Innovation, Research and Development (IRD) Website Development

Vickery Kumar - 2021299
Suyash Kumar- 2021293
Pourav Surya- 2021271

April 29, 2024

BTP report submitted in partial fulfillment of the requirements for the Degree of B.Tech. in Computer Science & Engineering

BTP Track: Engineering
BTP Advisor: Dr. Puspendra Singh

Indraprastha Institute of Information Technology, New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled "Innovation, Research and Development (IRD) IIITD Website Development" submitted by me for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of Dr. Puspendra Singh. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

Vickey Kumar **Place & Date: 29-04-2024**
Suyash Kumar
Pourav Surya

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Puspendra Singh **Place & Date: 29-04-2024**
...**Advisor**

Abstract

Abstract

Our project offers a specialized platform for IIIT Delhi Institute, streamlining research paper management within the IRD department. It empowers professors, students, and other stakeholders to seamlessly oversee their research endeavors by offering functionalities such as research paper creation, editing, deletion, and content management. Users can easily access and modify essential paper details including author information, publishers, and titles, year and organization within the database. Our project aims to enhance the research paper management efficiency, as user can easily collect all the research done by any author at IIIT Delhi.

Keywords: Dashboard, Table, Upload citation.txt, Form, Editing and upgradation / Delete, Search the keywords

Acknowledgments

I express my deepest gratitude to Pushpendra Singh (Professor, IIIT Delhi) for providing me this opportunity and their guidance throughout the project. Their continuous guidance and expertise have taught me a lot and helped in the project. I am also grateful to Kapil Dev Garg (IRD Manager) for guiding us throughout the project. I would also like to extend my gratitude to all the college authorities that helped by providing various resources required for the project.

Contents

Chapter 1

1 Introduction

The Innovation Research Development (IRD) department at IIIT-D handles operations related to sponsored research projects, consultancy jobs, and other RD related activities. We have developed a centralized online platform designed to streamline the management of research papers and associated data. This centralized system allows for the storage of comprehensive information related to research documents, including authors' names, paper titles, journals, and more. Key features of this platform include secure authentication, the ability to update existing records, and the functionality to add or delete research papers as needed. It supports multiple users, ensuring that individual contributors can only modify data they have uploaded, while administrators have the authority to edit any information across the system. Additionally, our platform offers a robust search capability, allowing users to efficiently locate data based on various criteria such as authors, publishers, and other relevant fields. This integration of advanced features aims to enhance accessibility and management of academic research documentation efficiently.

Chapter 2

2 Problems

- **Data Fragmentation:** Research data was scattered across multiple platforms, making it difficult to access and manage.

- **Inconsistent Access Controls:** There was no unified system for managing user permissions, leading to potential security risks.

- **Limited Search Functionality:** Searching for specific papers or authors was cumbersome and often inefficient.

- **Manual Data Management:** Updating, adding, or deleting information required manual intervention, which was time-consuming and error-prone.

- **Limited collaboration opportunities:** Lack of a research paper sharing platform hampers collaboration among students, professors, and faculty by limiting awareness of ongoing projects and findings.

- **Lack of transparency:** Users lacked access to their peers' research projects and publications, which restricted knowledge sharing, reduced awareness of the latest research developments, and hindered collaboration within the academic community.

- **Inefficient paper management:** Users faced difficulties in managing their own research papers, including organizing them, updating information, or making revisions, leading to potential confusion and duplication of efforts.

Chapter 3

3 Architecture

3.1 Technologies Used

- Client-Side: HTML, CSS, Bootstrap, JavaScript, EJS (Embedded JavaScript)
- Server-Side (Web Server): Python, Node.js with modules
- Database: MongoDB

3.2 Stack Overview:

- Our website, the Research Paper Management System, is built on a sophisticated and modern technology stack designed to offer a seamless and engaging user experience.

- On the client side, we utilize a combination of HTML, CSS, Bootstrap, JavaScript, and EJS (Embedded JavaScript) to create dynamic and visually appealing interfaces. HTML provides the structural foundation of our web pages, enhanced by CSS and Bootstrap for styling and layout, ensuring a consistent appearance across different devices and screen sizes. JavaScript adds interactivity, enabling dynamic content updates and enhancing user engagement. EJS assists in server-side templating, facilitating the dynamic generation of HTML content.

- On the server side, our platform leverages the strengths of both Python and Node.js to manage a wide range of server functionalities effectively. Node.js, with its essential modules, serves as the core of our server infrastructure, allowing us to handle HTTP requests, manage routing, and execute business logic accurately. Express.js, a powerful web application framework for Node.js, acts as the cornerstone of our server-side architecture, offering a comprehensive set of features for building RESTful APIs, integrating middleware, and performing server-side rendering.

- For our database needs, we have chosen MongoDB, known for its adaptability, scalability, and performance. As a NoSQL database, MongoDB adeptly manages both structured and unstructured data, making it particularly suitable for handling the varied data types involved in research papers, user profiles, and system configurations. Its document-oriented model supports flexible, schema-less data storage, promoting rapid development and iteration. Additionally, MongoDB's distributed architecture and its capabilities for replication and sharding provide high availability, fault tolerance, and scalability, making it an excellent choice to meet the growing demands of our application.

3.3 Module Overview:

- Within our system architecture, we incorporate various modules to facilitate seamless interaction between Node.js and MongoDB, ensuring efficient data management and retrieval.

- Primarily, we employ Pymongo, a Python-based driver for MongoDB, to establish a robust connection between our Node.js backend and the MongoDB database. This integration allows us to execute database operations, such as querying, inserting, updating, and deleting data, directly from our Node.js environment, leveraging the functionality and flexibility of MongoDB.

- In addition to Pymongo, we leverage other MongoDB-related libraries and modules to enhance our system's capabilities and streamline development processes. These auxiliary libraries complement Pymongo by providing additional functionalities and utilities tailored to specific requirements within our application. Whether it's handling complex data manipulations, implementing advanced querying techniques, or optimizing database performance, these supplementary MongoDB libraries play a vital role in augmenting our system's functionality and ensuring seamless data management operations.

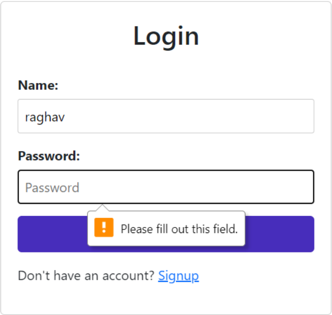
- By combining Pymongo and other MongoDB-related libraries within our system architecture, we establish a robust foundation for interacting with MongoDB from our Node.js environment. This cohesive integration enables us to harness the full potential of MongoDB's features and capabilities, empowering our application to efficiently manage data, scale seamlessly, and deliver a superior user experience.

Chapter 4

4 Workflow & Website Development

4.1 Login / Sign up

The user login process involves authentication through MongoDB. Users create a username and password, which are stored securely. When attempting to log in, the system checks if the entered credentials match those stored in the database. If the credentials match, the user is granted access. If not, they are prompted to re-enter the correct username and password. This authentication mechanism ensures that only authorized users can access the system's features and resources.



Login

Name: raghav

Password: Password

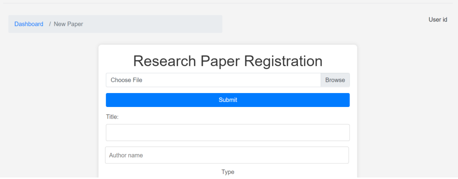
Please fill out this field.

Don't have an account? Signup

4.2 Add New Entries

Users have the option to upload and add a new research paper manually or upload a citation file containing metadata. The citation file should follow a specific format, which includes essential information about the research papers. This format ensures consistency and accuracy in the metadata provided. By offering these two methods, users can conveniently add their research papers to the system, whether they prefer manual entry or bulk upload via citation files. The citation format is the following:

@articleguo2008using, title=Using immersive video to evaluate future traveller information systems, author=Guo, Amy Weihong and Blythe, Phil and Olivier, Patrick and Singh, Pushpendra and Nam, Ha and others, journal=IET Intelligent Transport Systems, year=2008, publisher=Newcastle University



Title:

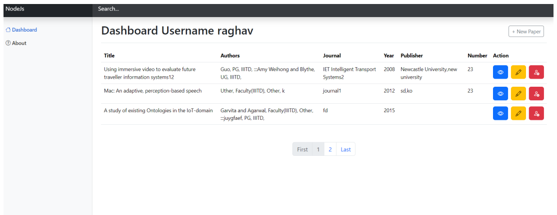
Author name

Type

When a user submits a citation file, the system parses all the data and displays it for review. Users have the opportunity to edit the information before final submission to the database. Similarly, during registration, users can fill out a simple form with details of the research paper or edit the data from an uploaded citation.txt file. On the backend, the file upload triggers the execution of the main.py Python code. This code reads the uploaded file, parses its contents, converts them into JSON format, and then pushes the data into the MongoDB database. This process ensures that the metadata from the citation file is accurately captured and stored in the system for further management and retrieval.

4.3 Table

The table is generated by retrieving data from MongoDB based on the username of the user stored in the database. In this table, three actions are available to the user: viewing, editing, and deleting table rows.

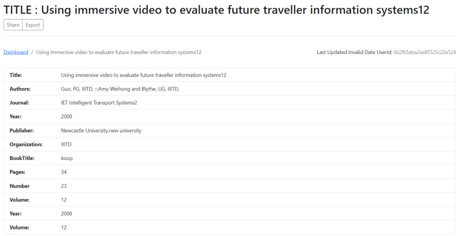


Dashboard Username raghav

Title	Authors	Journal	Year	Publisher	Number	Action
Using immersive video to evaluate future traveller information systems12	Guo, PG, IIITD, :::Amy Weihong and Blythe, UG, IIITD,	IET Intelligent Transport Systems2	2008	Newcastle University,new university	23	
Mac: An adaptive, perception-based speech	Uther, Faculty(IIITD), Other, k	journal1	2012	sd,ko	23	
A study of existing Ontologies in the IoT-domain	Garvita and Agarwal, Faculty(IIITD), Other, :::juygfaef, PG, IIITD,	fd	2015			

4.3.1 View the data

Data is retrieved from MongoDB using the "get" option through the view function, and then displayed in table format.



TITLE : Using immersive video to evaluate future traveller information systems12

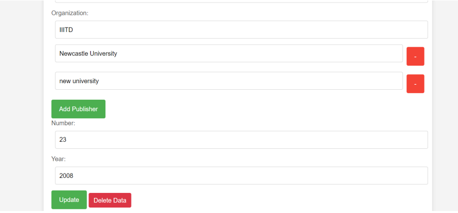
Dashboard / Using immersive video to evaluate future traveller information systems12

Last Updated Invalid Date UserId: 662fb5aba2ea85525c20a524

Title:	Using immersive video to evaluate future traveller information systems12
Authors:	Guo, PG, IIITD, :::Amy Weihong and Blythe, UG, IIITD,
Journal:	IET Intelligent Transport Systems2
Year:	2008
Publisher:	Newcastle University,new university
Organization:	IIITD
BookTitle:	koop
Pages:	34
Number	23
Volume:	12
Year:	2008
Volume:	12

4.3.2 Edit the Row in the Table

The data of a specific row is converted into a form format, allowing the user to easily perform the "Put" option through a MongoDB query. This process ensures that data updates are made accurately with the correct author attribution.



Organization:

IIITD

Newcastle University

new university

Number:

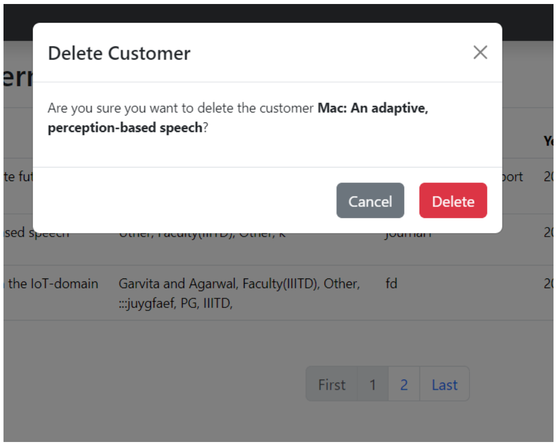
23

Year:

2008

4.3.3 Delete the particular Row

The ID of the specific data is passed to the query through the delete function to remove the paper. A confirmation notification appears beforehand, ensuring user confirmation. Upon confirmation, the paper data is removed from MongoDB through the deletion option.



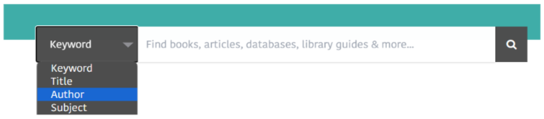
Delete Customer

Are you sure you want to delete the customer **Mac: An adaptive, perception-based speech**?

×
Cancel
Delete
the IoT-domain
Garvita and Agarwal, Faculty(IIITD), Other, :::juygfaef, PG, IIITD,
fd
First
1
2
Last

4.3.4 Searching by keywords

We search data in MongoDB by utilizing keywords related to the model used. Through the JavaScript function "Search," the respective data is queried from the database, and only relevant data is displayed based on the search criteria. For example Author Name : Raghav



Chapter 5

5 Implementation

5.1 Implementation Details

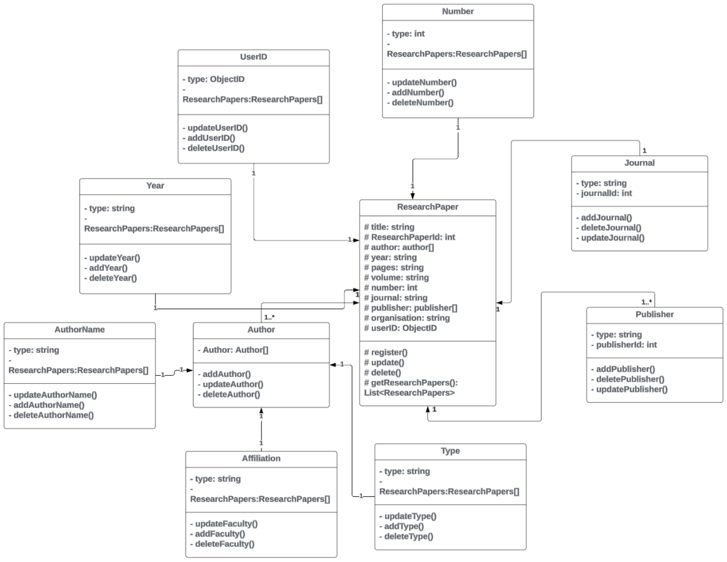


Figure 5.1: The new hierarchy of models for Research Paper Management

- Upon successful login, users are granted access to upload the metadata of their research papers into the College Research Papers Management System. The metadata includes essential information about the paper, such as author name, publisher, publication year, affiliation, type of publication, issue number (if applicable), and journal details.

- Each user's uploaded metadata is associated with their unique user ID, ensuring proper attribution and organization within the system. This association allows users to manage and retrieve their uploaded metadata efficiently while maintaining data integrity and security.

- The metadata is stored in a MongoDB database, utilizing a structured schema to ensure consistency and ease of retrieval. The database design facilitates efficient querying and retrieval of metadata based on various parameters, enabling users to search and access their research papers seamlessly.

- By incorporating this functionality into the system, users can easily catalog and manage the metadata of their research papers, facilitating better organization and retrieval of academic materials. This feature enhances the user experience and contributes to the overall effectiveness of the College Research Papers Management System in facilitating academic research endeavors.

Chapter 6

6 Future Plan

For the future development of our online platform for managing research papers, we have outlined several key areas of focus to enhance functionality, security, and user experience.

6.1 Security Enhancements

Advanced Firewall Implementation: Integrate IPTables or NFTables for robust server protection against various network-based DOS attacks. Enhanced Authentication: Expand our authentication framework by implementing and testing an LDAP authentication system, automating user additions and improving security. Secure File Transfer: Transition from FTPS to SFTP for safer and more secure file transfers between backend and frontend servers, despite potential trade-offs in speed.

6.2 Deployment Strategy

Staged Deployment: Plan for a phased rollout, initially deploying a beta version to a selected user base to gather feedback and perform live environment testing. Scalability Plans: Prepare infrastructure to scale efficiently as user base and data volume grow post-launch.

6.3 DevOps Improvements

Expanded Testing Protocols: Develop a comprehensive testing suite that includes integration, functional, and security testing to ensure robustness and reliability. Enhanced Logging and Monitoring: Augment the existing admin dashboard to include more detailed analytics, such as server usage and crash reports, to proactively manage system health.

6.4 Codebase and Technology Upgrades

Robust Form Handling: Utilize Django's Forms and FormViews for automated input sanitation, streamlining and securing the process of accepting user data. Framework Updates: Consider general improvements in the user interface, potentially updating or replacing existing frameworks (like Bootstrap) with more modern solutions such as Tailwind CSS for enhanced flexibility and customization.

6.5 User Interface and Usability Enhancements

Notifications System Overhaul: Completely redesign the notifications system to ensure functionality and user-friendliness. UI Modularization: Reduce redundancy by modularizing common UI elements across pages, such as file attachments, to improve maintainability and consistency. Component-Based Architecture: Transition to a component-based UI architecture, possibly integrating with tools like Storybook to facilitate development and visualization of UI components.

These planned upgrades and enhancements aim to solidify the platform's foundation while expanding its capabilities to better serve the needs of the users.