 Enhanced Search Engine Using Proposed Framework and Ranking Algorithm Based on Semantic Relations

[8] M M El-Gayar & Co. (2019) authors a paper on a semantic-relationship-based ranking improvement search framework. The authors develop relation-aware similarity metrics (beyond lexical matching) to reclassify search results on a conceptual basis and empirically demonstrate enhanced ranking proficient outcomes on educational web pages. They demonstrate that semantic link inclusion quietly eliminates irrelevant results and target underspecified but contextually relevant, pedagogically valuable, associations. The ontological domain is and interaction-driven semantic-relation ranking refinement is recommended, and frame semantic relations with relevant ontologies. Relation-extraction, especially domain-focused, is a pertinent limitation, alongside the underspecified code and multimedia experimentation. Approaches massive domain knowledge, preprocessing, and scrutiny. Our engineering draws on their principles but also exposes the difficulty of designing uni-purpose relation extractors over diverse datasets.

2.9 GENAUM: NEW SEMANTIC DISTRIBUTED SEARCH ENGINE

[9] I. Saif et al. With their work, Saif et al. developed GENAUM, a semantic distributed search engine that aims to amalgamate search results from multiple different sources. Their systems demonstrate how queries may be sent to different indexes and how the results can be semantically merged into single lists. Experiments suggest the concept has merit but show lags and discrepancies in source scoring and format variances. This paper explained investigations by which large amount of information can be summarized and also explained the principles behind heuristic data amalgamation. This work is important for Quick Search in that it serves both to showcase a federated architecture and highlight the real-world problems such as API rate quotas, schema mapping, and aggregate score that our work has to address.

2.10 How to Use Search Engine Optimization Techniques to Increase Website Visibility

[10] J. Killoran (2013). He looks into the impact of various Search Engine Optimization (SEO) techniques and what link structure and keyword placement signals mean for SEO visibility on generic search engines. He meticulously looks into ranking heuristics that prioritize (or have an abundance of) popular and linked content, regardless of the educational merit. He highlights that the popular geo-centric approach could easily disregard informative and high-quality educational materials. The suggestions recommending the SEO methods for content popularity to SEO visibility on the content quality proxies to refine the mechanistic... The boundaries set for educational search and retrieval are obvious, and in this case SEO signals are the best proxies for pedagogical values. This motivates our hybrid approach that combines popularity metrics with, but does not rely solely on, semantic relevance.

2.11 Literature survey of all the papers

Table 2.1 Literature Survey of all Papers

Author(s) & Year	Title / Focus	Key Contributions	Limitations
Hambarde & Proença (2023)	Information Retrieval:Recent Advances and Beyond	Overview of advances in information retrieval techniques.	General survey; no direct implementation for smart education.
Limongelli et al. (2022)	A Semantic Approach to Ranking Techniques	Improved web page searches for educational purposes using semantic raking	Limited to web content; no federated or cross-domain support.
Ingavélez-Guerra et al. (2022)	Automatic Adaptation of Open Educational Resources	Introduced AI-based adaptation of open educational resources based on user preferences and accessibility needs.	More focused on personalization, not federated search or integration.
Usha Yadav & Neelam Duhan (2021)	Efficient Retrieval of Data Using Semantic Search Engine Based on NLP and RDF	Proposed a semantic search engine using Natural Language Processing and RDF for better query understanding.	Focused mainly on structured RDF data; lacks cross-platform scalability.
Renjie Zhou et al. (2020)	Intelligent Video Tag Recommendation	Developed a method for recommending video tags to improve video visibility and popularity.	Focused only on video metadata; lacks semantic depth across platforms.
Daniel M. Muepu & Yutaka Watanobe (2019)	From Code to Ratings – Enhancing Collaborative Filtering in OJ Systems	Converted programming data into ratings to improve recommendation systems.	Domain-specific to online judge systems; not applicable to multi-source search.
Diomidis Spinellis et al. (2019)	Using Research Literature to Generate	Built feedback from scientific literature	Limited to research articles; ignores videos, code, and datasets.

	Datasets of Implicit Feedback	to recommend papers.	
El-Gayar et al. (2019)	Enhanced Search Engine Using Framework & Semantic Ranking	Proposed a framework with semantic relations for improving ranking algorithms.	Does not address integration of heterogeneous sources.
Saif et al. (2017)	GENAUM: New Semantic Distributed Search Engine	Proposed a semantic distributed search engine to combine multiple sources.	Limited scalability and did not fully cover diverse educational resources.
Killoran (2013)	Search Engine Optimization Techniques	Discussed SEO methods for improving website visibility.	SEO-based ranking not aligned with academic/educational relevance.

Chapter 3

Methodology

The main idea of this project is to design and implements a federated search platform that will enable the searching process through a variety of educational resources from a single platform. This process starts from the user interface module where the users start by logging in or registering to their account and then access the search bar where they can search for the required content and tabbed functions for code, videos, research papers and datasets. When a user searches by giving a, the input control modules send the query to the relevant API integration modules of YouTube, Kaggle, GitHub and arXiv based of the selection of the user. With every API call, the raw data is received and this raw data is normalized in to common schema.

The project's ranking algorithm is implemented based on the relevance and other quality measures like recency.

- The YouTube videos are ranked based on their views, likes and date of upload.
- GitHub repositories are ranked based on stars, forks and recent activity.
- The arXiv research papers are ranked based on their recency and citation count.
- Kaggle datasets are ranked ranks by how useful and popular they are.

Finally, the weighted formula of the system combines the above measures for each platform to provide a final ranked results that are relevant and trustworthy.

To enhance the experience of the users, "skeleton loader module" is added to the system that displays the contents in the form of the animated placeholders while the search results are loading. This interactive design of the project allows for the growth of the system, also finds a possible way for the other platforms such as YouTube, Kaggle, GitHub and arXiv to add ranking capability later. This process of providing the ranked results aims to help learners extract relevant results, and makes the information accessible from different platforms at a single place by addressing the problem of fragmented results, which was found while using traditional information retrieval methods.

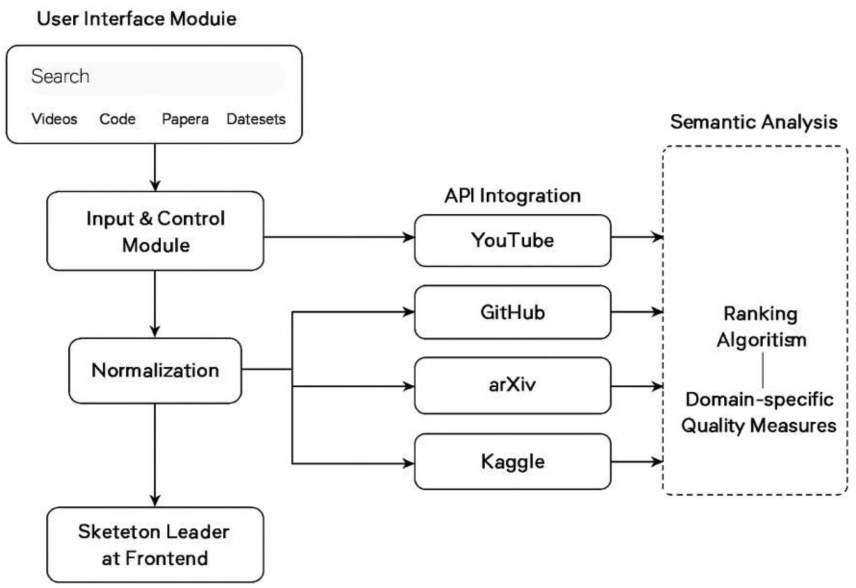


Fig 3.1 Methodology

- The above block diagram of the system explains the overall working of the proposed system.
- The User Interface Module allows the users to search for required educational resources such as videos, code repositories, research papers, and datasets by providing a search bar that has different categories like code, videos, datasets and papers. The query entered by the user is handled by the Input and Control Module, which directs the query entered to the relevant component for the processing.
- The Normalization Module standardizes the query entered by the user to maintain uniformity across different platforms. The query is then passed to the API Integration Layer, where the connection with external sources like YouTube, GitHub, arXiv, and Kaggle takes place to find the relevant results for the query.
- Fetched data is then processed through Semantic Analysis, which has a Ranking Algorithm and Domain-specific Quality Measure to refine the fetched data and to ensure that the most relevant and high-quality results are displayed to the user. This refined data is finally displayed to the user through the Skeleton Loader at the Frontend, ensuring a responsive UX.

Chapter-4

Project Management

4.1 Project Timeline.

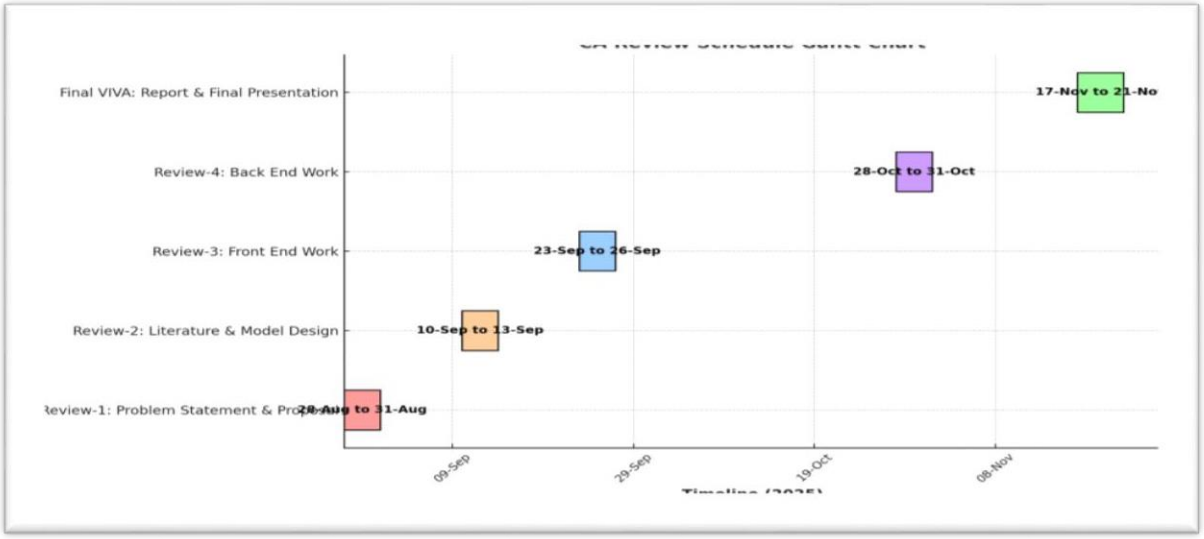


Fig 4.1 Project Timeline

- **Review 1:** Problem Identification and Title Confirmation.
- **Review 2:** Literature Survey and Data Collection.
- **Review 3**: Demonstrated the 50% working of the project where the front-end design was completed.
- **Review 4:** Demonstrated the 100% working of the project with all working functionalities of the frontend and the backend.
- **Final Review:** Viva, Report Submission and Research Paper publication.

4.2 Risk Analysis.

- There was a risk of incorrect ranking results if the algorithm contains logical errors. To overcome this: The ranking formula was tested with multiple datasets and validated for accuracy.

- The ranking process may slow down when handling large amounts of data. To overcome this: SQL queries were optimized, indexing was used, and unnecessary calculations were minimized.

- Incomplete or inaccurate input data can lead to unreliable ranking outputs. To overcome this: data validation checks were added, mandatory fields were enforced, and fallback values were applied.

- The ranking algorithm may unintentionally introduce bias and favour certain results. To overcome this: the logic was reviewed, normalized scoring was used, and bias testing was performed during development.

Chapter 5

Analysis and Design

5.1 Problem Analysis

The conventional method of educative content discovery is disjointed and ineffective. Educators, researchers, and students are forced to go to various sites including YouTube. Finding videos, open-source projects, academic papers, and GitHub, arXiv and Kaggle. datasets respectively. This is a multi-platform specific, multi-step process that results in:

- **Redundancy:** The search of the same topic in more than one site.
- **Inefficiency:** Wastage of time in navigation, filtering and comparison of results.
- **Absence of Personalization:** Uncustomed search results that fail to conform to the interests of the users or learning habits.
- **Unorganized Experience:** The users experience untidy interfaces, metadata types, and ranking criteria.
- **Poor Feedback Support:** There is no inbuilt mechanism of user rating or commenting on the learning materials retrieved.

Also, although there are current recommendation systems in the entertainment industry (e.g., YouTube, Netflix), not a lot of systems are as personalized towards educational search contexts. It is with this gap that a federated educational search engine should be promoted, which combines with all the significant learning platforms personalization, ranking and feedback mechanisms.

Key Issues Identified:

- Disjointed discovery of resources.
- Lack of customized, integrated academic search.

- None of the cross-platform ranking.
- Minimal interaction of user feedback in learning tools.

The proposed Study Search Engine can overcome these problems by providing a full-stack, integrated, smart, and demonstrative resolution to access international learning materials.

5.2 Requirement Analysis

The system requirements were classified into functional and non-functional categories.

- **Functional Requirements**

Table 5.1 Functional Requirements

ID	Requirement	Description
FR1	User Registration and Login	Users can register, log in securely using PHP– MySQL authentication.
FR2	Search Functionality	Search educational resources across YouTube, GitHub, arXiv, and Kaggle.
FR3	Result Ranking	Rank results by relevance metrics like likes, stars, citations, and downloads.
FR4	Personalized Recommendations	Suggest resources based on user search history.
FR5	Feedback System	Allow users to submit feedback stored in backend database.
FR6	Admin Dashboard	Admin can view feedback, users, and search logs.
FR7	Error Handling and Data Validation	System handles invalid inputs and API errors gracefully.
FR8	Session Management	Maintain secure login sessions until logout.

- **Non-Functional Requirements**

Table 5.2 Non- Functional Requirements.

Parameter	**Requirement Description**
Performance	Fast response time (<3s) for API retrievals and database queries.
Scalability	Architecture supports addition of new APIs or content platforms.
Security	Encrypted password storage (password_hash()), secure session management.
Usability	Responsive layout across devices, intuitive navigation.
Reliability	Fault-tolerant behavior for API failures; error messages displayed.
Maintainability	Modular code with separate files for frontend, backend, and configuration.
Portability	Fully compatible with XAMPP (Windows/Linux) and adaptable to cloud hosting.

5.3 System Design

The Study Search design is designed in a manner that is modular, scaled and effective in communication between the frontend and backend as well as the database layers.

5.3.1 High-Level Architecture

- **Frontend Layer (User Interface):**

The user interface, or frontend layer, is constructed with HTML, CSS, and JavaScript. Search bars, resource category tabs, skeleton loaders for a dynamic experience, and a responsive layout are all included.

- **Application Layer (Backend):**

PHP scripts manage sessions, process data normalization, respond to API requests, and store user activity in MySQL. Additionally, it manages the recommendation and ranking logic for a better content retrieval.

- **Database Layer:**

This layer is responsible for storing the user feedback, search history and the user data such as details, user-id and password that are stored in a MySQL database. It is helpful for generate recommendations for every user based on their stored activity and to preserve their sensitive data such as passwords, details.

- **External APIs:**

External sources of data include the data retrieved from YouTube, GitHub, arXiv, and Kaggle. The responses from different platform are retrieved and formatted into a single output by the backend.

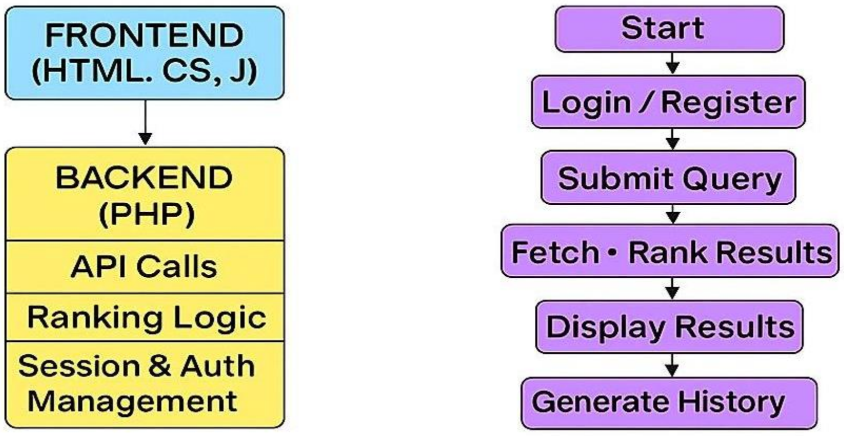


Fig 5.1 System Architecture Diagram

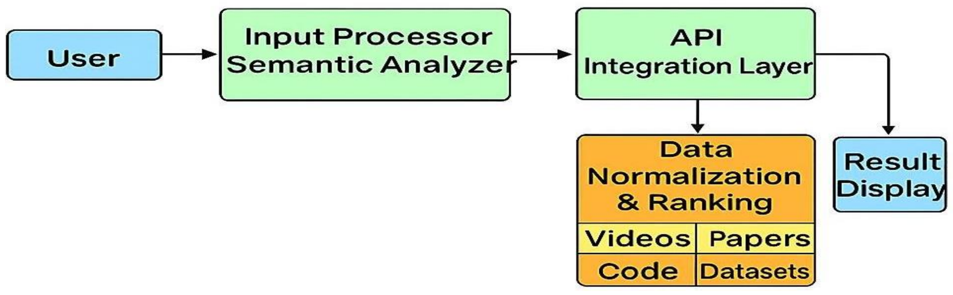


Fig 5.2 Block Diagram of Proposed System

5.3.2 Detailed Design

A. User Interface Design

- The homepage features a search bar and tabs for categories (Videos, Code, Papers, Datasets).
- PHP connects the login/registration page to MySQL.
- Result display cards that include a resource link, title, and thumbnail.
- Below the search results is a section with recommendations.
- A feedback form to gather user opinions.

B. Data Flow Design (DFD)

- User submits a query.
- Backend routes query to API functions.
- Results normalized, ranked and displayed
- User search stored in history table.
- Recommendations generated and displayed.
- Module Interaction.

Table 5.3 Module Interaction

Module	**Functionality**
Authentication	Handles registration, login, and session control.
Search Engine	Calls APIs and fetches data.
Ranking Engine	Scores and sorts results.
Recommendation Engine	Suggests content from user history.

Feedback Module	Captures and stores user comments.
Admin Panel	Monitors system activity and manages data.

5.3.3 Ranking Algorithms

Table 5.4 Ranking Algorithm

Platform	Ranking Formula	Description
YouTube	Score = 0.6 × Views + 0.4 × Likes	Prioritizes popular and engaging educational videos.
GitHub	Score = 0.7 × Stars + 0.3 × Forks	Focuses on well-maintained and useful repositories.
arXiv	Score = 1 / (CurrentYear - YearPublished + 1)	Prefers recent research for up-to- date learning.
Kaggle	Score = 0.5 × Downloads + 0.5 × (Usability × 1000)	Ensures quality and reliability of datasets.

5.3.4 Tools and Technologies

Table 5.5 Tools and Technologies

Category	Tool / Framework	Purpose
Front-End	HTML5, CSS3, JavaScript	Interface design, layout, responsivenes
Back-End	PHP (v8.x)	Data handling, API calls, session control.
Database	MySQL (via XAMPP)	Data storage and retrieval
APIs	YouTube, GitHub, arXiv, Kaggle	Resource retrieval
Server	XAMPP (Apache + MySQL)	Local testing and hosting
IDE	Visual Studio Code	Code development and debugging

Version Control	GitHub	Collaboration and code management
Deployment	GitHub Pages(frontend) /XAMPP localhost	Hosting environment

5.4 System Flow Diagram

The flow diagram of the system can by represented by the following diagram:

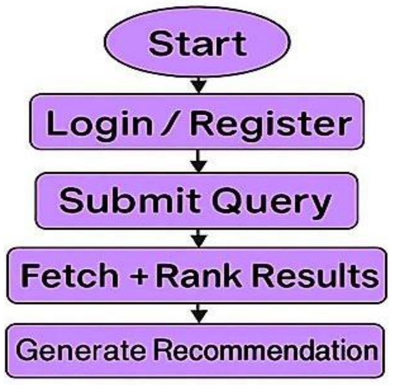


Fig 5.3 System Flow and User Interaction

5.5 Summary

The deployment of the full-stack architecture’s foundation was done during the analysis and design stage of this system. Secured authentication through user-id and passwords, accurate federated search retrieval by giving different options or categories of contents, intelligent ranking where the results are ranked, personalized recommendations for each user, and robust data management via MySQL in the backend are all made possible by the system's modular design structure. Study Search guarantees to be scalable, maintainable, and readily expandable with future cloud-based deployment or AI-driven recommendation features thanks to this architecture of the system.

Chapter 6

Software Implementation

6.1 Overview

The main goal of this software implementation phase is to turn the planned architecture into a fully functional system. The project has the databases, front-end, backend integrated together to provide a better and high-quality front-end experience to the user. The PHP and MySQL databases are hosted locally using XAMPP. These are used to manage the data at the backend and they ensure the database connectivity. The HTML, CSS and JavaScript are used to create the frontend interface for the user interaction. This implementation was done using in an iterative and a modular fashion.

6.2 Software and Tools Used

Table 6.1 Software and Tools Used.

Category	Tools / Technology	Purpose
Programming Languages	HTML5, CSS3, JavaScript, PHP	Frontend layout and backend logic
Database	MySQL (via XAMPP)	Persistent data management
APIs	YouTube Data API v3, GitHub REST API, arXiv API, Kaggle API	Fetch educational resources
IDE	Visual Studio Code	Code development and debugging
Version Control	Git and GitHub	Project tracking and collaboration
Server	XAMPP (Apache + MySQL)	Backend hosting and local testing
Browser	Google Chrome	Interface testing and display

6.3 Database Implementation

The phpMyAdmin was used to create the database and set up the MySQL database locally using XAMPP. This design of the database guarantees data security, efficient retrieval of contents and normalization based on the user's choice.

Name of database: study search db

Primary Tables:

1. Users Table: This table holds the details of the user that were provided while the registration process.

2. Search History Table: This table contains the timestamp and the category of the query that has been searched by the user.

3. Feedback: The feedback submitted by the user are stored in the feedback table.

4. Recommendations Table: This table keeps a track of the user's search history and the ranked results to provide recommendations to the user at the home page. PHP scripts linked using MySQL are used to implement all database operations at the backend, including data operations such as data insertion, retrieval and updation guaranteeing data integrity and safe query processing.

6.4 Backend Implementation

- PHP was used to create the backend, which handles
- user registration and login with secure password handling.
- API communication for multi-source federated search.
- The results are ranked and filtered. Creating suggestions based on user history.
- Feedback gathering and presentation for the administrator dashboard.

To guarantee modularity and maintainability of the project, each backend module was created separately and tested separately before the integration of this script.

6.5 Frontend Implementation.

The frontend is the interface where the user interacts with the system. It is designed to be responsive to the ser queries and also be visually engaging using the following techniques:

- HTML5 for structure
- CSS3 for style and layout.
- JavaScript for updating content dynamically and interactivity.

Key Features:

- Intuitive search interface with category tabs
- Smoothened animations when loading data.
- Responsive cards displaying search results
- Recommendation and feedback section.
- Clean, minimalist design that follows the rules of accessibility.

6.6 Integration and Workflow

The integration of the system was done in a step-by-step approach:

1. Setting up the frontend interface: Initially the HTML templates were created and linked with the JavaScript.

2. Database Configuration: MySQL tables were created using phpMyAdmin locally using XAMPP.

3. Implementation of the backend: PHP modules are built for user login, searching the contents and the user feedback.

4. Integration of the API keys: The API keys of the different platforms such as YouTube, GitHub, arXiv, and Kaggle were implemented through backend scripts.

5. Testing and Validation of the System: The functioning of the tabs, modules, security and performance of the system was test using a few sample queries.

6.7 Simulation Workflow

1. User logs in to the system by using the user id and password.

2. The user enters the query and click on the search category like videos, code, papers or datasets.

3. The Backend fetches the results through the APIs of the different platforms.

4. The fetched results are ranked and displayed on the screen using the ranking algorithm.

5. The search history of every user is stored in the database for providing future recommendations.

6. The user can submit the feedback using the feedback form which is available at the end. This submitted feedback is visible in the admin panel at the backend.

Chapter 7

Evaluation and Results

7.1 Functional Evaluation

The designed system was evaluated for functionality, accuracy and responsiveness of all the functions.

Table 7.1 Functional Evaluation

Feature	**Expected Output**	**Observed Result**
User Registration	User added to database	Working correctly
Login Validation	Session created	Working correctly
Federated Search	Data retrieved from APIs	Consistent
Ranking Mechanism	Results sorted by relevance	Verified
Recommendation	Personalized suggestions displayed	Accurate
Feedback Module	Stores and displays messages	Working
Admin Dashboard	Shows users & feedback	Operational

7.2 Usability Evaluation

Evaluation Criteria:

- Ease of navigation
- Design appeal
- Readability and responsiveness
- Clarity of feedback forms

- Accessibility across devices

Average User Satisfaction: 4.7 / 5 Average Loading Time: <3 seconds

The Users, who used the unified search engine appreciated the unified dashboard, it's clean UI and the responsive design.

7.3 Performance Evaluation

Table 7.2 Performance Evaluation

Parameter	Average Time	Results
API Response (YouTube)	2.1 sec	Optimal
API Response (GitHub)	1.8 sec	Optimal
API Response (arXiv)	1.6 sec	Optimal
API Response (Kaggle)	2.3 sec	Acceptable
DB Query Execution	<0.5 sec	Excellent

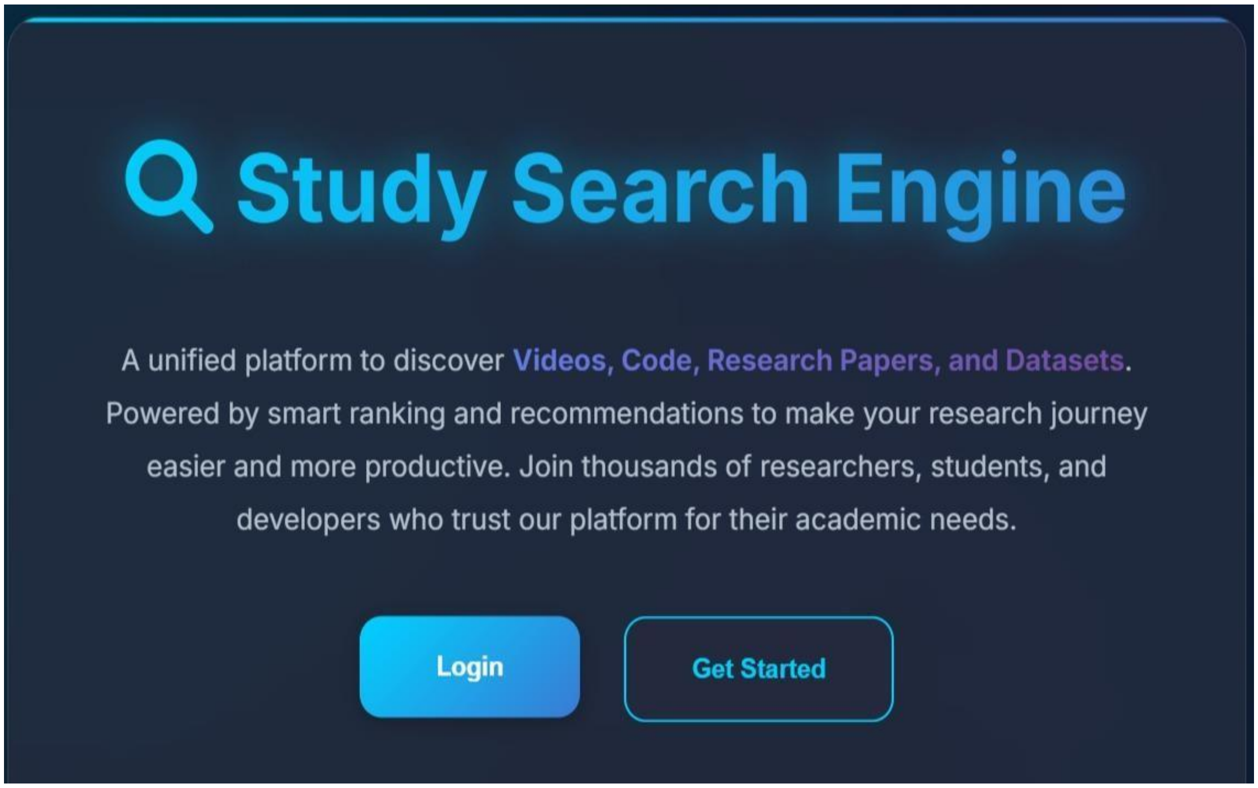
7.4 Security and Reliability Testing

- The passwords of each user are securely hashed using PHP at the backend using the function called password_hash().
- The SQL Injection is prevented through the use of parameterized queries.
- The session timeout mechanism is used to prevent unauthorized access and provide security.
- The error messages are used to avoid the information leaks.

By this the application has passed all the major test cases under the loading of the contents and the authentication conditions

7.5 Results

1.Main Page



Study Search Engine

A unified platform to discover Videos, Code, Research Papers, and Datasets. Powered by smart ranking and recommendations to make your research journey easier and more productive. Join thousands of researchers, students, and developers who trust our platform for their academic needs.

Fig 7.1 Main Page.

The main page of the Study search engine is the entry point for all the users, where the user has two options displayed on the screen. Login and get started. The existing users who have already created their account can click on the login button and enter their user id and password and access their accounts. The other option is Get Started where the new users can register by providing all the necessary details to create an account and once creating their account a message will be displayed saying registration successful and they will be redirected to the login page where the user has to enter the user id and the password to login into the system for accessing their account. This part ensures the smooth and secured onboarding of the users to access personalized features within the platform.

2.Login Page

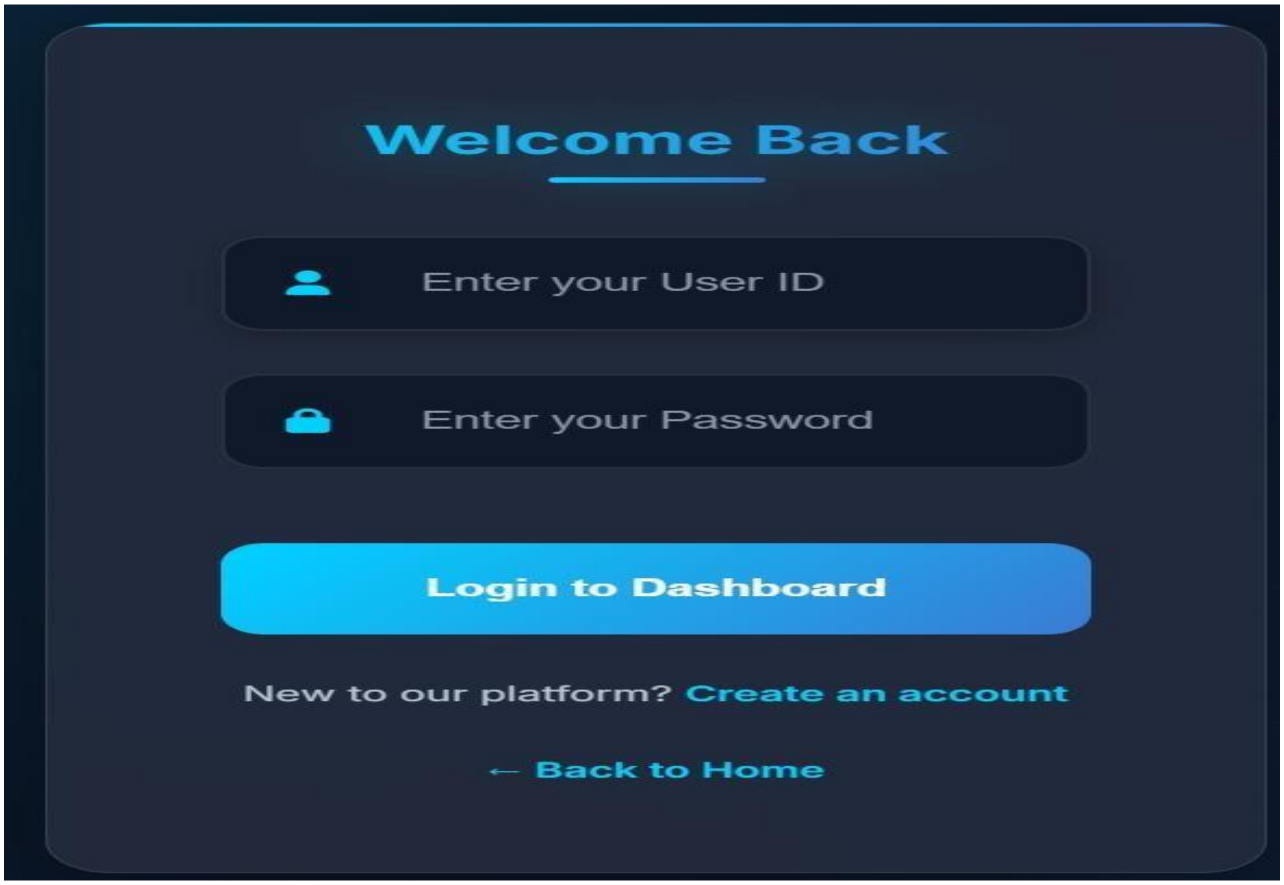
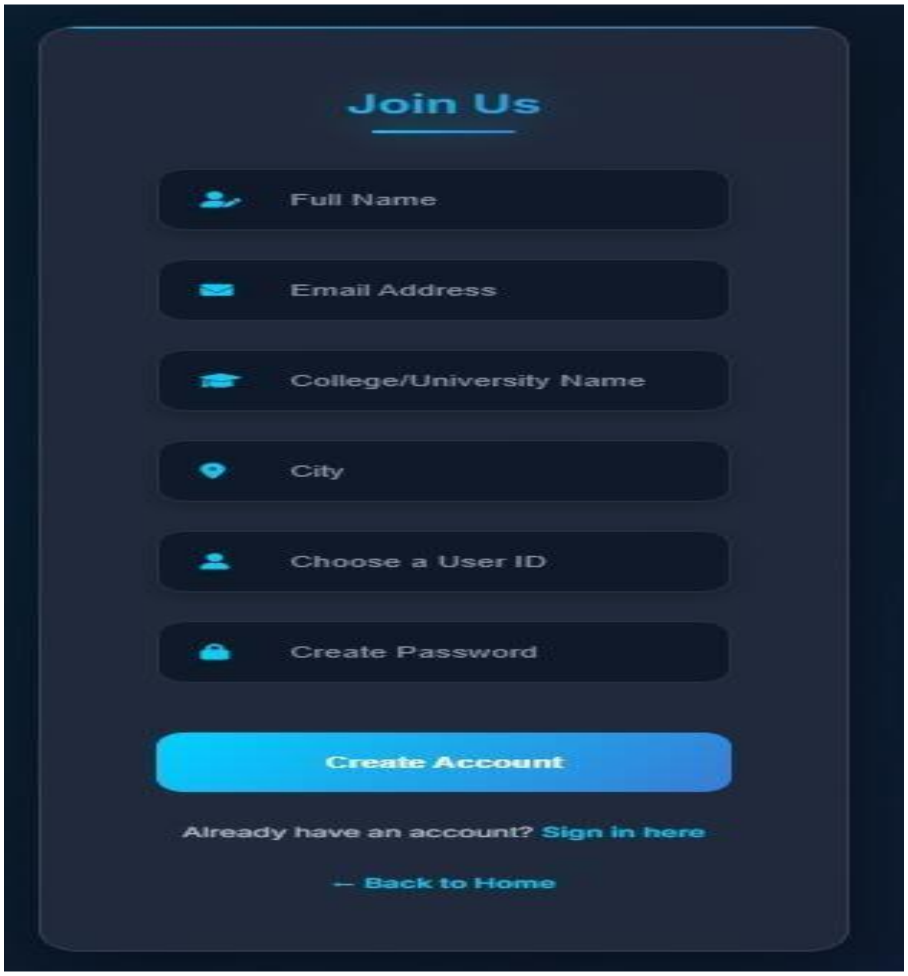


Fig 7.2 Login Page

The Login Page of the study search engine allows the registered users to access their accounts using their user id and the password. By entering the used id and the password and with the authentication at the backend the user can access to their account. This ensures the privacy of the user data and also provides a quick access to the user-specific search and recommendation features. This also provides an option for the non-existing users to register by clicking on Create an account option also Back to Home option where the user lands on the Main Page by clicking on this option. Overall, the login page ensures security by a strong backend authentication and by protecting the user data.

3.Registration Page


Join Us

Full Name

Email Address

College/University Name

City

Choose a User ID

Create Password

Create Account

Already have an account? Sign in here

← Back to Home

Fig 7.3 Registration Page

The registration page is used by the new user who wish to access the system. For this first the user has to register by creating an account. This is done by providing all the required details in the space given. The details that have to be given are Full Name of the user, email address, college name, city, user id and the password. All these details will be stored at the backend in the user table. The user id and password selected by the user are later used for the login purpose by the user to access the search engine and fetch the results.

4.Home Screen Layout

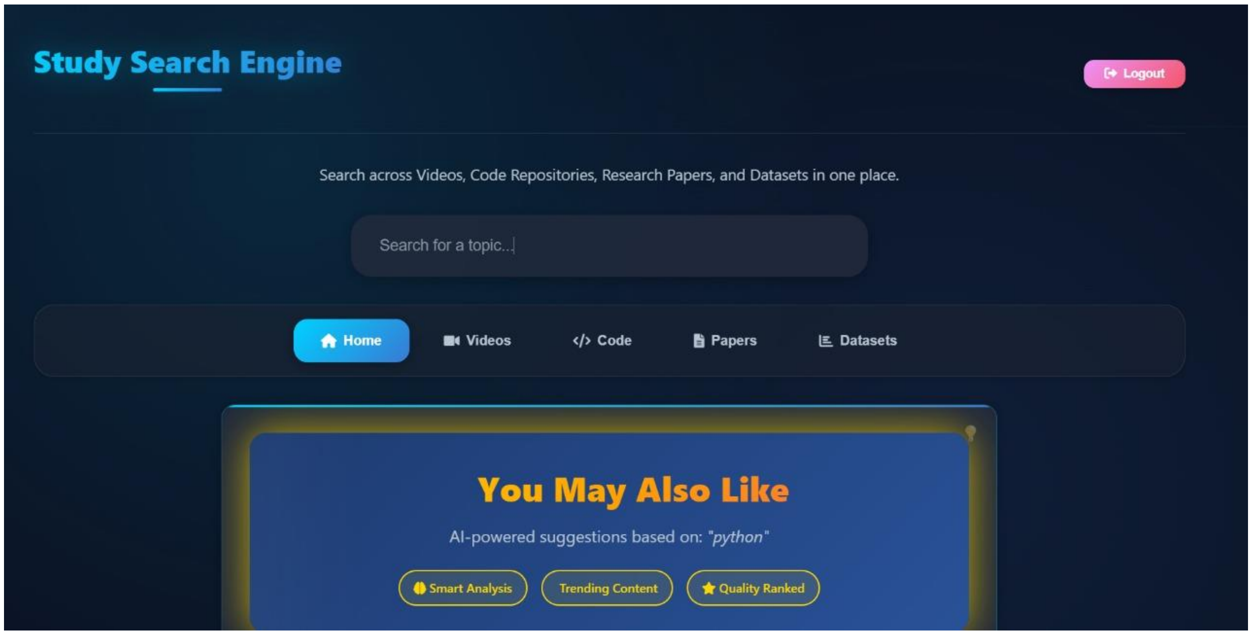
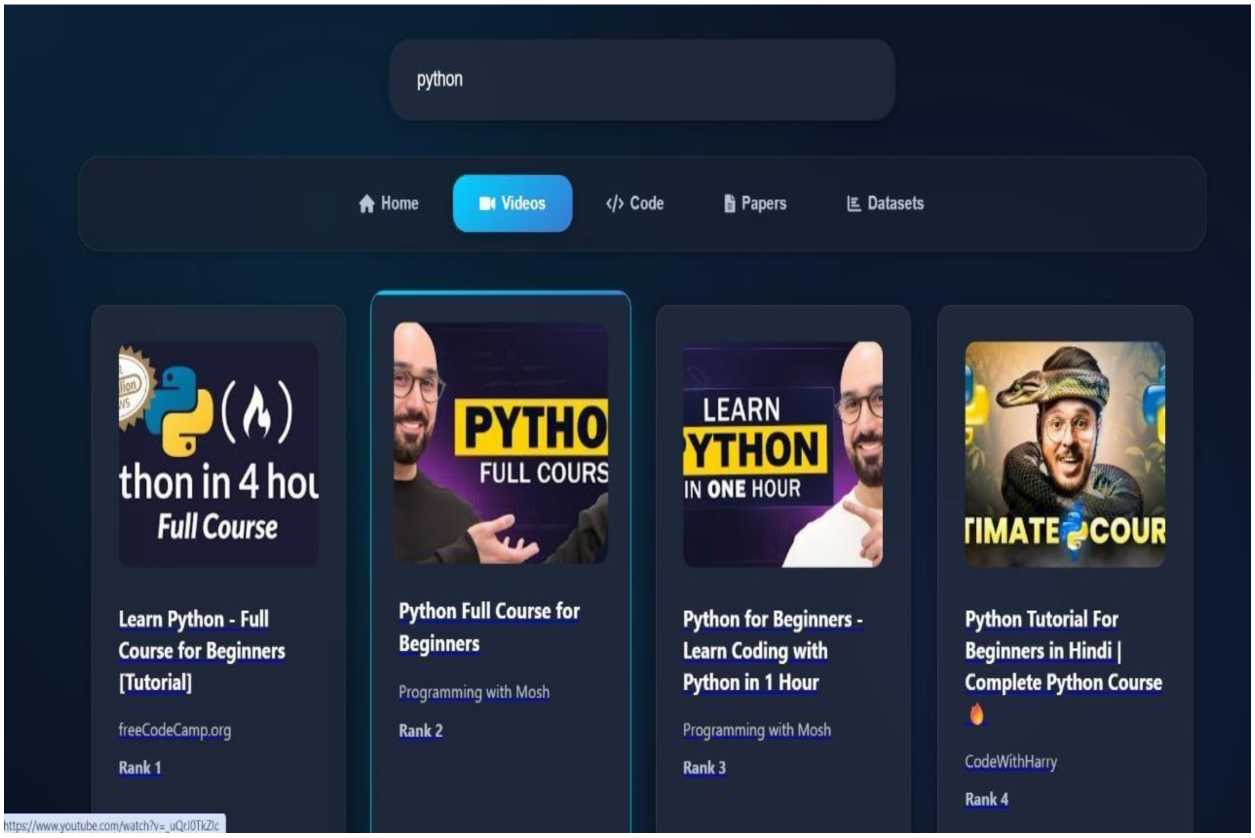


Fig 7.4 Home Screen Layout.

Once the user completes the registration process and logs in into their account by using the user id and the password the user will enter into their profile. The above image is the home screen layout of the website. They provide a search bar for searching the contents and the categories for the search results such as code, videos, datasets and papers. The user enters the query in the search bar and clicks on the required contents like code, videos, datasets or papers to get the required results. This interface is mobile responsive as the viewport meta tag and a flexible CSS tag has been used in the design. This helps in creating a dynamic first impression in the user’s mind, by providing a better user experience.

5.Videos Tab



Learn Python - Full Course for Beginners [Tutorial]
freeCodeCamp.org
Rank 1

Python Full Course for Beginners
Programming with Mosh
Rank 2

Python for Beginners - Learn Coding with Python in 1 Hour
Programming with Mosh
Rank 3

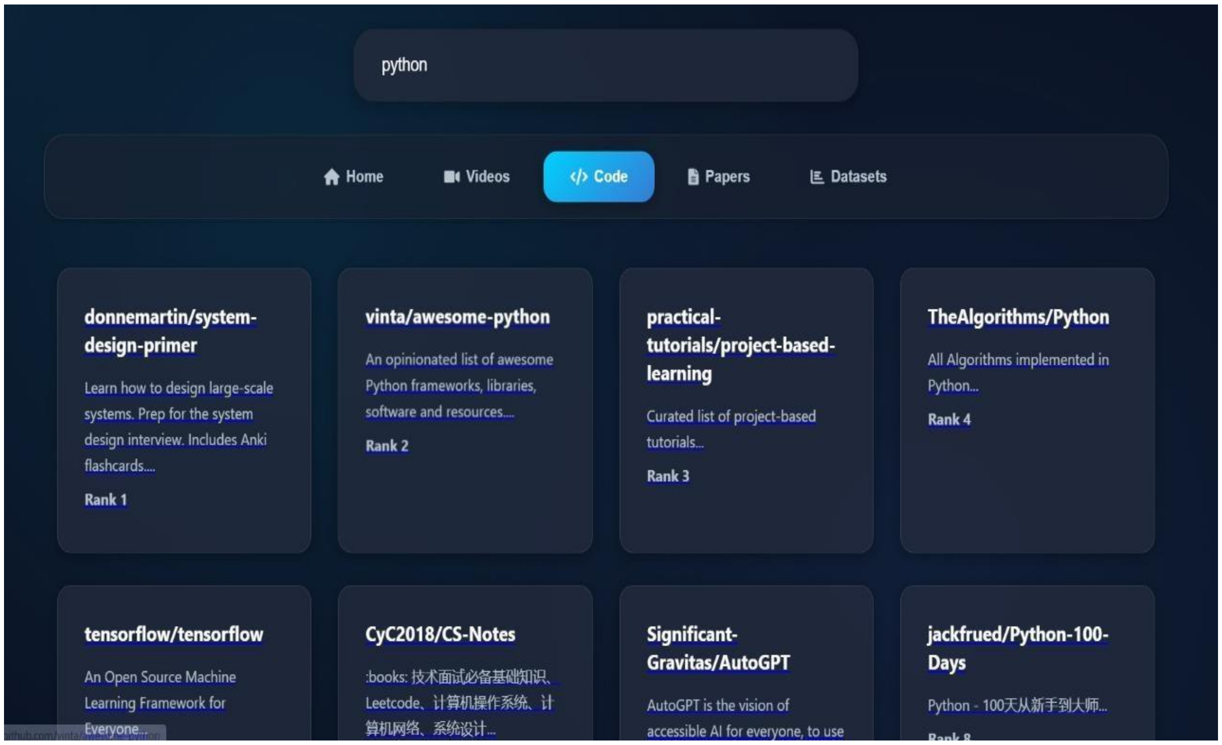
Python Tutorial For Beginners in Hindi | Complete Python Course 🔥
CodeWithHarry
Rank 4

https://www.youtube.com/watch?v=_uQrJ0TkZlc

Fig 7.5 Search Results for Videos

The video search feature uses the YouTube API key and fetches the video results for the query entered. Once the query is entered in the search bar and the user clicks on the videos tab the top 25 videos are fetched from the YouTube and the results are ranked and sorted before it is being displayed using the ranking algorithms. The most relevant results are displayed in an order. Each result is displayed in a card format where the video thumbnail, titles of the video, channel name and their rank is displayed. By clicking on this card, the video directly opens on YouTube in a new tab and this provides a seamless access for the users. The above image shows the search results for the topic python and the user has clicked on the videos tab and the results are displayed in the same format.

6.Code Repositories Tab



donnemartin/system-design-primer

Learn how to design large-scale systems. Prep for the system design interview. Includes Anki flashcards...

Rank 1

vinta/awesome-python

An opinionated list of awesome Python frameworks, libraries, software and resources...

Rank 2

practical-tutorials/project-based-learning

Curated list of project-based tutorials...

Rank 3

TheAlgorithms/Python

All Algorithms implemented in Python...

Rank 4

tensorflow/tensorflow

An Open Source Machine Learning Framework for Everyone...

CyC2018/CS-Notes

:books: 技术面试必备基础知识、Leetcode、计算机操作系统、计算机网络、系统设计...

Significant-Gravitas/AutoGPT

AutoGPT is the vision of accessible AI for everyone, to use

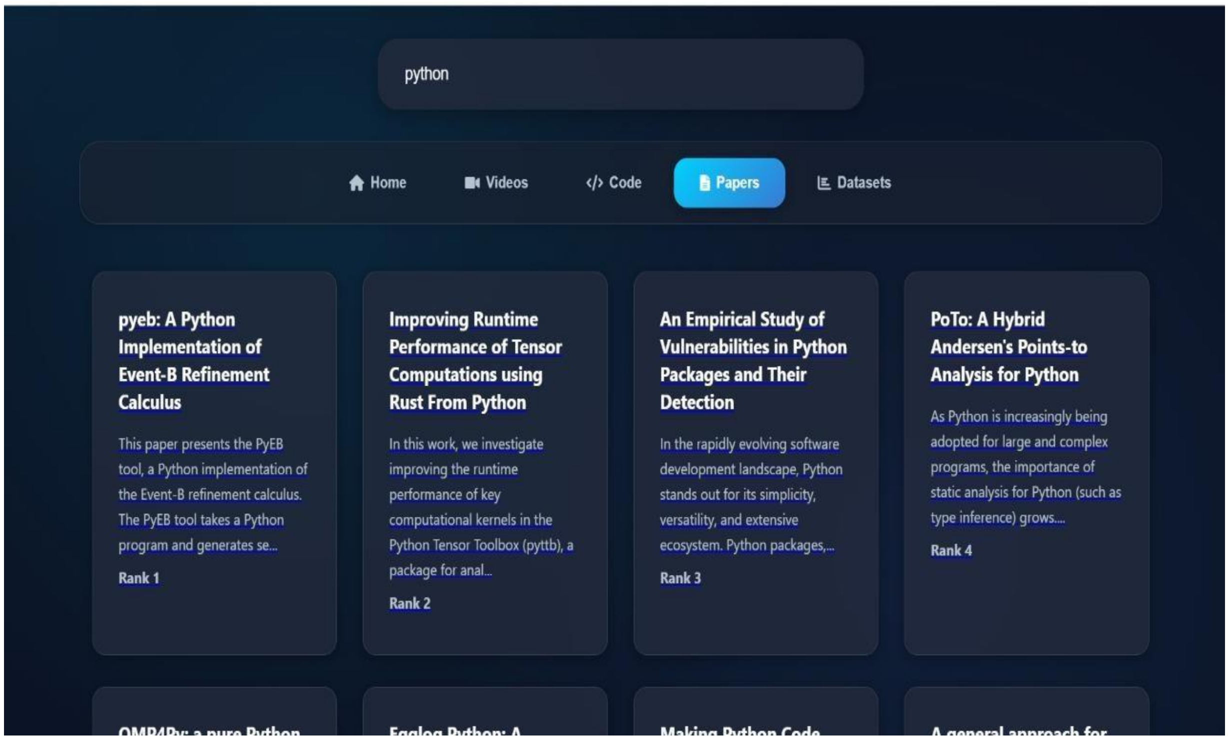
jackfrued/Python-100-Days

Python - 100天从新手到大师...

Fig 7.6 Search Results for Code Repositories

The code repositories tab displays the search results for the code repositories from the GitHub. Once the user searches for the query and clicks on the code tab the results are retrieved and ranked based on the ranking algorithms of the GitHub by considering a few factors such as stars, forks and other aspects these ranked results are retrieved and sorted based on their ranks and displayed in the form of placeholders or a card like structure where the repository name, description and the rank of the results is displayed. By clicking on this the user will be directed to the GitHub webpage and the user can directly access this repository.

7.Research Papers Tab



pyeb: A Python Implementation of Event-B Refinement Calculus

This paper presents the PyEB tool, a Python implementation of the Event-B refinement calculus. The PyEB tool takes a Python program and generates se...

Rank 1

Improving Runtime Performance of Tensor Computations using Rust From Python

In this work, we investigate improving the runtime performance of key computational kernels in the Python Tensor Toolbox (pyttb), a package for anal...

Rank 2

An Empirical Study of Vulnerabilities in Python Packages and Their Detection

In the rapidly evolving software development landscape, Python stands out for its simplicity, versatility, and extensive ecosystem. Python packages,...

Rank 3

PoTo: A Hybrid Andersen's Points-to Analysis for Python

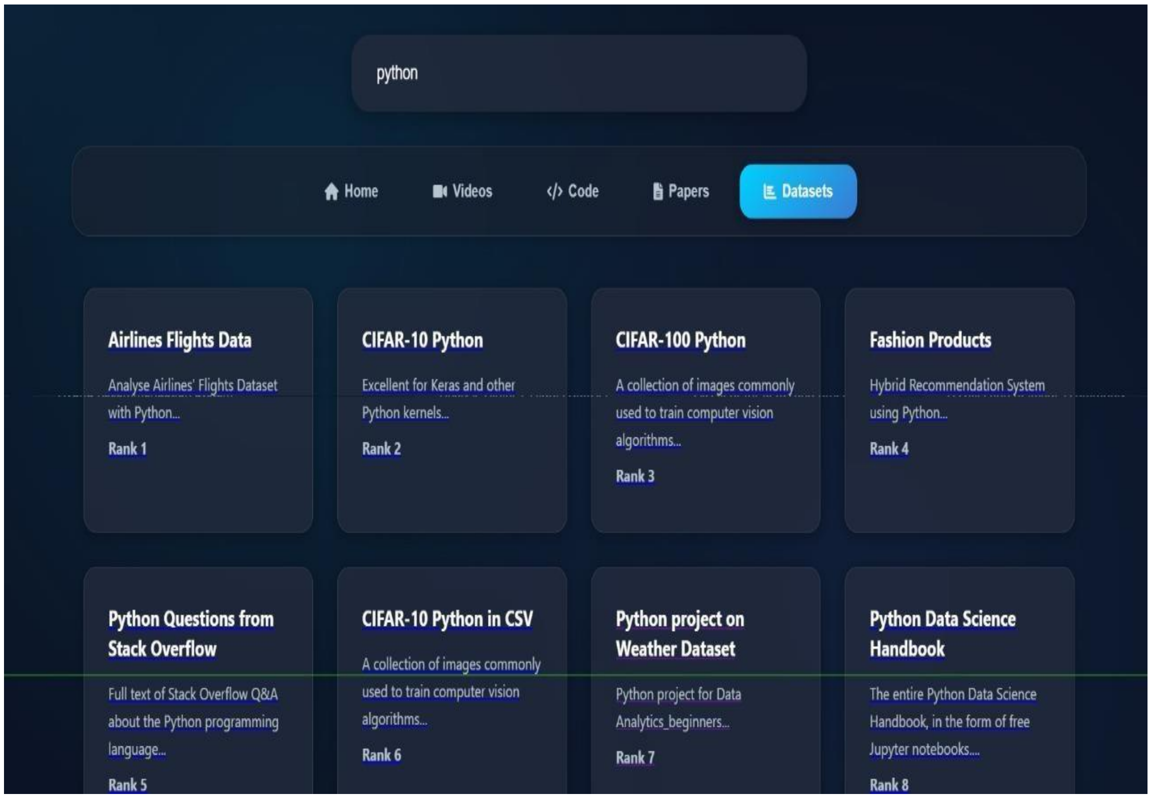
As Python is increasingly being adopted for large and complex programs, the importance of static analysis for Python (such as type inference) grows....

Rank 4

Fig 7.7 Search Results for Research Papers

The research papers are displayed using the arXiv data. This helps to fetch the academic papers based on the user query entered in the search bar. Each paper is analyzed based on the ranking algorithm, the factors such as recency and a score is used to determine whether the paper is relevant or not. After analyzing this, the results are ranked and sorted in an order for being displayed in the card like structure. The name of the papers, a small intro of the paper is displayed along with the rank. By clicking on this result, the user will be directed to the arXiv webpage that will open in a new tab and the user can directly access the paper.

8.Datasets Tab



Airlines Flights Data
Analyse Airlines' Flights Dataset with Python...
Rank 1

CIFAR-10 Python
Excellent for Keras and other Python kernels...
Rank 2

CIFAR-100 Python
A collection of images commonly used to train computer vision algorithms...
Rank 3

Fashion Products
Hybrid Recommendation System using Python...
Rank 4

Python Questions from Stack Overflow
Full text of Stack Overflow Q&A about the Python programming language...
Rank 5

CIFAR-10 Python in CSV
A collection of images commonly used to train computer vision algorithms...
Rank 6

Python project on Weather Dataset
Python project for Data Analytics_beginners...
Rank 7

Python Data Science Handbook
The entire Python Data Science Handbook, in the form of free Jupyter notebooks....
Rank 8

Fig 7.8 Search Results for Datasets

The dataset tab uses the Kaggle API to retrieve the results from the Kaggle site and display them based on the user's query. The user searches for the required topic on the search bar and clicks on the datasets tab, the datasets are displayed using the ranking algorithm using the high popularity datasets. These results are displayed in a card structure with the name of the dataset, description of the dataset and the rank of the results. These results are sorted and displayed based on their rank. Once the user finds the required content and the user clicks on the dataset, the user will be directed to the Kaggle website that opens on a new tab, where the user can directly access the dataset. This allows the users to explore the relevant datasets for their projects and work on them efficiently.

9.Skeleton Loader

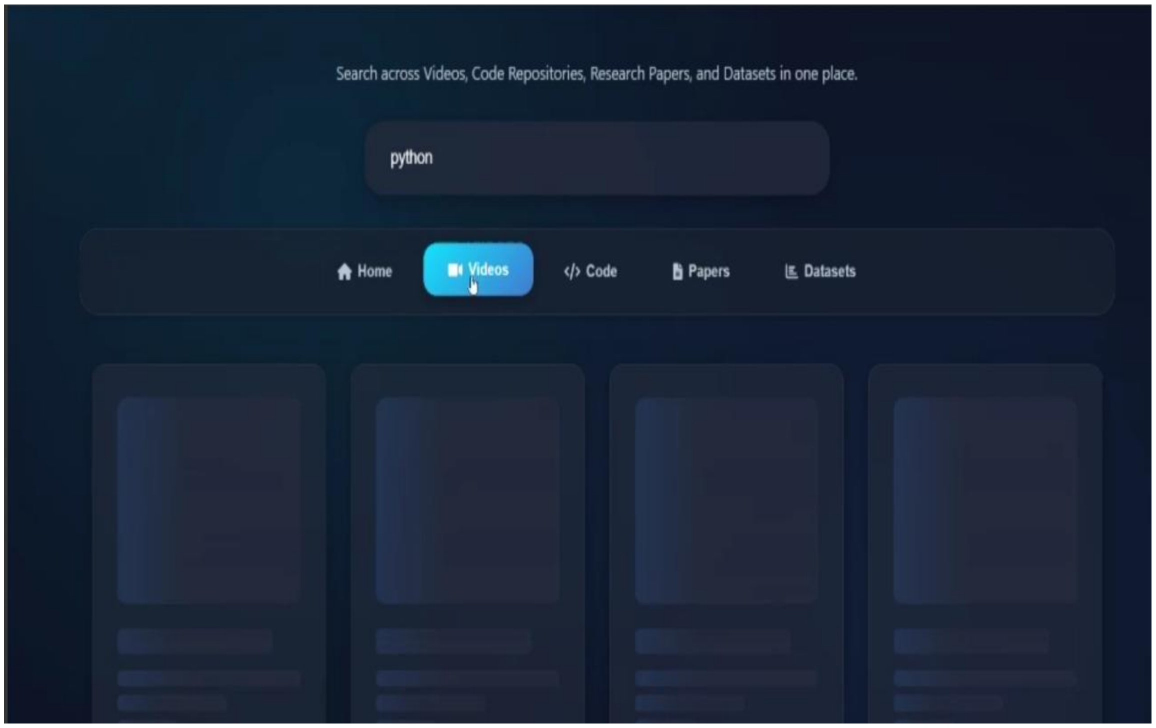

Search across Videos, Code Repositories, Research Papers, and Datasets in one place.
python
Home
Videos
Code
Papers
Datasets

Fig 7.9 Skeleton Loader Design

To enhance the user experience while the API calls are being made and the contents are loading, the skeleton loader module is added. This should the animated placeholder while the contents are loading in the structure of the result card. The skeleton loader is developed using CSS animations to create a smooth effect. This maintains visual engagement and prevents abrupt blank screens during the slow network. Overall, the skeleton loader module is used for the better user experience and to prevent blank screens.

10.Recommendations on Home Tab-Videos and code

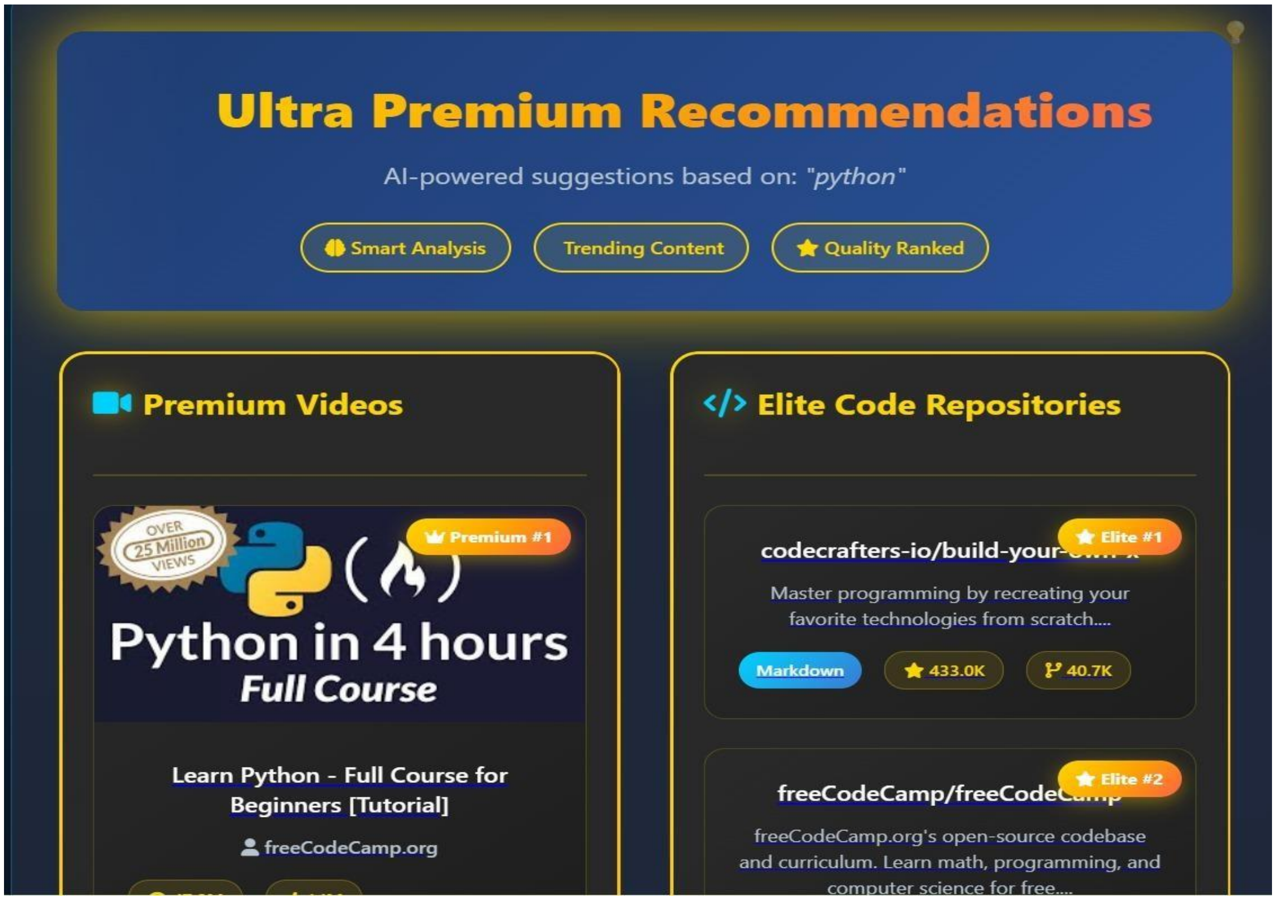
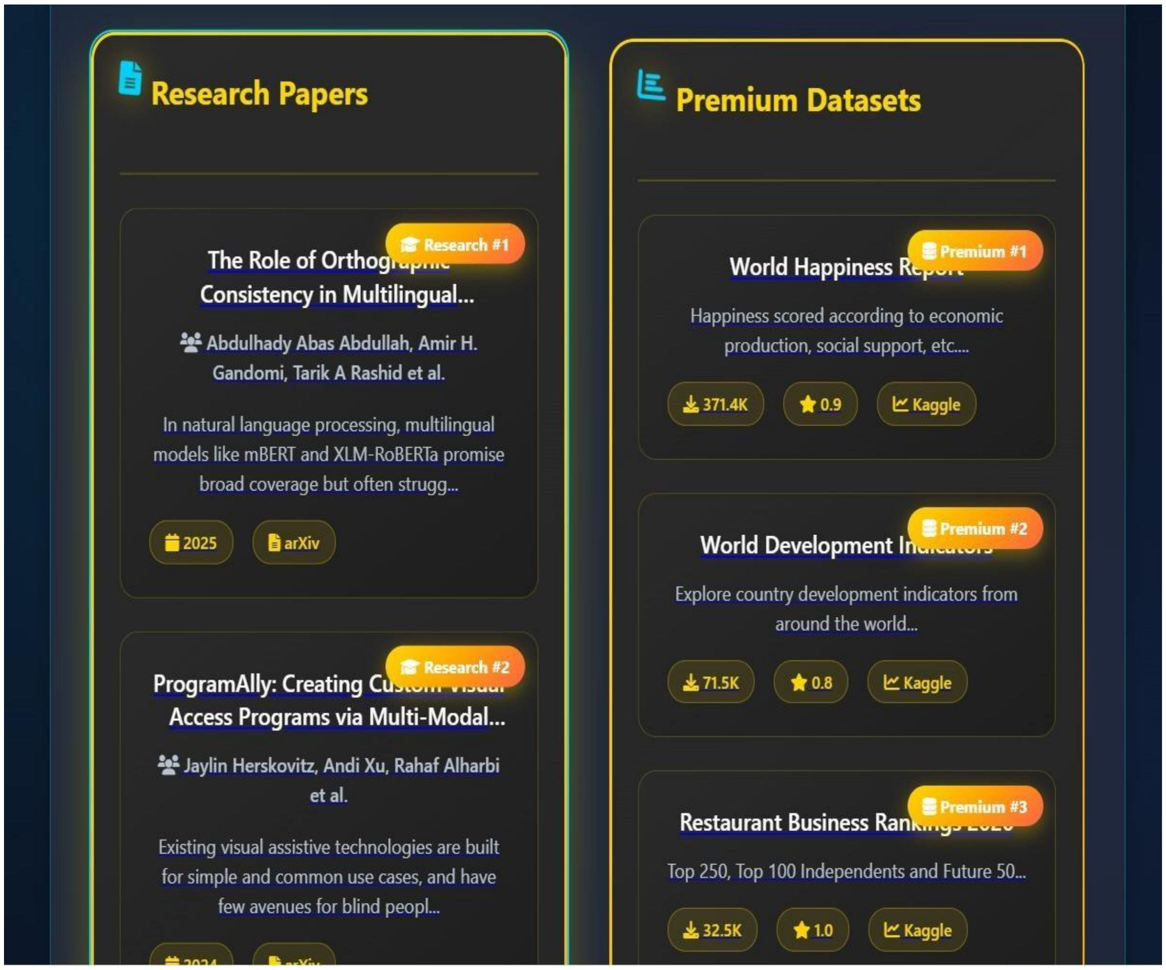

Ultra Premium Recommendations
AI-powered suggestions based on: "python"
Smart Analysis
Trending Content
Quality Ranked
Premium Videos
Premium #1
OVER 25 Million VIEWS
Python in 4 hours
Full Course
Learn Python - Full Course for Beginners [Tutorial]
freeCodeCamp.org
Elite Code Repositories
codecrafters-io/build-your-own-x
Elite #1
Master programming by recreating your favorite technologies from scratch....
Markdown
433.0K
40.7K
freeCodeCamp/freeCodeCamp
Elite #2
freeCodeCamp.org's open-source codebase and curriculum. Learn math, programming, and computer science for free....

Fig 7.10 Recommendations at the Home Tab for Videos and Code Repositories.

Once the user starts using the application and searches for the contents, these queries of the user are stored at the backend and based on these results the recommendations are given to the user at the home tab based on the previous search results. This recommendation keeps on updating. The above image shows the video and code recommendations given to a user, where the user can access these recommendations by scrolling down through these results, where the results are displayed in the grid structure where the recommended code repositories are videos are aligned side by side.

11.Recommendation for Datasets and Research Paper


Research Papers

Research #1
The Role of Orthographic Consistency in Multilingual...
Abdulhady Abas Abdullah, Amir H. Gandomi, Tarik A Rashid et al.
In natural language processing, multilingual models like mBERT and XLM-RoBERTa promise broad coverage but often strugg...
2025 arXiv

Research #2
ProgramAlly: Creating Custom Visual Access Programs via Multi-Modal...
Jaylin Herskovitz, Andi Xu, Rahaf Alharbi et al.
Existing visual assistive technologies are built for simple and common use cases, and have few avenues for blind peopl...

Premium Datasets

Premium #1
World Happiness Report
Happiness scored according to economic production, social support, etc....
371.4K 0.9 Kaggle

Premium #2
World Development Indicators
Explore country development indicators from around the world...
71.5K 0.8 Kaggle

Premium #3
Restaurant Business Rankings 2020
Top 250, Top 100 Independents and Future 50...
32.5K 1.0 Kaggle

Fig. 7.11 Recommendations for Research Papers and Datasets.

The recommendations for the Research Papers and the data sets are provided below the recommendations given for the code repositories and videos. These contents can be accessed by scrolling down the recommended videos and code repositories. These recommended results are organized in the containers one below the other in two columns. They provide the recommendations to the user based on the search history and the user interaction with the contents retrieved earlier. Overall, the recommendations for the code repositories, videos, datasets and the research papers provided on the home tab where the contents are aligned side-by-side in two columns one below the other.

12.Feedback Form Interface

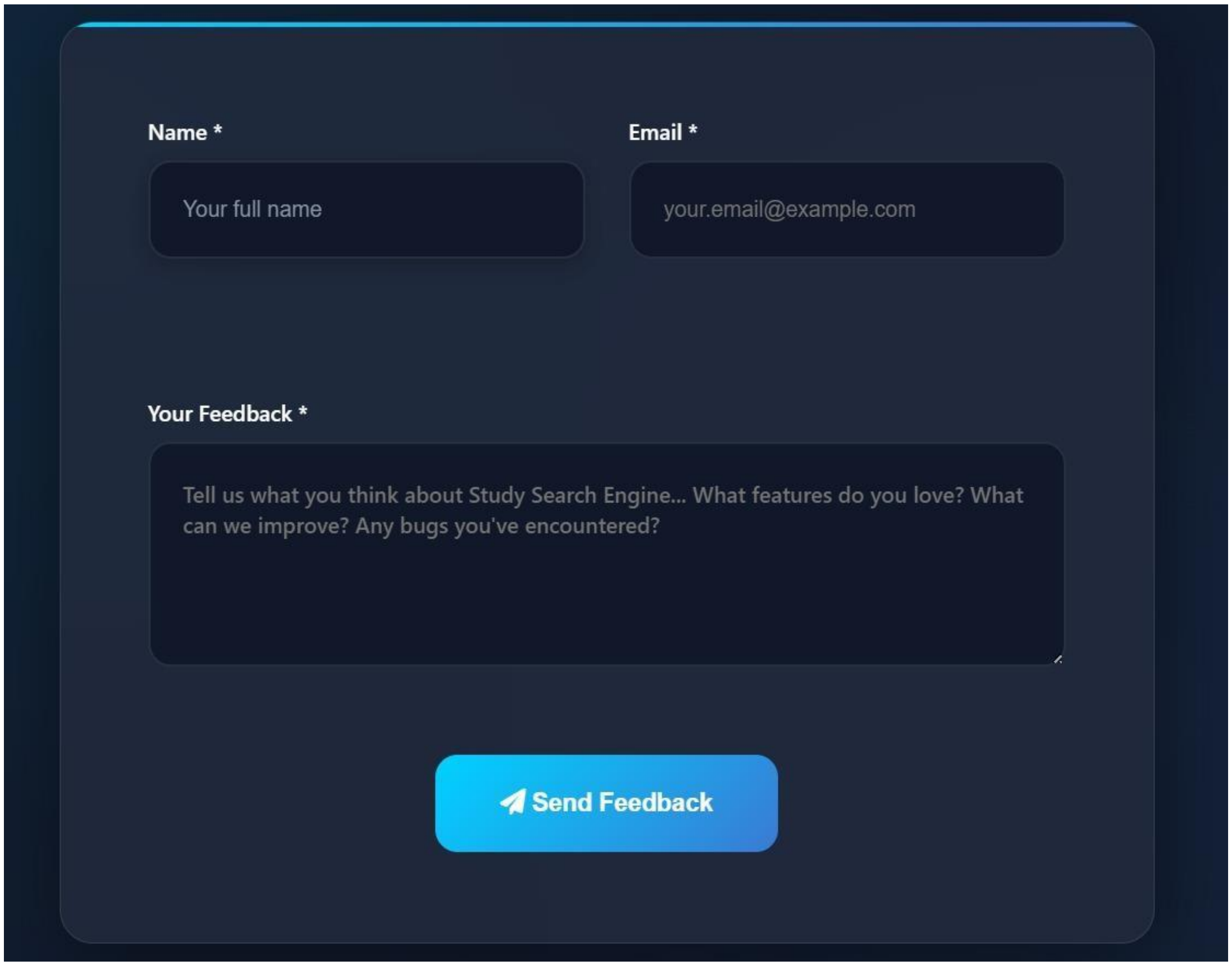

Name *
Your full name
Email *
your.email@example.com
Your Feedback *
Tell us what you think about Study Search Engine... What features do you love? What can we improve? Any bugs you've encountered?
Send Feedback

Fig.7.12 Feedback Form Interface

The feedback section allows the users to provide their feedback and suggestions that helps the system to improve. This form has three input fields one for the name where the user enters the username, next the Email where the user enters their email-id and finally the feedback where the user types the feedback. After filling all these details, the user ensures that all the details entered are correct and clicks on the submit button. The submitted feedback will be stored at the backend inside the feedback table. This feedback can be accessed using the backend. This feature ensures continuous improvement in the development of the system.

13.Submission Message

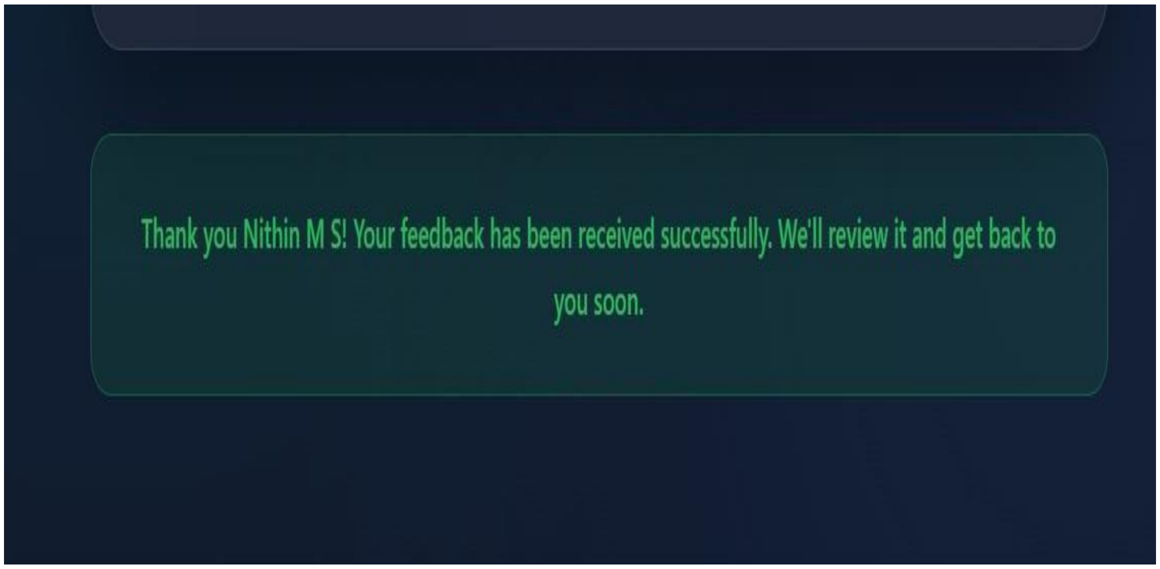
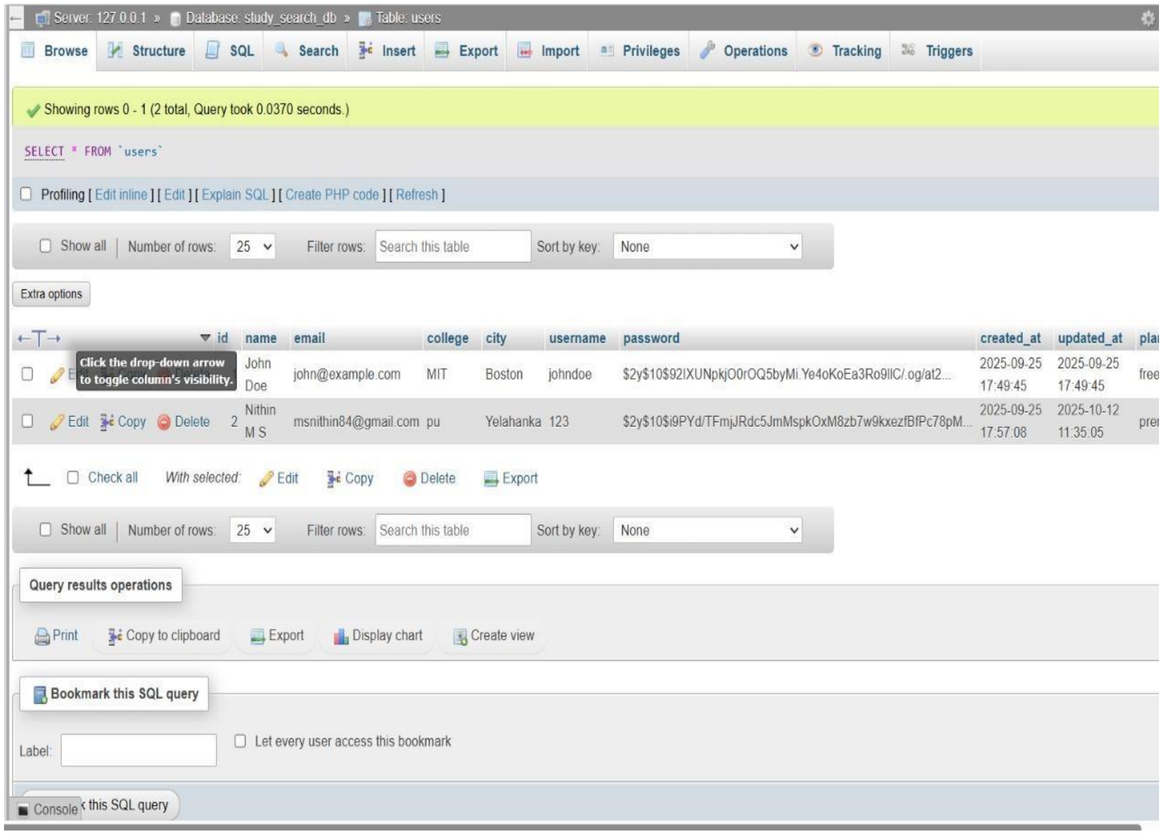

Thank you Nithin M S! Your feedback has been received successfully. We'll review it and get back to you soon.

Fig.7.13 Submission message displayed after entering feedback

After Once the user types the feedback content and clicks on submit option, a thank you message will be displayed at the screen saying that “Thank you, your feedback has been received successfully. We’ll review this and get back to you soon”. The same message would be displayed as above as shown in the figure while the user submits the feedback. This encourages the user to use the same platform. Finally, the feedback form uses the user suggestion and improve the Study Search Engine’s features and experience.

14.User Details Page (Admin View)


Server: 127.0.0.1 » Database: study_search_db » Table: users
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

Showing rows 0 - 1 (2 total, Query took 0.0370 seconds.)

```
SELECT * FROM `users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
Click the drop-down arrow to toggle column's visibility.

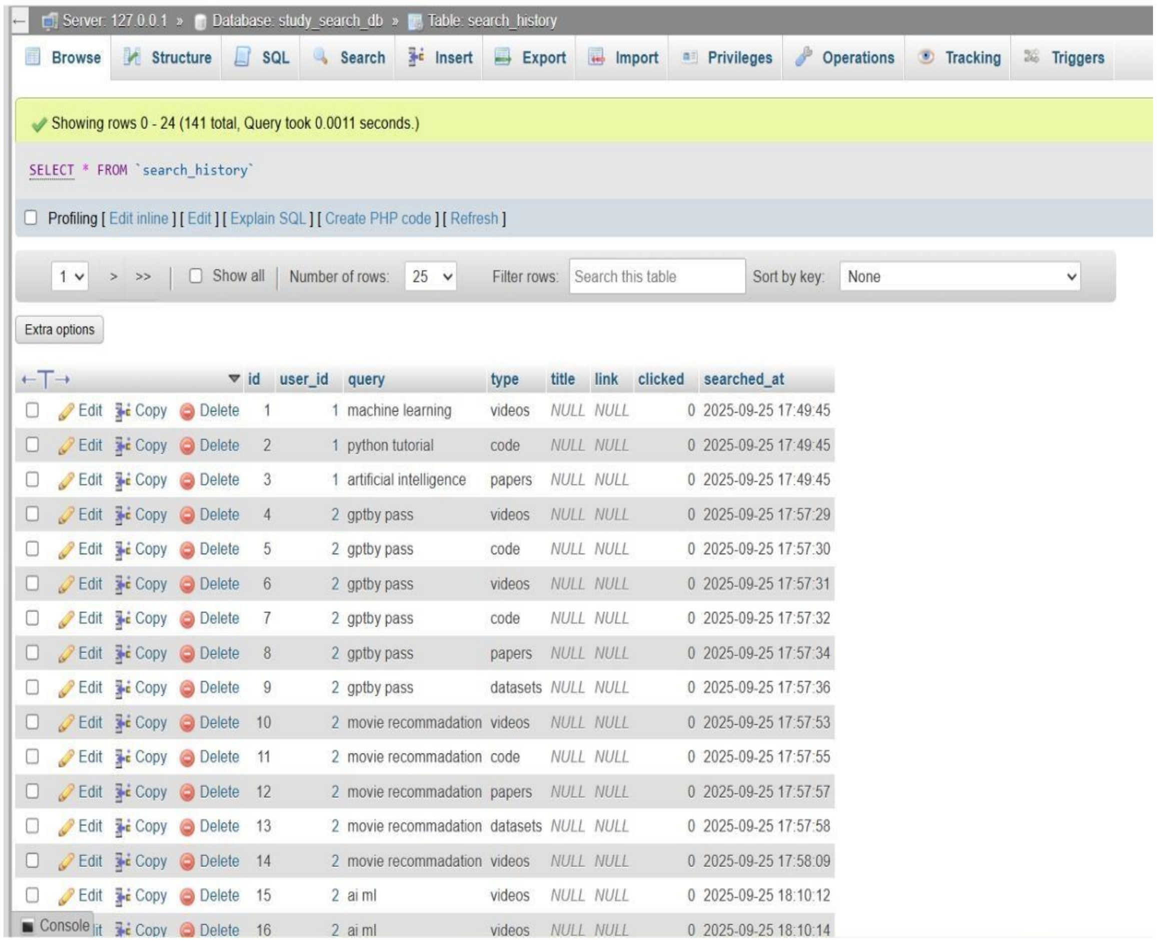
	id	name	email	college	city	username	password	created_at	updated_at	pla
E		John Doe	john@example.com	MIT	Boston	johndoe	$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2...	2025-09-25 17:49:45	2025-09-25 17:49:45	free
Edit Copy Delete	2	Nithin M S	msnithin84@gmail.com	pu	Yelahanka	123	$2y$10$i9PYd/TFmjJRdc5JmMspkOxM8zb7w9kxezfBfPc78pM...	2025-09-25 17:57:08	2025-10-12 11:35:05	pre

Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Console
k this SQL query

Fig 7.14 User Details Page (Admin View)

The user details page is accessible using the XAMPP admin panel. This allows the administrator to view and manage all the registered user information stored in the database. This information is fed into the database by the user while registering their account on the site. It displays key details such as usernames, email address, and account related data. Through these records the admin can monitor the details and can also update the information if required, this ensures data integrity across the different platforms by monitoring and maintaining the user profiles within the Study Search Engine.

15.Search History Page (Admin View)


Server: 127.0.0.1 » Database: study_search_db » Table: search_history
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Filter rows:
Search this table
Extra options
Console

Showing rows 0 - 24 (141 total, Query took 0.0011 seconds.)

```
SELECT * FROM `search_history`
```

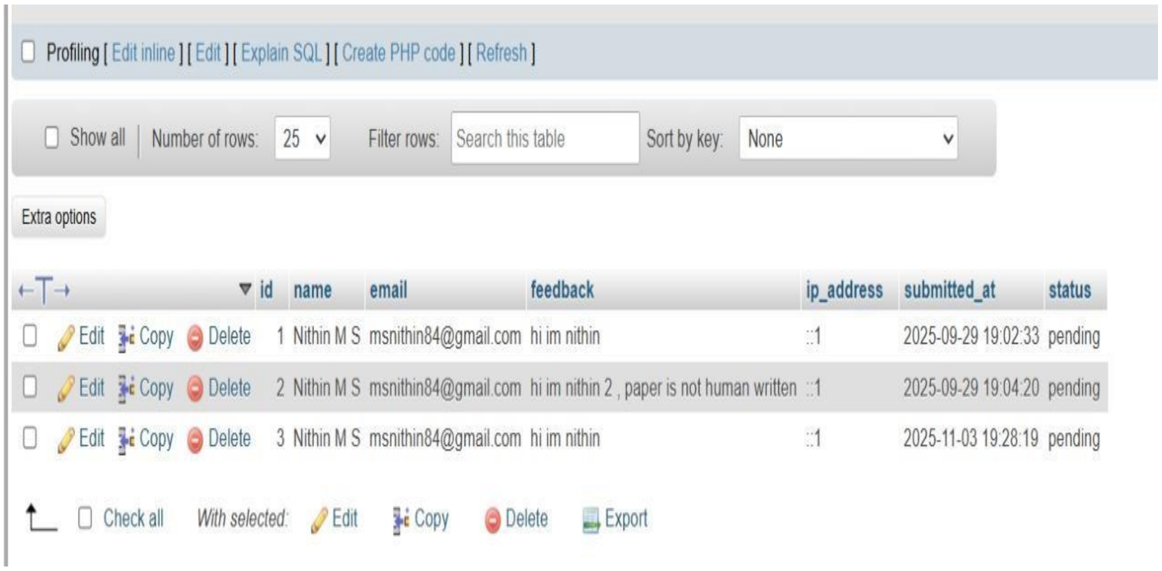
Number of rows: 25 | Sort by key: None

	id	user_id	query	type	title	link	clicked	searched_at
Edit Copy Delete	1	1	machine learning	videos	NULL	NULL	0	2025-09-25 17:49:45
Edit Copy Delete	2	1	python tutorial	code	NULL	NULL	0	2025-09-25 17:49:45
Edit Copy Delete	3	1	artificial intelligence	papers	NULL	NULL	0	2025-09-25 17:49:45
Edit Copy Delete	4	2	gptby pass	videos	NULL	NULL	0	2025-09-25 17:57:29
Edit Copy Delete	5	2	gptby pass	code	NULL	NULL	0	2025-09-25 17:57:30
Edit Copy Delete	6	2	gptby pass	videos	NULL	NULL	0	2025-09-25 17:57:31
Edit Copy Delete	7	2	gptby pass	code	NULL	NULL	0	2025-09-25 17:57:32
Edit Copy Delete	8	2	gptby pass	papers	NULL	NULL	0	2025-09-25 17:57:34
Edit Copy Delete	9	2	gptby pass	datasets	NULL	NULL	0	2025-09-25 17:57:36
Edit Copy Delete	10	2	movie recommadation	videos	NULL	NULL	0	2025-09-25 17:57:53
Edit Copy Delete	11	2	movie recommadation	code	NULL	NULL	0	2025-09-25 17:57:55
Edit Copy Delete	12	2	movie recommadation	papers	NULL	NULL	0	2025-09-25 17:57:57
Edit Copy Delete	13	2	movie recommadation	datasets	NULL	NULL	0	2025-09-25 17:57:58
Edit Copy Delete	14	2	movie recommadation	videos	NULL	NULL	0	2025-09-25 17:58:09
Edit Copy Delete	15	2	ai ml	videos	NULL	NULL	0	2025-09-25 18:10:12
it Copy Delete	16	2	ai ml	videos	NULL	NULL	0	2025-09-25 18:10:14

Fig 7.15 Search History Page (Admin View).

The search history page can be accessed by the admin using the XAMPP server where a separate table for search history has been created and when every time the user searches for a content an entry is made in the search history table and this data is stored securely. This is a vital component for the recommendation logic and also makes it a vital component for the better analytics of the system enhancement.

16.Feedback Page (Admin View).


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
Check all
With selected: Edit Copy Delete Export

	id	name	email	feedback	ip_address	submitted_at	status
Edit Copy Delete	1	Nithin M S	msnithin84@gmail.com	hi im nithin	::1	2025-09-29 19:02:33	pending
Edit Copy Delete	2	Nithin M S	msnithin84@gmail.com	hi im nithin 2 , paper is not human written	::1	2025-09-29 19:04:20	pending
Edit Copy Delete	3	Nithin M S	msnithin84@gmail.com	hi im nithin	::1	2025-11-03 19:28:19	pending

Fig 7.16 Feedback Page (Admin View).

The feedback submitted by the user is visible using the feedback table. The XAMPP interface displays the user submitted feedback entries along with the user ID of the user, email id, feedback, date and time at which the feedback was submitted. This helps the administrators to review the suggestions given by the users and also plan for the improvements based on the user feedbacks. The centralized access for this data through XAMPP ensures that the data is protected and securely collected and the it is used for platform enhancement.

17.Database Structure (Admin View).

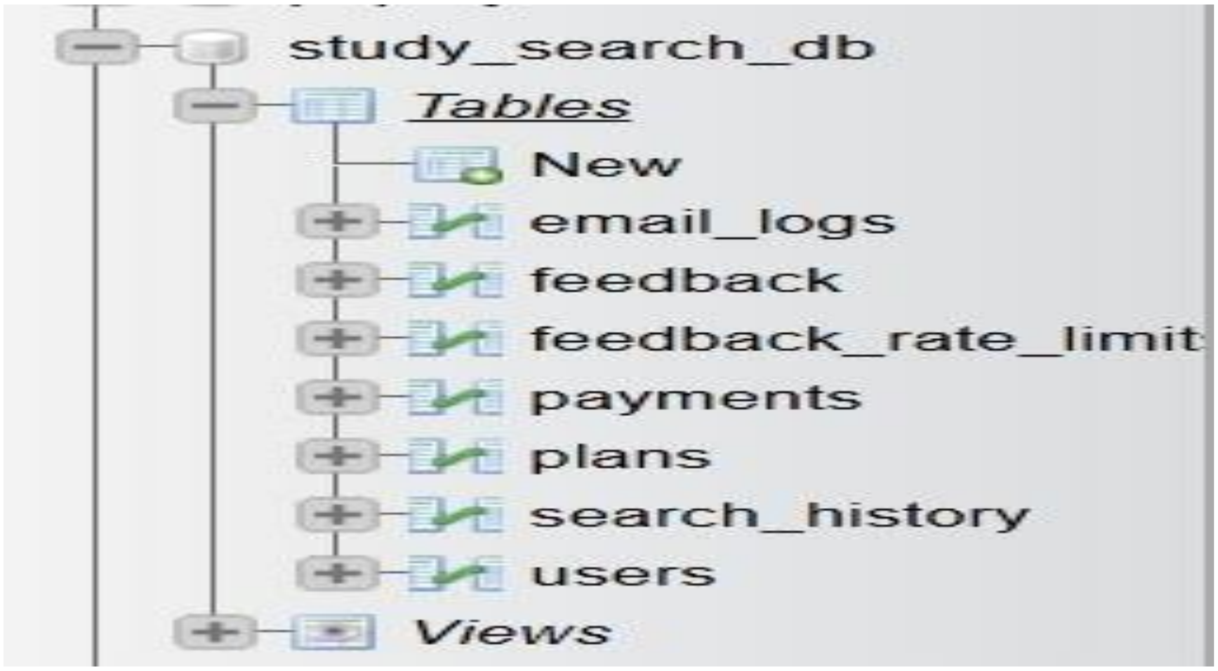


Fig 7.17 Database Structure (Admin View).

The above image is the final database structure that can be accessed through XAMPP, this represents how different tables like users, feedbacks, search_history are linked with each other and it provides the admin a clear view of the backend, data flow and relationship of the entities. This structure design ensures better data management, efficient data retrieval and smooth and secure backend operations. This serves as a foundation for maintaining the consistency and scalability of the system.

Chapter 8

Social, Legal, Ethical, Sustainability and Safety Aspects

8.1 Social Aspects

- Students are given access to a centralized platform of open resources, by this the project improves educational inclusivity.
- It encourages students all over the world to learn on their own.
- Using open-source tools like GitHub and arXiv to collaborate on research.
- Lessening educational disparities by making resources freely accessible. materials.

8.2 Legal Aspects

- All APIs of the various platforms like (GitHub, YouTube, arXiv, and Kaggle) are accessible in accordance with acceptable API usage guidelines.
- Only references and links are displayed; no copyrighted content is saved or shared.
- Because user data stays local, it complies with GDPR and data privacy standards.

8.3 Ethical Aspects

- Transparent ranking logic — no bias in recommendations.
- Ethical handling of data with no exploitation or sharing.
- Encourages responsible digital learning and open access principles.

8.4 Sustainability Aspects

- Environmentally friendly: This process of digital learning minimizes the use of paper like the normal methods.
- Energy efficient: This process is done using lightweight PHP at the backend with minimal server load.
- Long-term sustainability: The modular architecture supports easy maintenance and feature expansion.

8.5 Safety Aspects

- The HTTPS compatibility ensures secure data transmission from various platforms.
- Password encryption prevents unauthorized access, as the user has to provide the user id and password before they login in into the system.
- Robust error-handling features ensures stable operation of the system even under API downtime or at the time or low internet connection.

Chapter 9
Conclusion And Future Work

9.1 Conclusion

The viability and significance of a full-stack, unified educational search and recommendation system are effectively illustrated by Search Engine. The system significantly cuts down on the time and effort students spend looking for high-quality learning resources on different platforms and by integrating YouTube, GitHub, arXiv, and Kaggle into a single platform.

Principal Accomplishments:

- Setting up a safe system for user authentication using PHP and MySQL.
- The creation of a federated API search that integrates four significant learning platforms.
- Developed a ranking algorithm that guarantees the quality and relevancy if the results that are being retrieved
- The launch of a recommendation system that uses past user searches.
- Adding an admin dashboard and feedback system to enhance the system.

A full-stack architecture with modular scalability that is ready for deployment.

Thus, by providing learners with the accessibility of the resources, persistent contextual information, the project advances in both the domains that is technological and pedagogical.

9.2 Limitations

- Frequency-based logic is used for providing the current recommendations; deep learning or natural language processing may be incorporated into future models.
- Only Two APIs are available; more are required for a wider range of education.
- Requires cloud deployment for real-world access; hosted on localhost (XAMPP).

9.3 Future Enhancements

- **AI-Powered Suggestions:** For individualized ranking they use TF-IDF similarity or semantic embeddings.

- **Advanced Analytics Dashboard:** They provide graphs illustrating user demographics, learning trends, and popular search trends.

- **Cloud Integration:** For greater accessibility, host on AWS or Azure.

- **Chatbot Integration:** Include a conversational assistant to support queries in natural language.

- **Multi-Language Search Support:** Use NLP-based translation APIs to reach users worldwide

References

[1] Hambarde, K.A. and Proença, H., 2023. Information Retrieval: Recent Advances and Beyond. *Proc.*, Volume 11, 14 July 2023. Available at: https://ieeexplore.ieee.org/document/10184013 .

[2] Limongelli, C., Lombardi, M., Marani, A. and Taibi, D., 2022. A Semantic Approach to Ranking Techniques: Improving Web Page Searches for Educational Purposes. *Proc.*, Volume 10, 27 June 2022. Available at: https://ieeexplore.ieee.org/document/9807311 .

[3] Ingavélez-Guerra, P., Robles-Bykbaev, V.E., Pérez-Muñoz, A., Hilera-González, J. and Otón-Tortosa, S., 2022. Automatic Adaptation of Open Educational Resources: An Approach From a Multilevel Methodology Based on Students' Preferences, Educational Special Needs, Artificial Intelligence and Accessibility Metadata. *Proc.*, Volume 10, 27 January 2022. Available at: https://ieeexplore.ieee.org/document/9669174 .

[4] Yadav, U. and Duhan, N., 2021. Efficient Retrieval of Data Using Semantic Search Engine Based on NLP and RDF. *Proc.*, Volume 6, 27 October 2021. Available at: https://ieeexplore.ieee.org/document/10246838.

[5] Zhou, R., Xia, D., Wan, J. and Zhang, S., 2020. An Intelligent Video Tag Recommendation Method for Improving Video Popularity in Mobile. *Proc.*, Volume 8, 14 January 2020. Available at: https://ieeexplore.ieee.org/document/8938802 .

[6] Muepu, D.M. and Watanobe, Y., 2019. From Code to Ratings – Enhancing Collaborative Filtering in OJ Systems. *Proc.*, Volume 7, 31 December 2019. Available at: https://ieeexplore.ieee.org/document/10813164 .

[7] Spinellis, D., Louridas, P., Gousios, G. and Chatzigiannakis, I., 2019. Using Research Literature to Generate Datasets of Implicit Feedback (LIBRETTI). *Proc.*, Volume 7, 23 December 2019. Available at: https://ieeexplore.ieee.org/document/8924687 .

[8] El-Gayar, M.M., Mekky, N.E., Atwan, A. and Soliman, H., 2019. Enhanced Search Engine Using Proposed Framework and Ranking Algorithm Based on Semantic Relations. *Proc.*, Volume 7,7 October 2019.Available at:
https://ieeexplore.ieee.org/document/8840848 .

[9] Saif, I., Doukkali, A.S. and Enaanai, A., 2017. GENAUM: New Semantic Distributed Search Engine. *Mobile Multimedia*, 12(3&4), pp.210-221. Available at: https://ieeexplore.ieee.org/document/11079888 .

[10] Killoran, J.B., 2013. How to Use Search Engine Optimization Techniques to Increase Website Visibility. *Proc.*, Volume 56, Issue 1, March 2013. Available at: https://ieeexplore.ieee.org/document/6463486 .

Base Paper

Yadav, U. and Duhan, N., 2021. Efficient retrieval of data using semantic search engine based on NLP and RDF. *Journal of Web Engineering*, *20*(8), pp.2285-2318. Available at: https://ieeexplore.ieee.org/document/10246838.

Appendix

1. Data Sheets

Since the project is completely s software-only implementation, hardware datasheets are not applicable for the project. However, the **software specifications and APIs used** are details summarized below:

Table A.1 Datasheets

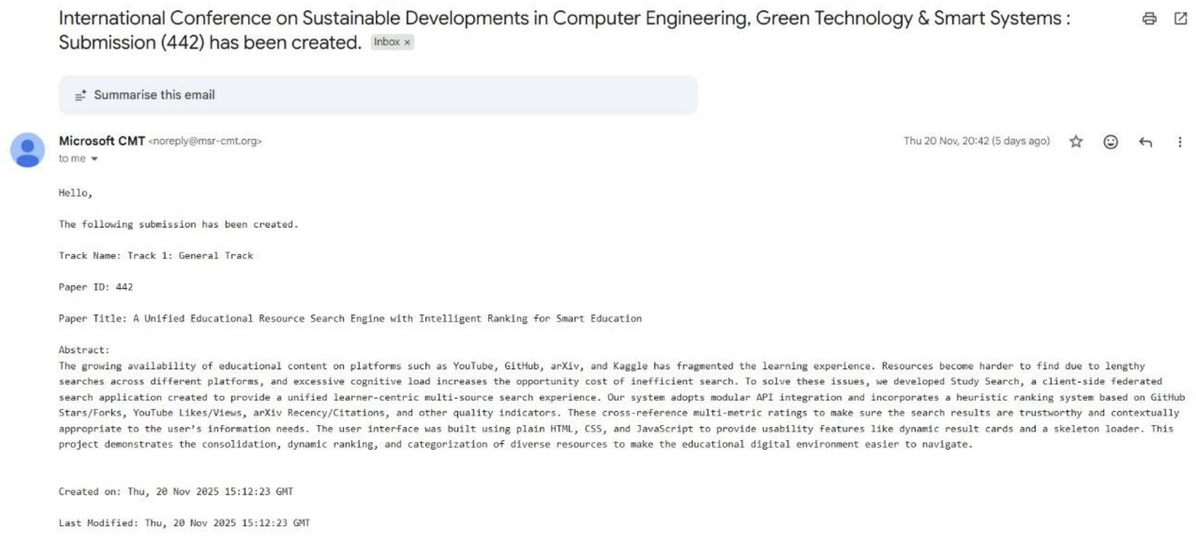
Component	**Description**
YouTube API	Fetches videos based on search queries; returns metadata such as title, description, channel name, view count, and likes. Documentation: YouTube Data API v3
GitHub API	Retrieves repository information, including stars, forks, description, and owner details. Documentation: GitHub REST API
arXiv API	Provides research paper metadata such as title, abstract, publication date, and link. Documentation: arXiv API User Manual
Kaggle API	Fetches datasets metadata including downloads, usability ratings, and links to datasets. Documentation: Kaggle API
Web Technologies	HTML5, CSS3, JavaScript ES6, and GitHub Pages for deployment. Responsive design ensured across devices.

2. Publications

As this project is an academic project submission:

- Submitted the paper to International Conference on Sustainable Developments in Computer Engineering, Green Technology & Smart Systems that will be held on 20/12/2025.

- **Email of Paper Submission:**



International Conference on Sustainable Developments in Computer Engineering, Green Technology & Smart Systems : Submission (442) has been created.

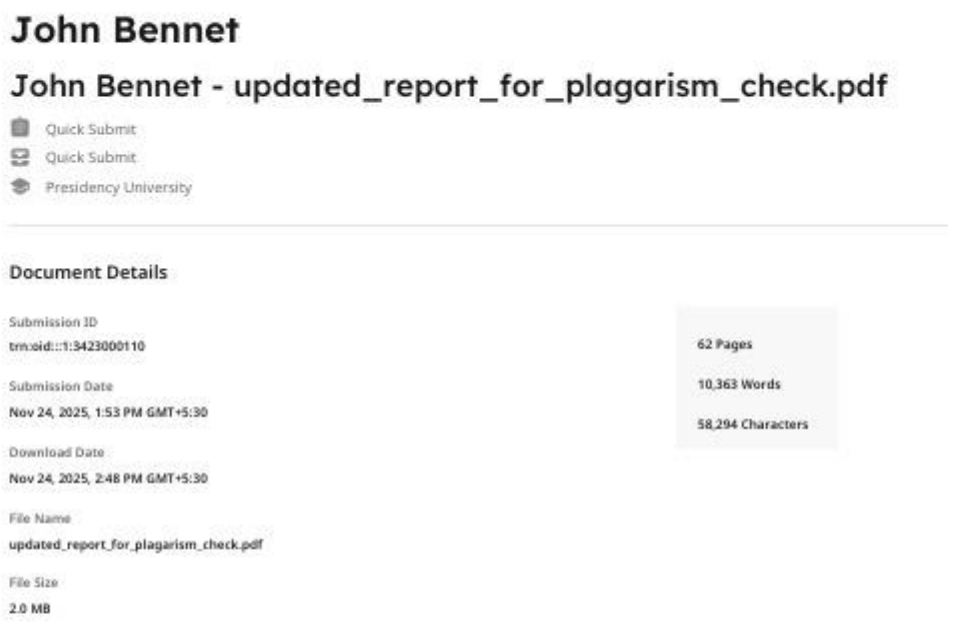
Microsoft CMT <noreply@msr-cmt.org>
to me

Thu 20 Nov, 20:42 (5 days ago)

```
Hello,

The following submission has been created.

Track Name: Track 1: General Track

Paper ID: 442

Paper Title: A Unified Educational Resource Search Engine with Intelligent Ranking for Smart Education

Abstract:
The growing availability of educational content on platforms such as YouTube, GitHub, arXiv, and Kaggle has fragmented the learning experience. Resources become harder to find due to lengthy
searches across different platforms, and excessive cognitive load increases the opportunity cost of inefficient search. To solve these issues, we developed Study Search, a client-side federated
search application created to provide a unified learner-centric multi-source search experience. Our system adopts modular API integration and incorporates a heuristic ranking system based on GitHub
Stars/Forks, YouTube Likes/Views, arXiv Recency/Citations, and other quality indicators. These cross-reference multi-metric ratings to make sure the search results are trustworthy and contextually
appropriate to the user's information needs. The user interface was built using plain HTML, CSS, and JavaScript to provide usability features like dynamic result cards and a skeleton loader. This
project demonstrates the consolidation, dynamic ranking, and categorization of diverse resources to make the educational digital environment easier to navigate.


Created on: Thu, 20 Nov 2025 15:12:23 GMT

Last Modified: Thu, 20 Nov 2025 15:12:23 GMT
```

Fig A.1 Email of Paper Submission

3. Project Similarity Report



John Bennet

John Bennet - updated_report_for_plagarism_check.pdf

Quick Submit
Quick Submit
Presidency University

Document Details

Submission ID
trn:oid:::1:3423000110

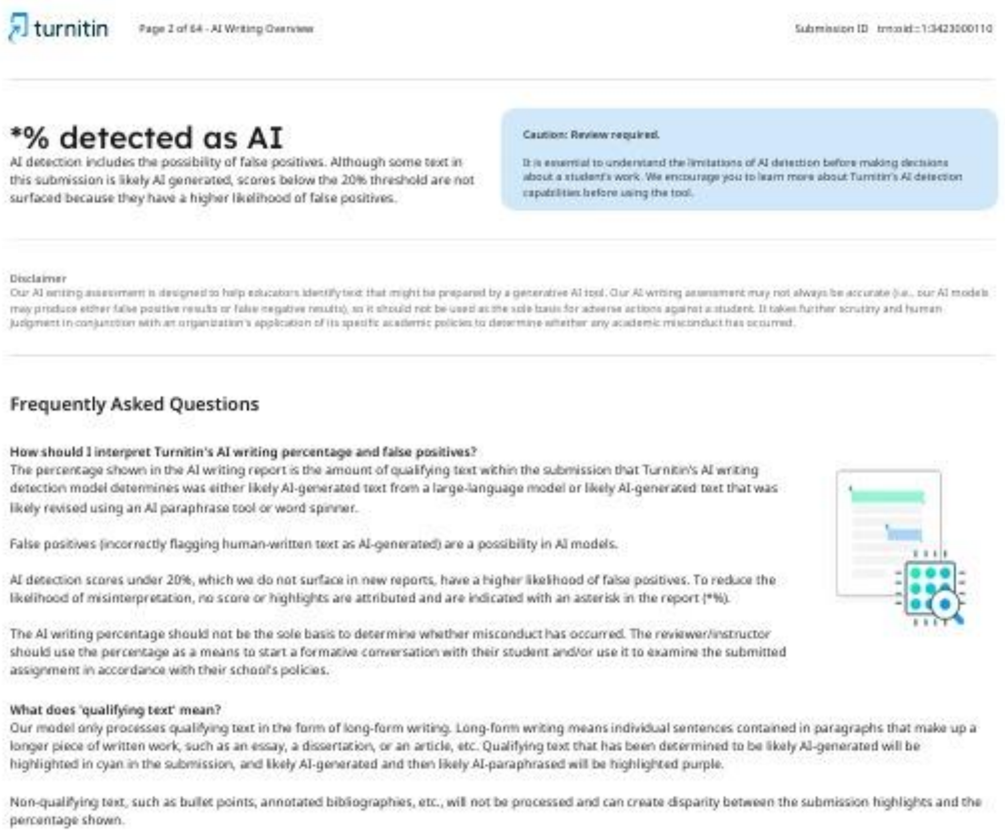
Submission Date
Nov 24, 2025, 1:53 PM GMT+5:30

Download Date
Nov 24, 2025, 2:48 PM GMT+5:30

File Name
updated_report_for_plagarism_check.pdf

File Size
2.0 MB

62 Pages
10,363 Words
58,294 Characters



turnitin Page 2 of 64 - AI Writing Overview Submission ID trn:oid:::1:3423000110

***% detected as AI**

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

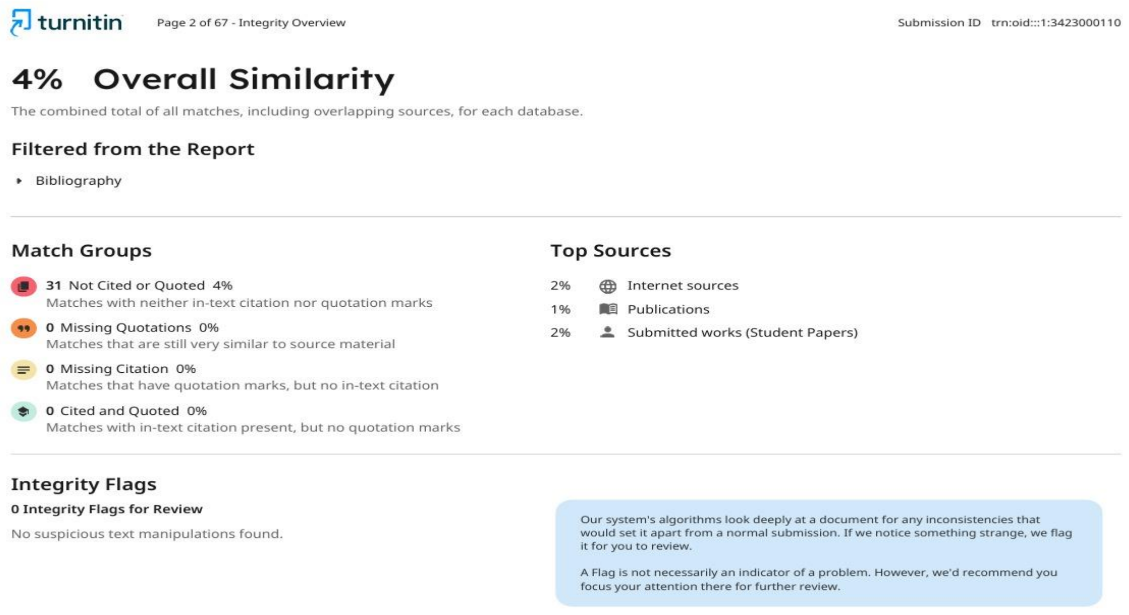
AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

Fig A.2 Plagiarism Check Report



turnitin Page 2 of 67 - Integrity Overview Submission ID trn:oid:::1:3423000110

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- **31** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
1%	Publications
2%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Fig A.3 Similarity Check Report.

4. Project Demonstration Images

Includes screenshots explaining the different scenarios:

- **Home Page / Search Interface:** Displays the search bar and the different categories of the tabs provides by the search engines.

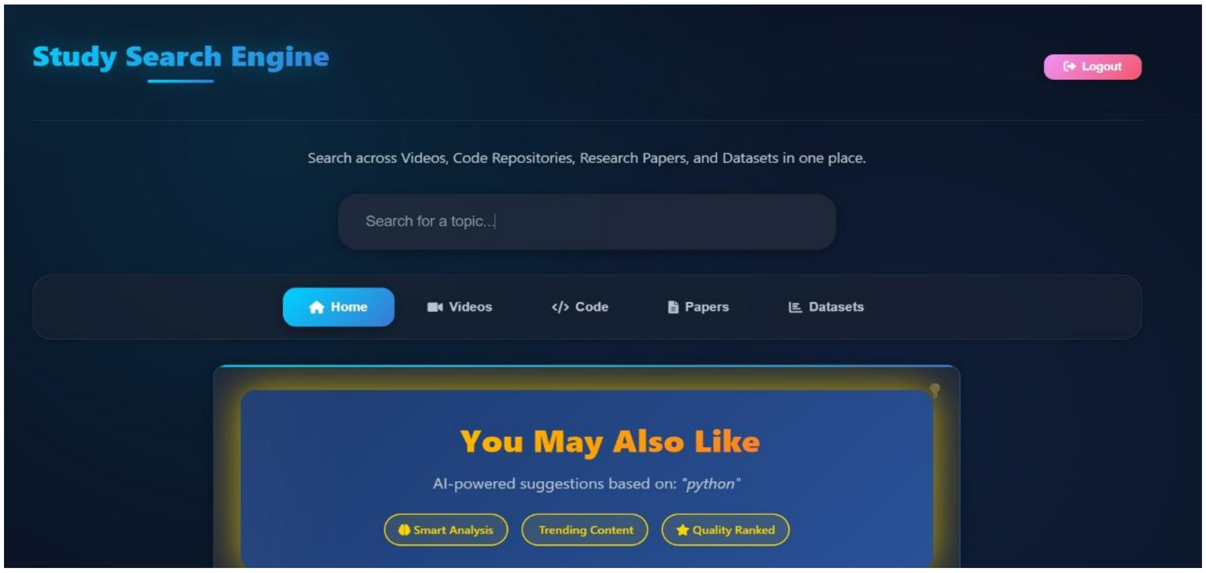
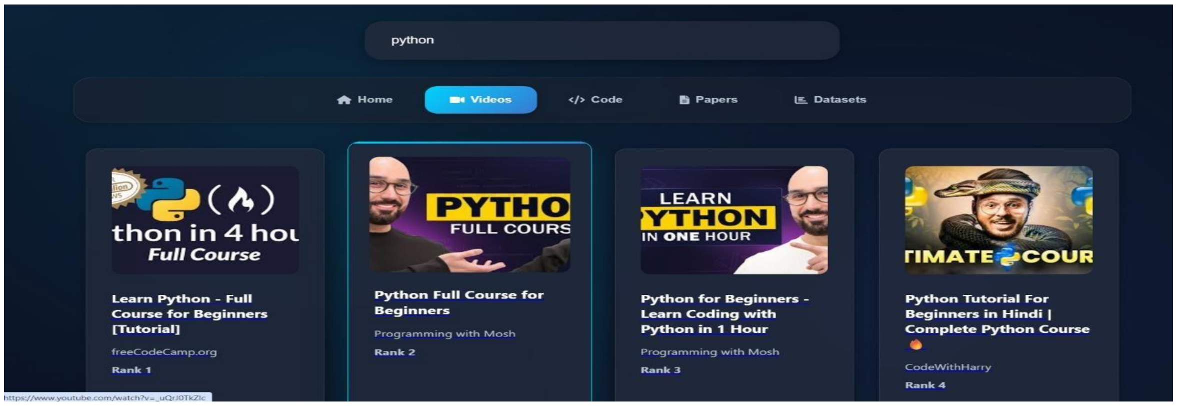


Fig A.4 Home Page

- **Video Search Results:** Displays top YouTube videos for a query based on the ranking algorithm.



Learn Python - Full Course for Beginners [Tutorial]
freeCodeCamp.org
Rank 1

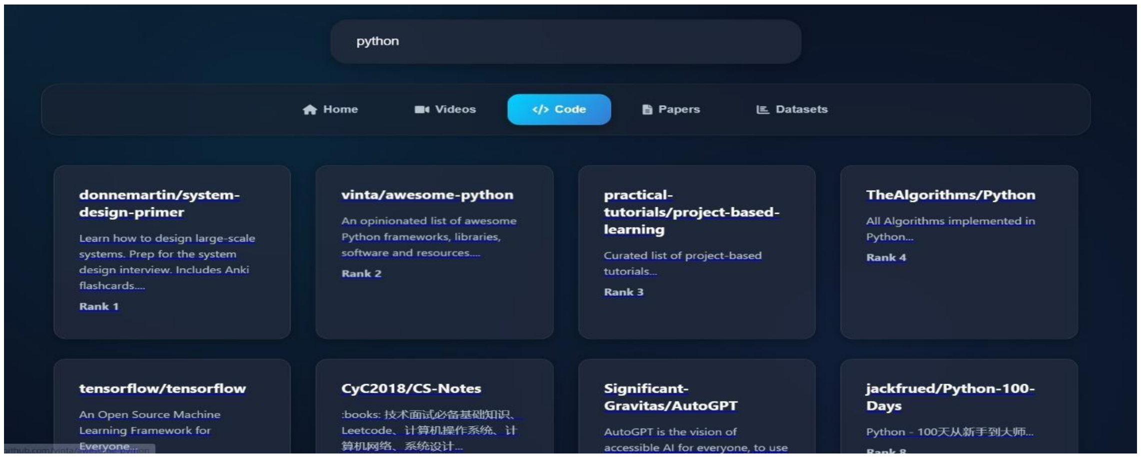
Python Full Course for Beginners
Programming with Mosh
Rank 2

Python for Beginners - Learn Coding with Python in 1 Hour
Programming with Mosh
Rank 3

Python Tutorial For Beginners in Hindi | Complete Python Course
CodeWithHarry
Rank 4

Fig A.5 Video Search Results

- **Code Search Results:** Lists the top most GitHub repositories with their ranks.



donnemartin/system-design-primer
Learn how to design large-scale systems. Prep for the system design interview. Includes Anki flashcards....
Rank 1

vinta/awesome-python
An opinionated list of awesome Python frameworks, libraries, software and resources....
Rank 2

practical-tutorials/project-based-learning
Curated list of project-based tutorials...
Rank 3

TheAlgorithms/Python
All Algorithms implemented in Python...
Rank 4

tensorflow/tensorflow
An Open Source Machine Learning Framework for Everyone...

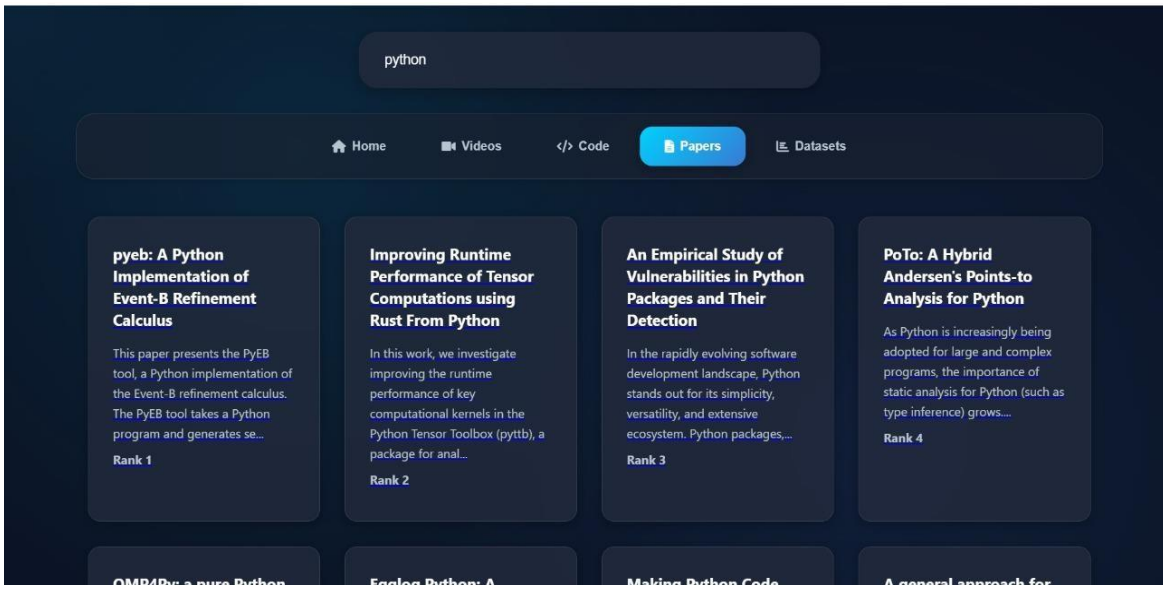
CyC2018/CS-Notes
:books: 技术面试必备基础知识、Leetcode、计算机操作系统、计算机网络、系统设计...

Significant-Gravitas/AutoGPT
AutoGPT is the vision of accessible AI for everyone, to use

jackfrued/Python-100-Days
Python - 100天从新手到大师...

Fig A.6 Code Search Results

- **Research Paper Results:** Displaying latest papers from arXiv.



pyeb: A Python Implementation of Event-B Refinement Calculus
This paper presents the PyEB tool, a Python implementation of the Event-B refinement calculus. The PyEB tool takes a Python program and generates se...
Rank 1

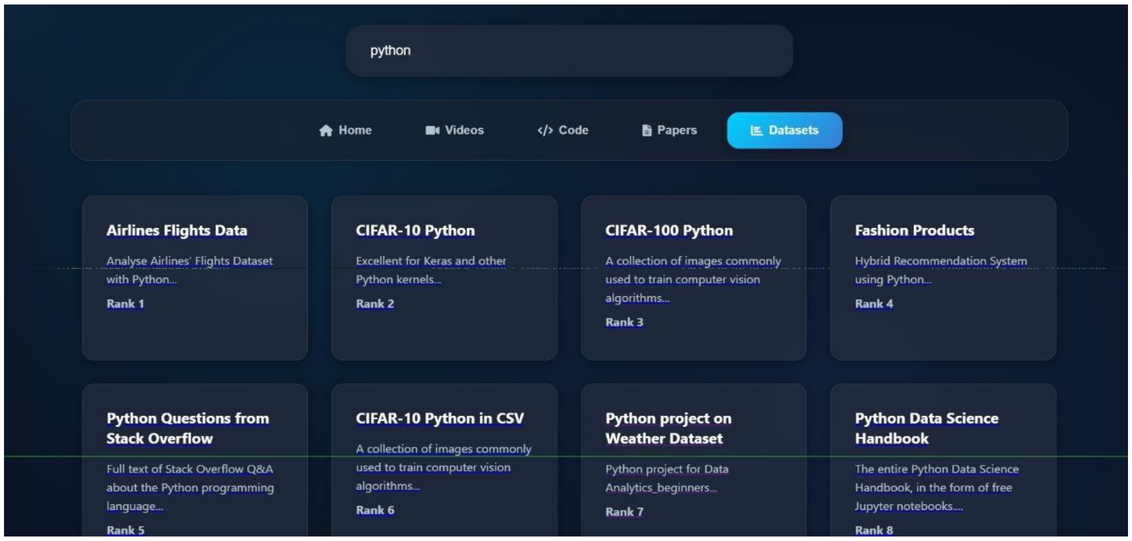
Improving Runtime Performance of Tensor Computations using Rust From Python
In this work, we investigate improving the runtime performance of key computational kernels in the Python Tensor Toolbox (pyttb), a package for anal...
Rank 2

An Empirical Study of Vulnerabilities in Python Packages and Their Detection
In the rapidly evolving software development landscape, Python stands out for its simplicity, versatility, and extensive ecosystem. Python packages,...
Rank 3

PoTo: A Hybrid Andersen's Points-to Analysis for Python
As Python is increasingly being adopted for large and complex programs, the importance of static analysis for Python (such as type inference) grows....
Rank 4

Fig A.7 Research Paper Results

- **Dataset Results:** Displays the ranked datasets from Kaggle.



Airlines Flights Data
Analyse Airlines' Flights Dataset with Python...
Rank 1

CIFAR-10 Python
Excellent for Keras and other Python kernels...
Rank 2

CIFAR-100 Python
A collection of images commonly used to train computer vision algorithms...
Rank 3

Fashion Products
Hybrid Recommendation System using Python...
Rank 4

Python Questions from Stack Overflow
Full text of Stack Overflow Q&A about the Python programming language...
Rank 5

CIFAR-10 Python in CSV
A collection of images commonly used to train computer vision algorithms...
Rank 6

Python project on Weather Dataset
Python project for Data Analytics_beginners...
Rank 7

Python Data Science Handbook
The entire Python Data Science Handbook, in the form of free Jupyter notebooks....
Rank 8

Fig A.8 Dataset Results

- **Loading Skeleton Cards:** Demonstrating a better user experience during API calls.

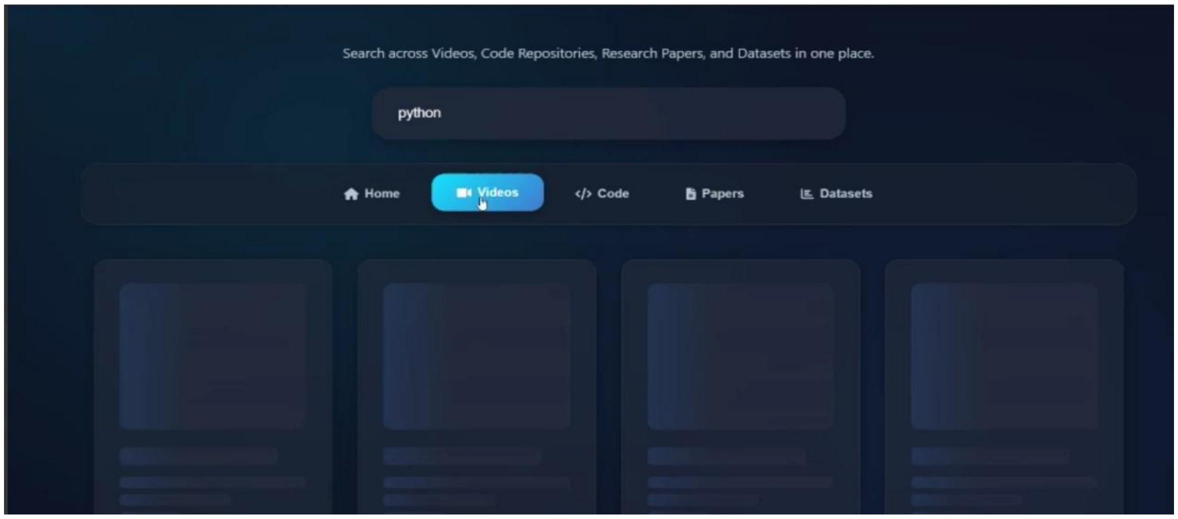

Search across Videos, Code Repositories, Research Papers, and Datasets in one place.

Fig A.9 Loading Skeleton Cards