

# NCF RESEARCH NEXUS

A Research Repository for Naga  
College Foundation, Inc.

Arnel E. Almario Jr.  
and  
John Rey S. Tolosa



# Purpose of the Project

---

- Promote Academic Excellence and Innovation
- Enhancing Visibility and Recognition
- Supporting Teaching and Learning
- Knowledge Preservation and Sharing

# OBJECTIVES

- To create a centralized platform for storing and accessing research papers, articles, and resources pertaining to Non-Cognitive Factors in education.
- To facilitate collaboration and knowledge exchange among researchers, educators, and policymakers interested in NCF research.
- To promote awareness and understanding of the importance of NCF in shaping student academic achievement and holistic development.
- To contribute to the advancement of educational theory and practice by providing a comprehensive repository of NCF-related literature and empirical studies.

# Scope

---

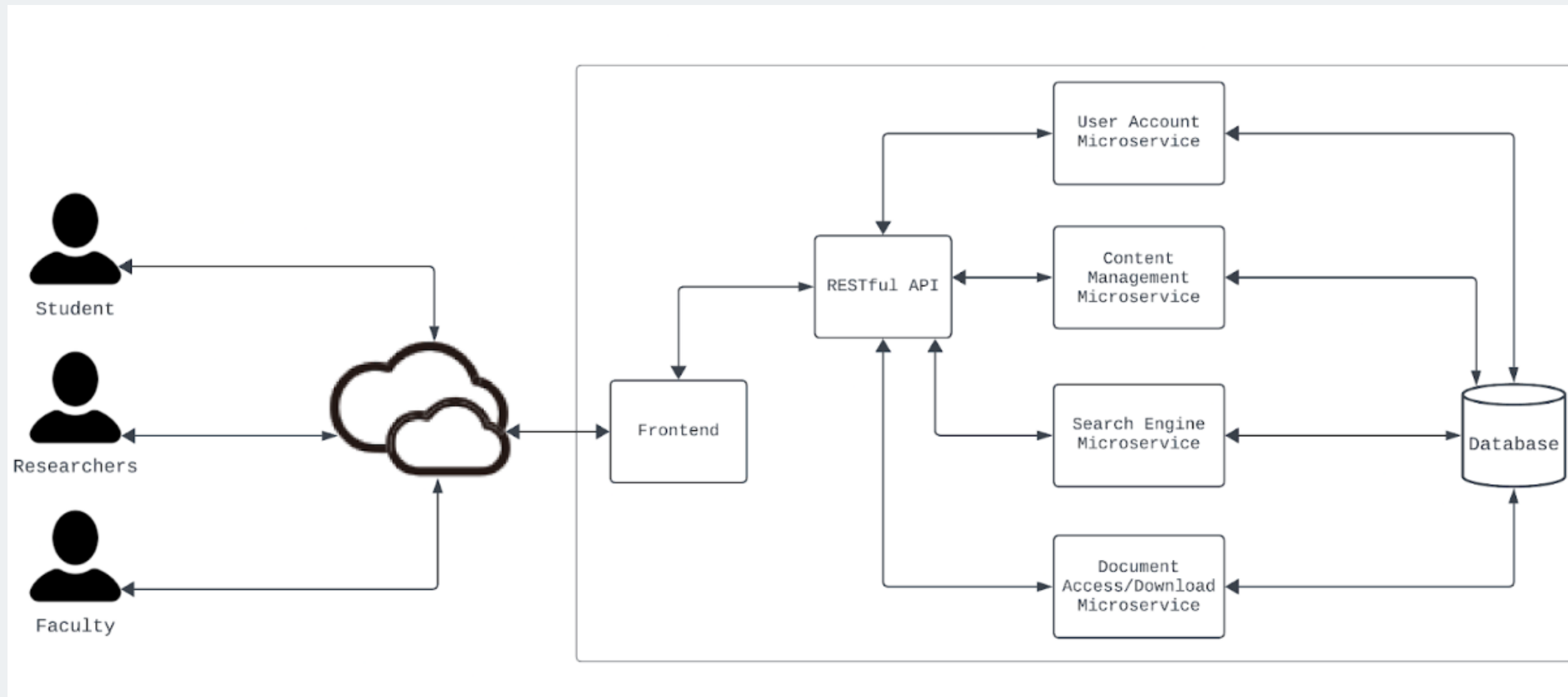
- Document Management
- Search and Discovery
- Access Control and Permissions
- Metadata Management
- Version Control
- Compliance and Preservation
- Usage Analytics and Reporting

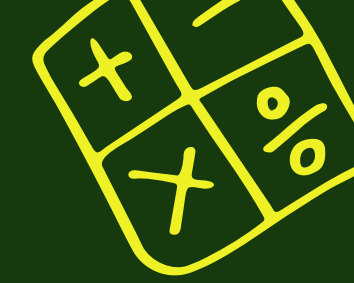
# Delimitation

---

- Types of Content
- Copyright and Licensing
- Non-academic materials
- User Access

# System Architecture

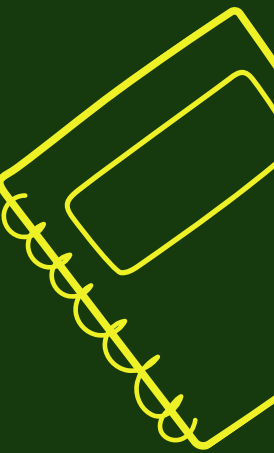




# Algorithm

## Fuzzy Search/Matching Algorithm

The proposed system is anchored into one algorithm: Fuzzy Search/Matching Algorithm for the content searching algorithm. This algorithm returns a list of results based on likely relevance even though search argument words and spellings may not be an exact match. It consists of several algorithms that work together to produce accurate and fast results. Examples are N-gram, and Levenshtein distance. Overall, the algorithm browses datasets looking for strings matched to the user query and marks each with relevance score. This relevance score would determine the sorting/ranking of results. However, the one that would be used in the study is a hybrid approach and compiled into one library called FuzzyWuzzy. This type of algorithm would be suitable for one of the key features of NCF Research Nexus, which is the full-text typographical error-safe search engine. It also displays any suggested relevant results based on the user's query.



# Workflow and Components

consists of several components in its backend to handle user login, search, upload, and download functionalities.

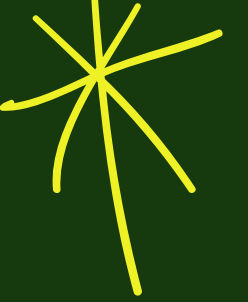
## User Login:

- ***Authentication Service:*** Handles user authentication and authorization. This service verifies user credentials (username/email and password) against a database of registered users.
- ***Token Generation:*** Upon successful authentication, the authentication service generates a token (e.g., JSON Web Token, JWT) containing user information and permissions. This token is sent to the client and used for subsequent requests to authenticate the user.



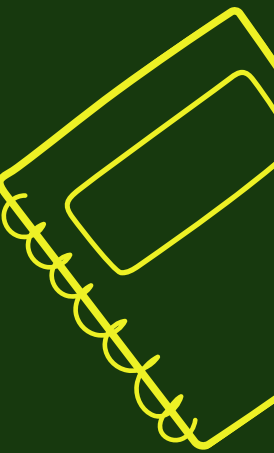


# Workflow and Components

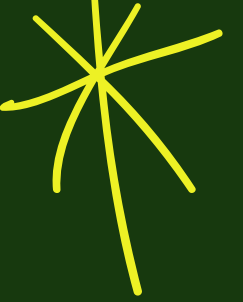


## Searching:

- ***Search Engine:*** Manages indexing and searching of research documents. It crawls, indexes, and organizes documents based on metadata (e.g., title, author, keywords) and content.
- ***Search API:*** Provides an interface for users to perform search queries. Users can enter search terms, filters, and other parameters to retrieve relevant documents.
- ***Query Processing:*** The search engine processes user queries, matches them against indexed documents, and ranks the results based on relevance.
- ***Results Presentation:*** The search API returns search results to the reactJS, typically in JSON format. The reactJS interface displays the results to the user.

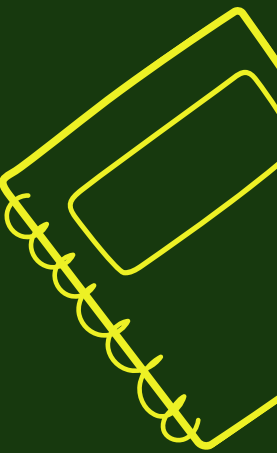


# Workflow and Components



## Uploading:

- ***Upload Service:*** Handles file uploads from users. This service validates file formats, checks permissions, and stores uploaded files in a storage system (e.g., file system, cloud storage).
- ***Metadata Extraction:*** Extracts metadata from uploaded documents (e.g., title, author, publication date) and stores it in a database.
- ***Query Processing:*** The search engine processes user queries, matches them against indexed documents, and ranks the results based on relevance.
- ***Indexing:*** The search engine indexes uploaded documents, making them searchable for other users.



# Workflow and Components



Downloading:

- ***Download Service:*** Handles requests to download documents. This service verifies user permissions and retrieves the requested document from storage.
- ***Access Control:*** Ensures that only authorized users can download documents. Users may need to meet certain criteria (e.g., subscription status, institutional affiliation) to access certain documents.
- ***Download Logging:*** Logs download activities for auditing and analytics purposes.



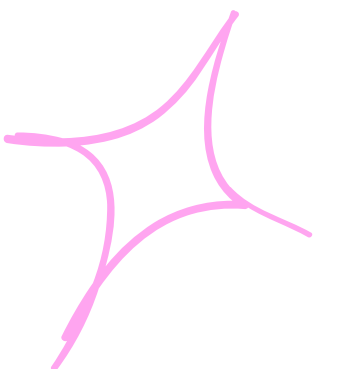
## DATABASE SCHEMA

### Roles Table:

- This is to store information about different roles researchers might have.
- This table is related to the Users table through a foreign key relationship if researchers have specific roles assigned to them.

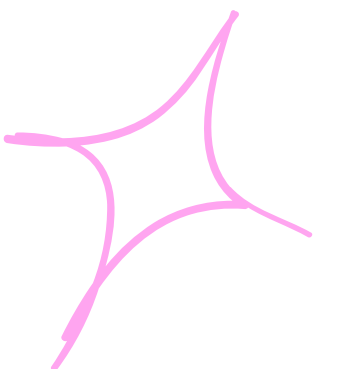
### Users Table:

- Contains information about individual researchers, such as their unique identifier (user\_id), name (user\_name), email, password, and role\_id .
- The **user\_id** serves as the primary key, uniquely identifying each researcher in the table.
- The **role\_id** columns in this table serve as foreign keys referencing the corresponding primary keys in the Roles Table, establishing relationships between role and users.



## Publications Table:

- Stores information about research publications authored by users.
- Each publication has a unique identifier (`publication_id`) and includes details such as the publication title (`publication_title`), publication date, publication type (`publication_type`), and references to both the researcher (`user_id`) and the research project (`project_id`) associated with the publication.
- The `user_id` and `project_id` columns in this table serve as foreign keys referencing the corresponding primary keys in the Researchers and Research Projects tables, establishing relationships between researchers, projects, and publications.



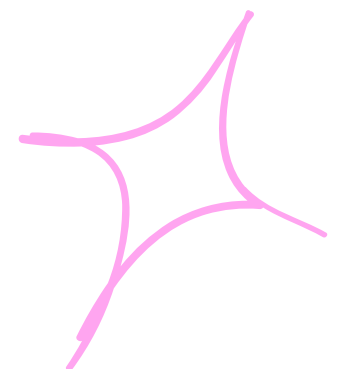
## DATABASE SCHEMA

### Projects Table:

- Stores details about research projects conducted by researchers.
- Each project is identified by a unique `project_id` and includes information such as the project title (`project_title`), description, start date, end date, and the `user_id` of the researcher leading the project.
- The `user_id` column in this table serves as a foreign key referencing the `user_id` column in the Researchers table, establishing a one-to-many relationship between researchers and their projects.

### Keywords Table:

- Manages research keywords or tags used to categorize publications.
- Each keyword is assigned a unique identifier (`keyword_id`) and includes the actual keyword text (`keyword`).



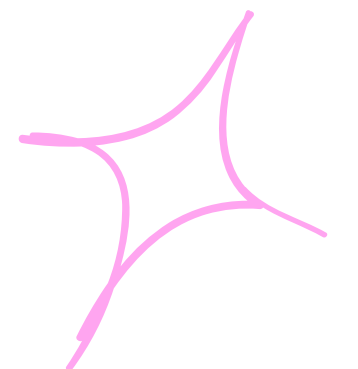
## DATABASE SCHEMA

### Publications\_Keywords Table:

- Represents a many-to-many relationship between publications and keywords.
- Each record in this table contains foreign keys (publication\_id and keyword\_id) referencing specific publications and keywords, indicating which keywords are associated with each publication.

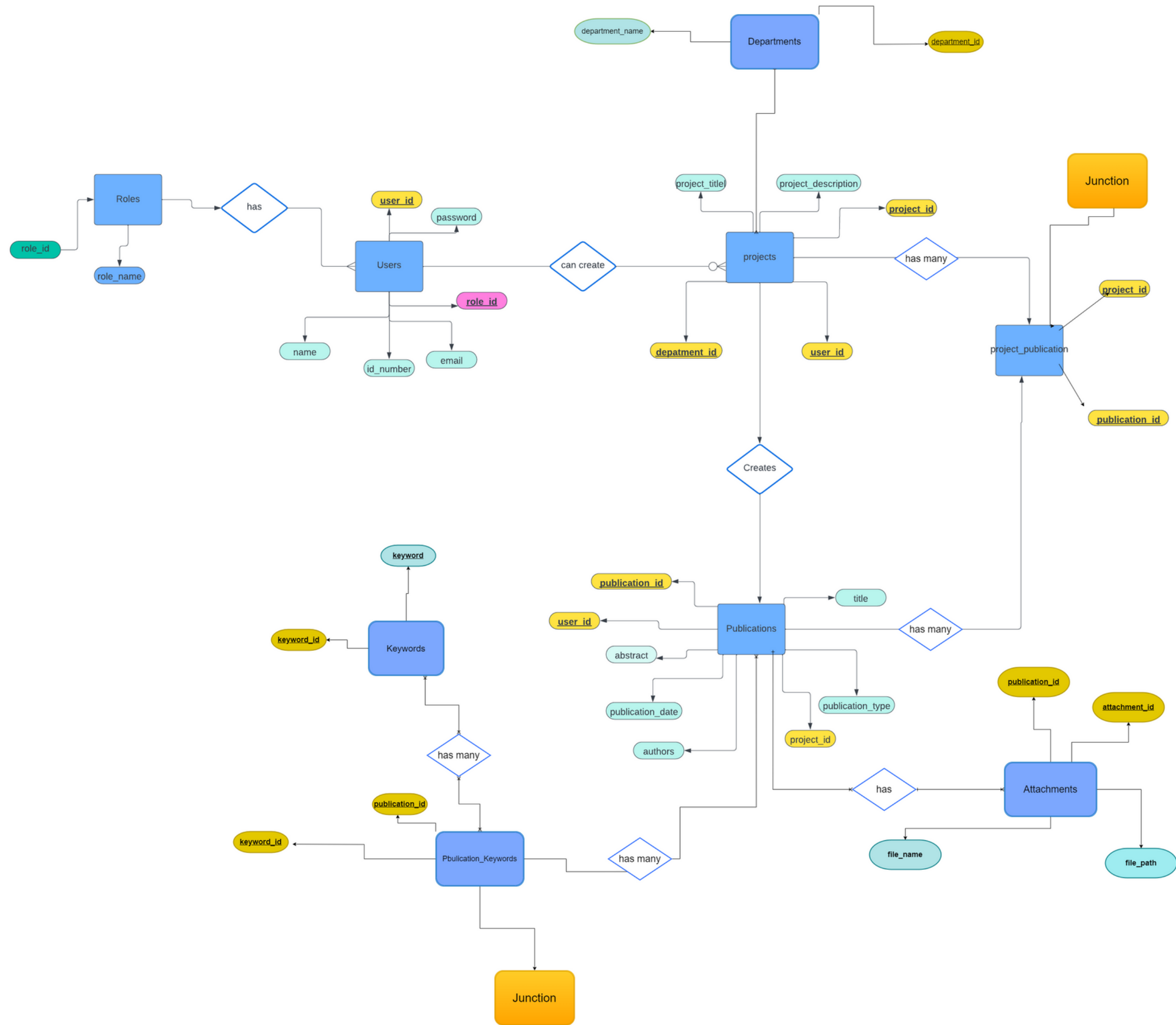
### Attachments Table:

- Stores files or documents related to research projects or publications.
- Each attachment is assigned a unique identifier (**attachment\_id**) and includes details such as the file name and file path.



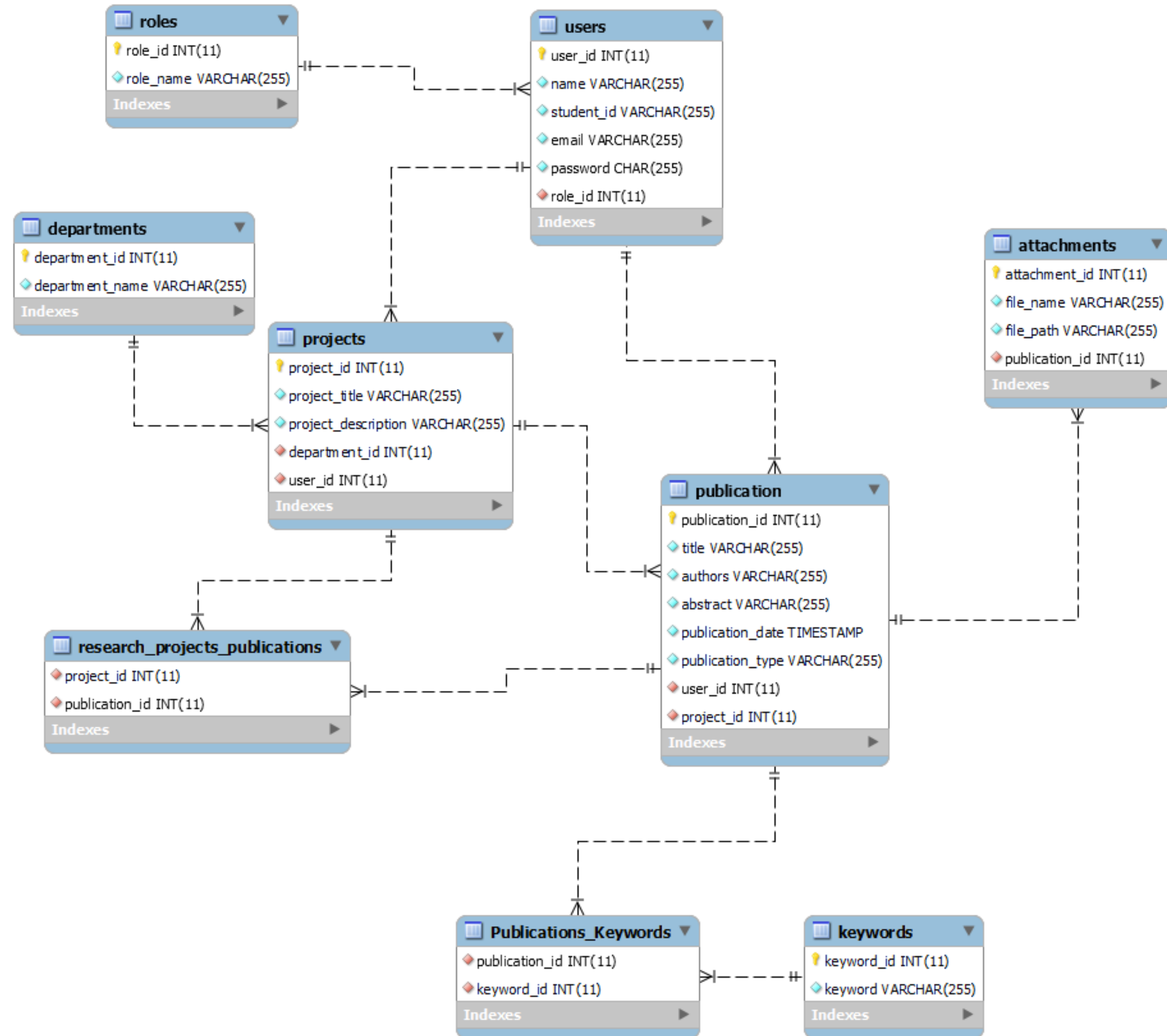


# ENTITY RELATIONSHIP DIAGRAM





# MYSQL ERD



**THANK  
YOU!**