DBMS Project Report

Conference
Paper Management System

Made in collaboration with :

Name : Vedant Varma
SRN : PES1UG22CS681

Name : Vivek M
SRN : PES1UG22CS707

Github :

Description about the project :

This document outlines the requirements for the Conference Paper Management System, designed to facilitate interactions between authors, reviewers, and readers. The system allows authors to submit and manage their papers, reviewers to provide feedback, and readers to search for and access published research.

User Classes:

The system serves three primary user groups—Authors, Reviewers, and Reader with tailored features for each, such as paper submission and profile management for authors, paper review tools for reviewers, and advanced search functionalities for readers.

Purpose:

This document outlines the requirements for the Conference Paper Management System. The system will serve as an intermediary application, facilitating interaction between authors, reviewers, and readers. The system allows authors to submit papers, reviewers to manage reviews, and readers to browse and search for published papers. This document is intended for both the stakeholders and the developers of the system.

Scope:

The system is a comprehensive platform for managing academic publications, designed to streamline interactions among authors, reviewers, and readers. Authors can upload their manuscripts, provide detailed metadata, and manage revisions through the system. It supports the entire submission process, including tracking paper updates and handling supplementary materials. Reviewers are integral to the peer review process, using the system to provide structured feedback, assess paper quality, and communicate with authors. Readers benefit from advanced search and browsing features, allowing them to discover and access published content easily. The system maintains extensive databases for papers, authors, and review data, ensuring efficient retrieval and secure management of sensitive information.

System Features

1 Author Registration and Profile Management

1.1 Description

- This feature allows authors to create and manage their profiles, update personal and academic information, and track their submissions.

1.2 Stimulus/Response Sequences

1. **Stimulus**: Author accesses the registration page.
   - **Response**: System displays a registration form with fields for name, affiliation, email, password, and research interests.
2. **Stimulus** : Author submits the completed form.
   - **Response** : System validates the input data, sends a verification email, and displays a confirmation message.
3. **Stimulus** : Author updates their profile information.
   - **Response** : System saves changes and updates the author's profile page.

1.3 Functional Requirements

- **REQ-4.1.1**: The system shall allow new authors to register by providing their name, affiliation, contact information, and research interests.
- **REQ-4.1.2**: The system shall verify the author's email address through a confirmation link.
- **REQ-4.1.3**: The system shall enable authors to update their biography, research interests, and publication history.
- **REQ-4.1.4**: The system shall display an error message for invalid inputs (e.g., incorrect email format).
- **REQ-4.1.5**: The system shall allow authors to delete their profiles.

2 Paper Submission and Management

2.1 Description

- This feature allows authors to submit papers for conferences or journals and manage the details of their submissions.

2.2 Stimulus/Response Sequences

1. **Stimulus**: Author selects the "Submit Paper" option.
   - **Response**: System displays a submission form with fields for paper title, abstract, keywords, category, and co-authors.
2. **Stimulus**: Author uploads a paper file in PDF or DOCX format.
   - **Response**: System checks the file format and size, then uploads the paper and displays a success message.
3. **Stimulus**: Author views the status of their submitted paper.
   - **Response**: System shows the status (submitted, under review, published) and any reviewer comments.

2.3 Functional Requirements

- **REQ-4.2.1**: The system shall allow authors to submit papers by filling out a submission form.
- **REQ-4.2.2**: The system shall support multiple file formats for uploads, including PDF and DOCX.
- **REQ-4.2.3**: The system shall provide a status tracker for submitted papers, including "submitted," "under review," and "published."
- **REQ-4.2.4**: The system shall allow authors to view reviewer feedback.

3 Reviewer Management

3.1 Description

- This feature allows reviewers to manage their profiles, receive assigned papers, and submit reviews.

3.2 Stimulus/Response Sequences

1. **Stimulus**: Reviewer logs into their account.
   - **Response**: System displays the reviewer's dashboard with assigned papers.
2. **Stimulus**: Reviewer opens a paper assigned for review.
   - **Response**: System displays the paper along with a review form to provide feedback and ratings.
3. **Stimulus**: Reviewer submits the completed review.
   - **Response**: System saves the review and notifies the author.

3.3 Functional Requirements

- **REQ-4.3.1**: The system shall allow reviewers to register and create a profile with their expertise.
- **REQ-4.3.2**: The system shall assign papers to reviewers based on their expertise.
- **REQ-4.3.3**: The system shall allow reviewers to view papers assigned to them.
- **REQ-4.3.4**: The system shall provide a form for reviewers to submit feedback and recommendations.

4 Search and Retrieval

4.1 Description

- Provides tools for readers to search and filter research papers based on various criteria.

4.2 Stimulus/Response Sequences

1. **Stimulus**: Reader enters a query in the search bar.
   - **Response**: System displays a list of relevant papers matching the query.
2. **Stimulus**: Reader applies filters for category and author.
   - **Response**: System updates the search results based on the applied filters.
3. **Stimulus**: Reader selects a paper from the search results.
   - **Response**: System displays the paper's details and download options.

4.3 Functional Requirements

- **REQ-4.4.1**: The system shall provide a search bar for querying papers by author, category, and keywords.
- **REQ-4.4.2**: The system shall support advanced search options with multiple filters
- **REQ-4.4.3**: The system shall display search results in a list format.
- **REQ-4.4.4**: The system shall allow readers to view detailed information about a selected paper.
- **REQ-4.4.5**: The system shall provide options to download papers in available formats.

5 Journals and Conferences Management

5.1 Description

- Manages dedicated pages for each journal or conference.

5.2 Stimulus/Response Sequences

1. **Stimulus**: User selects a specific journal or conference from the list.
   - **Response**: System displays a dedicated page with all associated papers.
2. **Stimulus**: Author submits a paper to a specific journal or conference.

  - **Response**: System processes the submission and updates the author's dashboard.

5.3 Functional Requirements

- **REQ-4.5.1**: The system shall provide dedicated pages for each journal and conference.
- **REQ-4.5.2**: The system shall list all papers associated with a journal or conference on its dedicated page.
- **REQ-4.5.3**: The system shall support direct submissions to journals and conferences.
- **REQ-4.5.4**: The system shall display journal/conference-specific metrics such as impact factor and acceptance rate.

List of Softwares/Tools/Programming languages used :

1) Database Interfaces

=> MySQL (Workbench + Client) (Version 8):

Purpose: Used as the primary relational database management system for storing structured data, including user profiles, paper submissions, review details, and journal/conference information.

2) Application Frameworks and Libraries

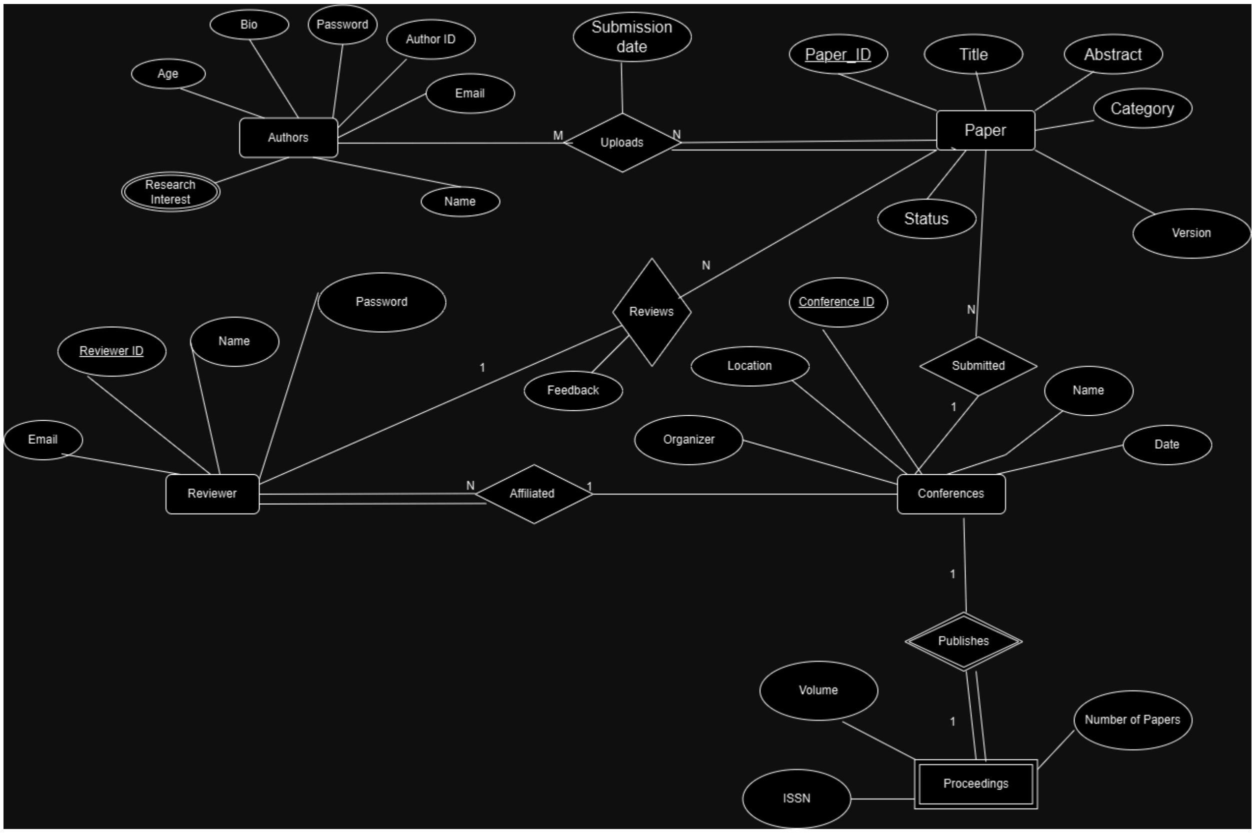
=> **Backend Framework (Flask)**

Purpose: Provides the application framework for managing web requests , handling business logic, and interfacing with the database.

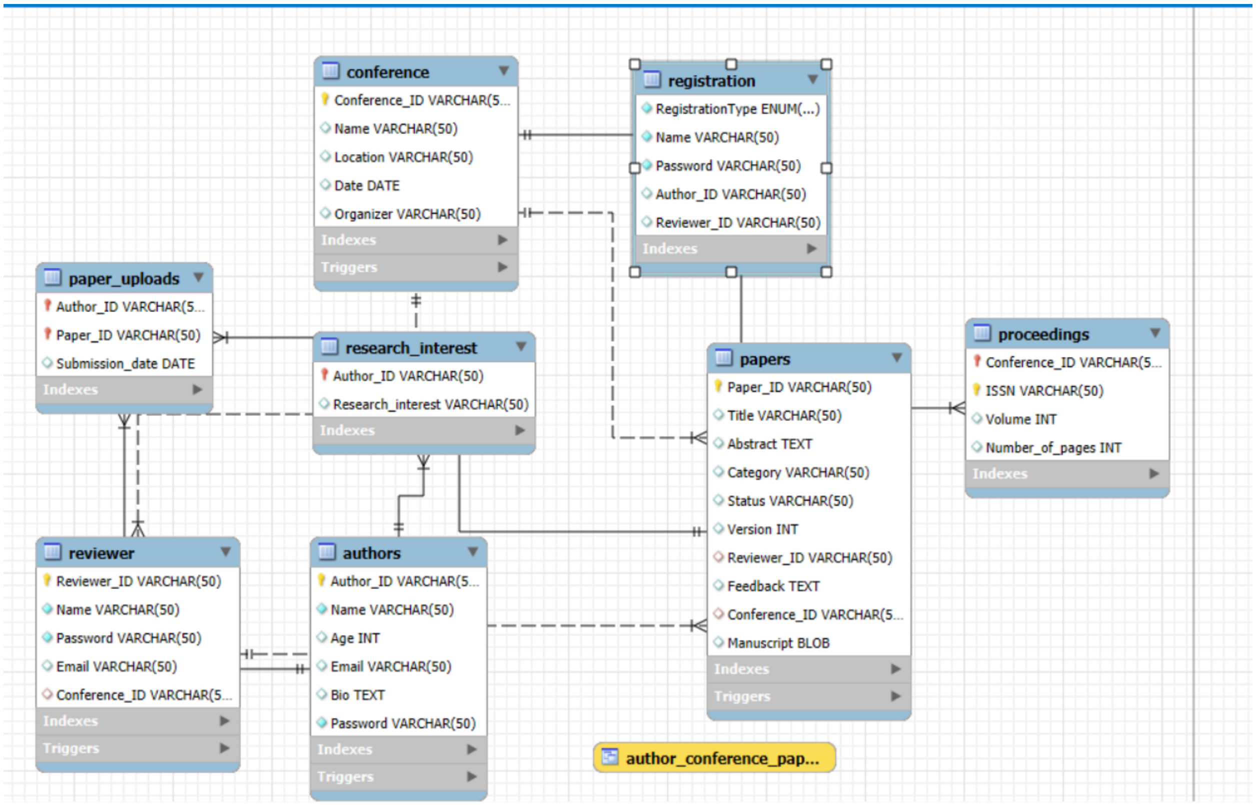
=> **Frontend Framework (HTML,CSS,JS)**:

Purpose: Handles the user interface, providing a dynamic, responsive experience across devices.

ER Diagram :


Bio
Password
Author ID
Age
Email
Authors
Research Interest
Name
Submission date
M
Uploads
N
Paper_ID
Title
Abstract
Category
Paper
Status
Version
N
Reviews
Password
Reviewer ID
Name
1
Feedback
Conference ID
Location
N
Submitted
Name
1
Organizer
Date
Email
Reviewer
N
Affiliated
1
Conferences
1
Publishes
Volume
Number of Papers
1
Proceedings
ISSN

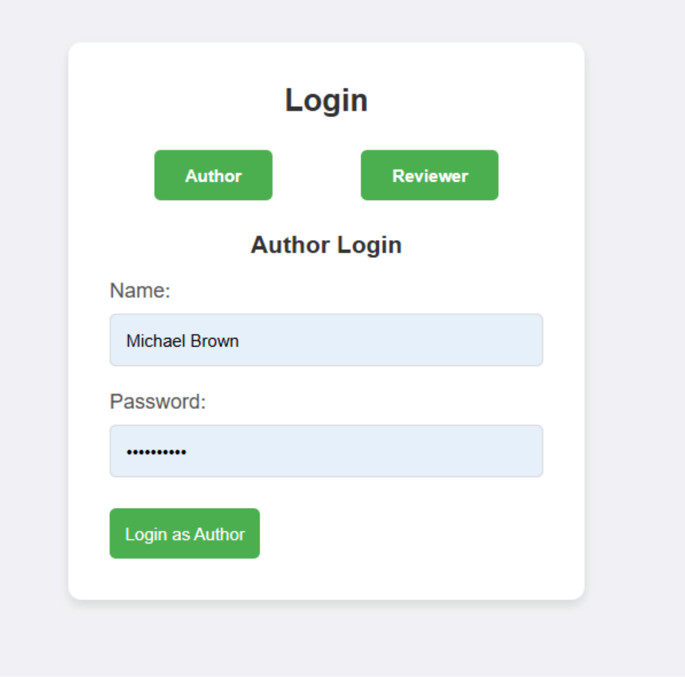
Relational Schema :



Functionalities and Features of the Application

AUTHENTICATION FUNCTIONALITIES

- You can create an account as an author or a reviewer.
- You can log in as an author or a reviewer.

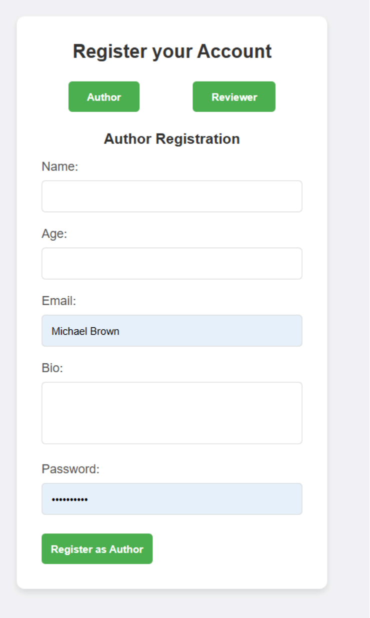


Login

Author Login

Name: Michael Brown

Password: ••••••••••



Register your Account

Author Registration

Name:

Age:

Email: Michael Brown

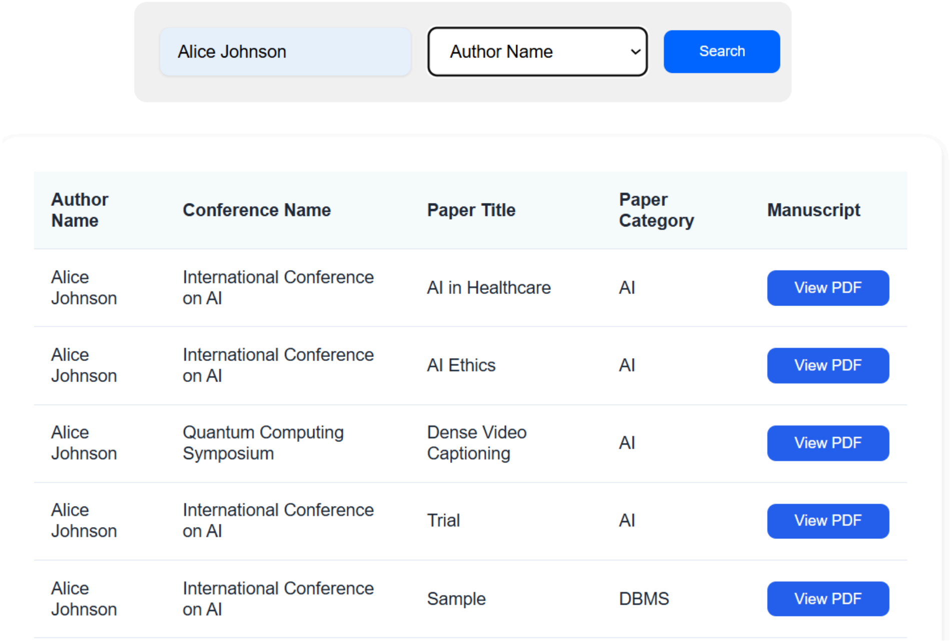
Bio:

Password: ••••••••••

USER FUNCTIONALITIES

1. <u>Search Bar:</u>

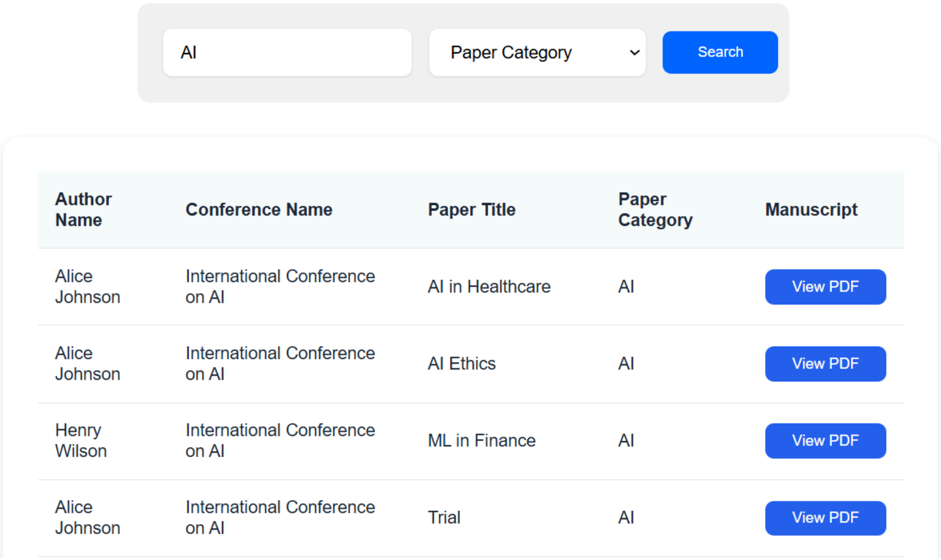
- Papers can be searched using filters like author name, category, and paper title.
- The search results display author name, conference name, paper title, paper category, and manuscript details in a table.
- A view table integrating author information, paper details, and upload records was created to facilitate efficient SQL queries.



Alice Johnson

Author Name

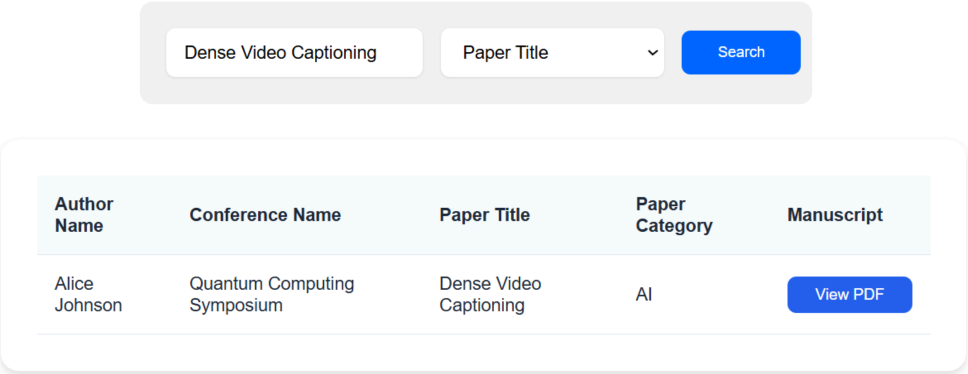
Author Name	Conference Name	Paper Title	Paper Category	Manuscript
Alice Johnson	International Conference on AI	AI in Healthcare	AI	View PDF
Alice Johnson	International Conference on AI	AI Ethics	AI	View PDF
Alice Johnson	Quantum Computing Symposium	Dense Video Captioning	AI	View PDF
Alice Johnson	International Conference on AI	Trial	AI	View PDF
Alice Johnson	International Conference on AI	Sample	DBMS	View PDF



AI

Paper Category

Author Name	Conference Name	Paper Title	Paper Category	Manuscript
Alice Johnson	International Conference on AI	AI in Healthcare	AI	View PDF
Alice Johnson	International Conference on AI	AI Ethics	AI	View PDF
Henry Wilson	International Conference on AI	ML in Finance	AI	View PDF
Alice Johnson	International Conference on AI	Trial	AI	View PDF

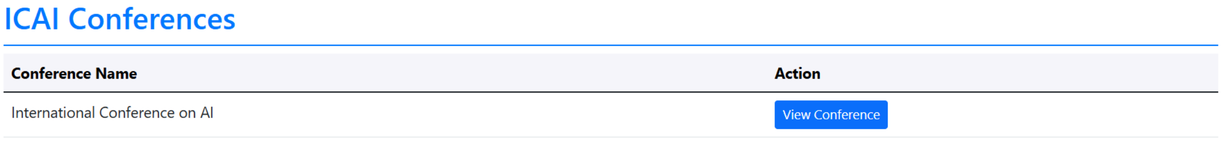


Dense Video Captioning | Paper Title

Author Name	Conference Name	Paper Title	Paper Category	Manuscript
Alice Johnson	Quantum Computing Symposium	Dense Video Captioning	AI	View PDF

2. View Conference List:

- Users can view all conferences organized by any specific organizer [IEEE, ICAI etc]
- The system retrieves and displays all conferences hosted by the organizer.
- Queries are executed on the database to filter and present relevant conferences.

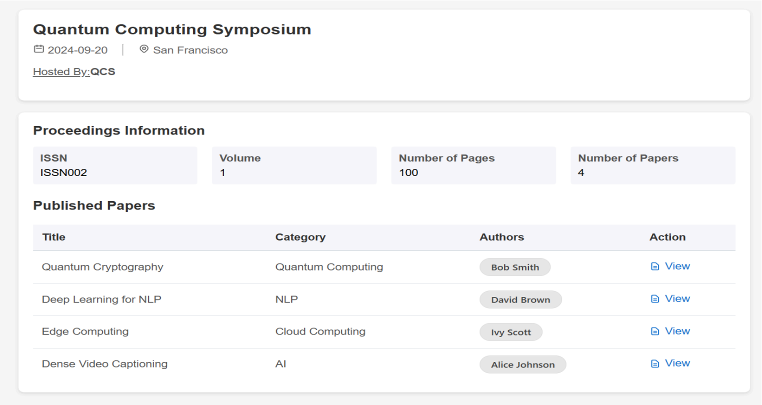


ICAI Conferences

Conference Name	Action
International Conference on AI	View Conference

3. View Conference Details and Proceeding:

- Selecting a conference allows users to view detailed information about the selected conference.
- The conference proceedings display the ISSN, a list of all published papers, and an option to view the PDFs of the papers.
- The system retrieves and organizes the proceedings for easy access and navigation.



Quantum Computing Symposium

2024-09-20 | San Francisco

Hosted By:QCS

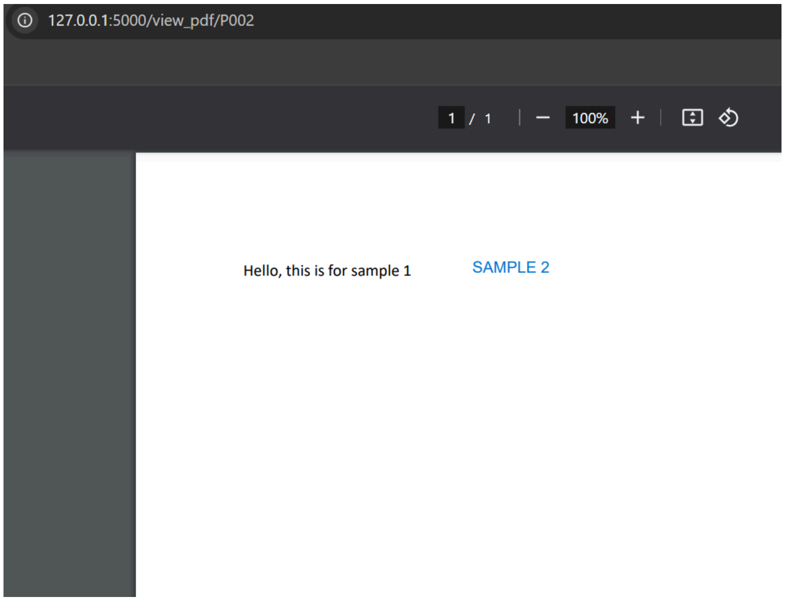
Proceedings Information

ISSN	Volume	Number of Pages	Number of Papers
ISSN002	1	100	4

Published Papers

Title	Category	Authors	Action
Quantum Cryptography	Quantum Computing	Bob Smith	View
Deep Learning for NLP	NLP	David Brown	View
Edge Computing	Cloud Computing	Ivy Scott	View
Dense Video Captioning	AI	Alice Johnson	View

4. <u>Option to View and Download the PDF of research Papers:</u>
    - All manuscripts are stored as a BLOB attribute in the database and rendered to the front end.
    - Users can view the manuscripts directly through the interface.
    - Download functionality is provided for users to access the manuscripts offline.



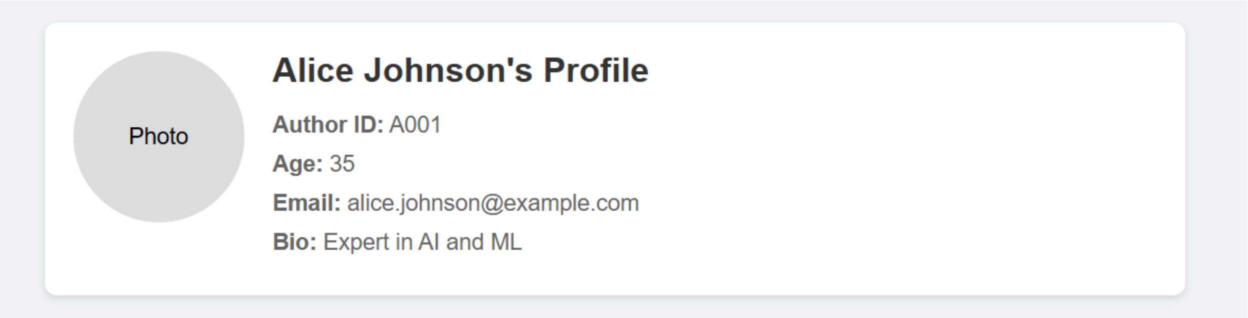
Hello, this is for sample 1

SAMPLE 2

AUTHOR FUNCTIONALITIES

1. <u>View Author Profile:</u>

- The author profile displays all author details in a dedicated section.
- Published papers are listed in a table with information such as the title, conference published, and an option to view the paper.
- A tracking feature allows authors to enter a paper ID to check its status and view reviewer feedback.
- All submissions by the author, including manuscripts and corresponding paper IDs, are systematically maintained and accessible.
- The interface ensures seamless navigation and provides comprehensive details about the author's contributions and submissions.



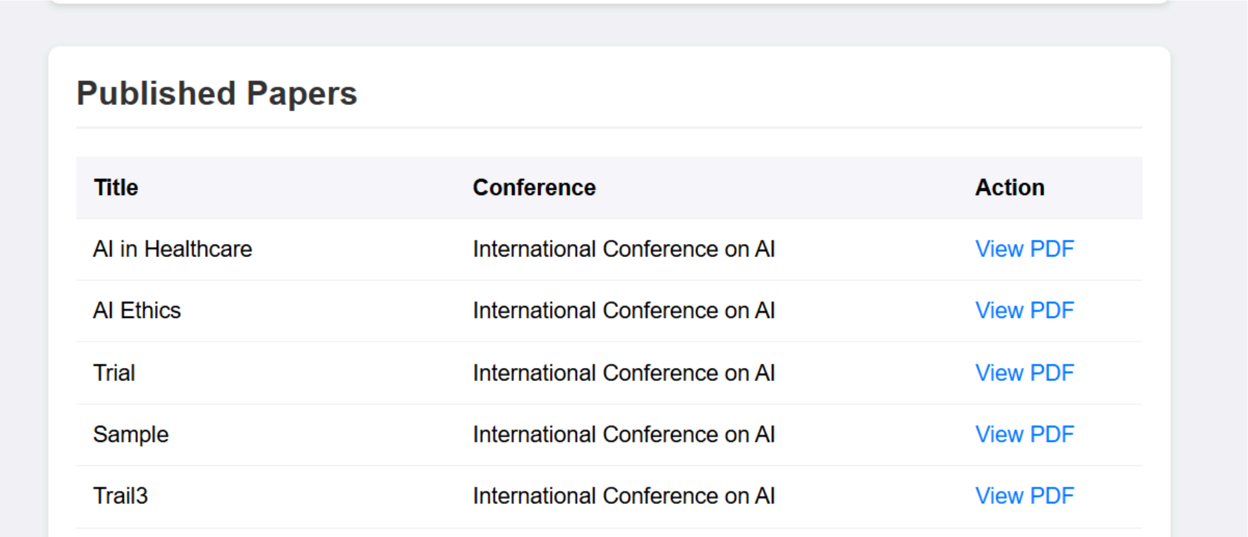
Alice Johnson's Profile

Author ID: A001

Age: 35

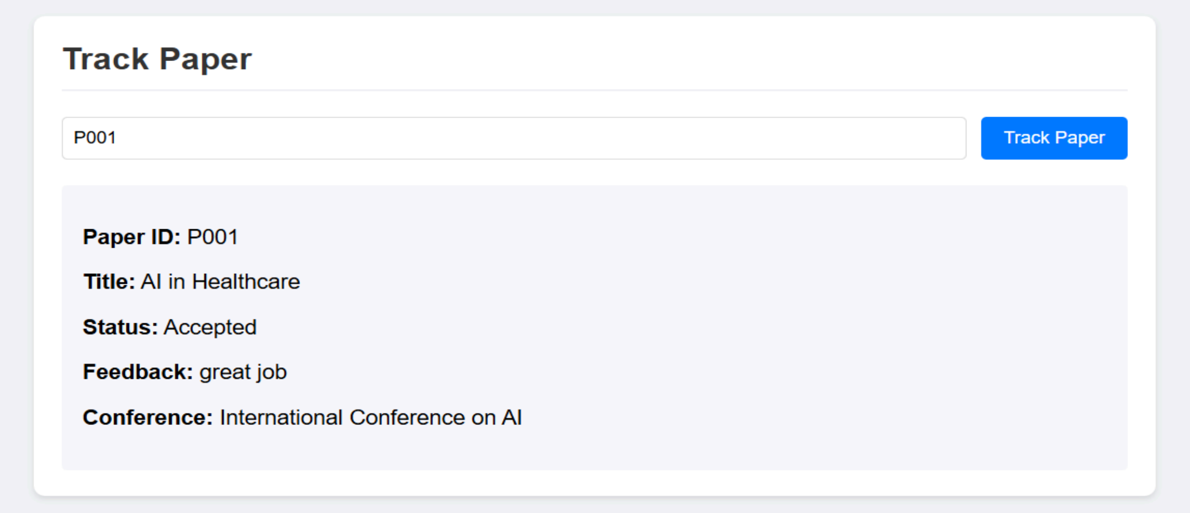
Email: alice.johnson@example.com

Bio: Expert in AI and ML



Published Papers

Title	Conference	Action
AI in Healthcare	International Conference on AI	View PDF
AI Ethics	International Conference on AI	View PDF
Trial	International Conference on AI	View PDF
Sample	International Conference on AI	View PDF
Trail3	International Conference on AI	View PDF



Track Paper

P001

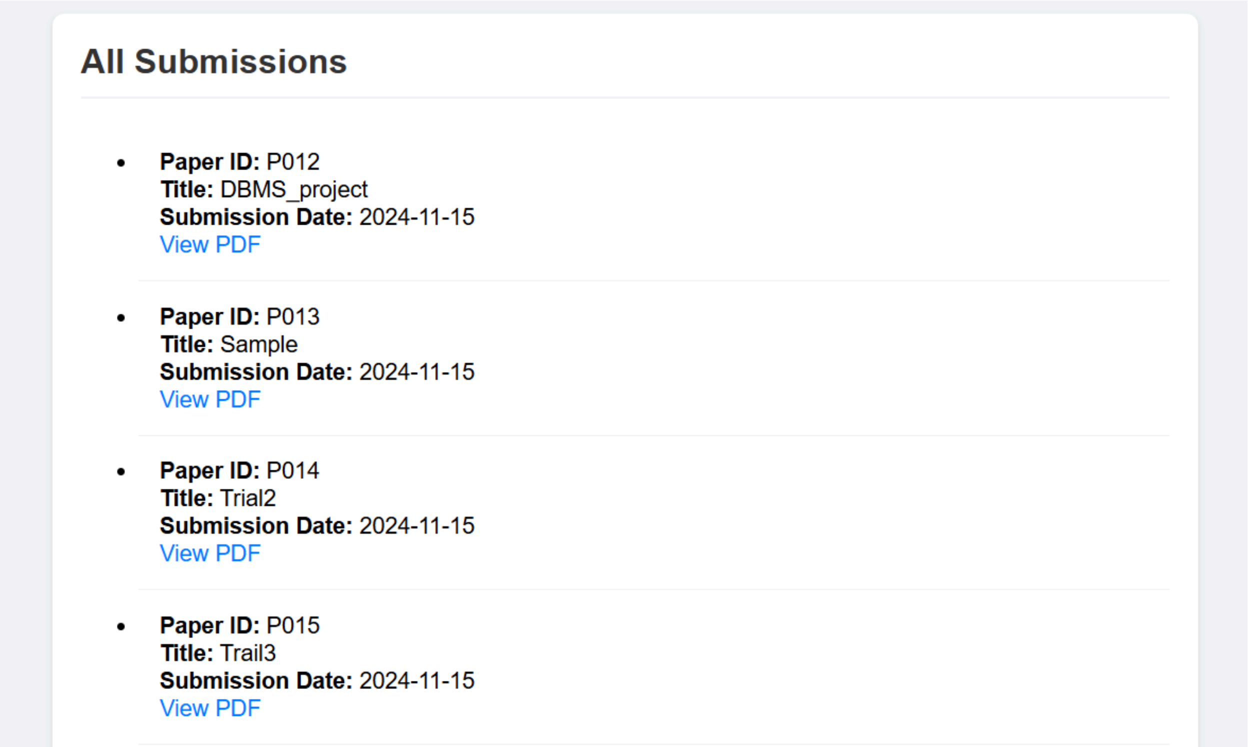
Paper ID: P001

Title: AI in Healthcare

Status: Accepted

Feedback: great job

Conference: International Conference on AI

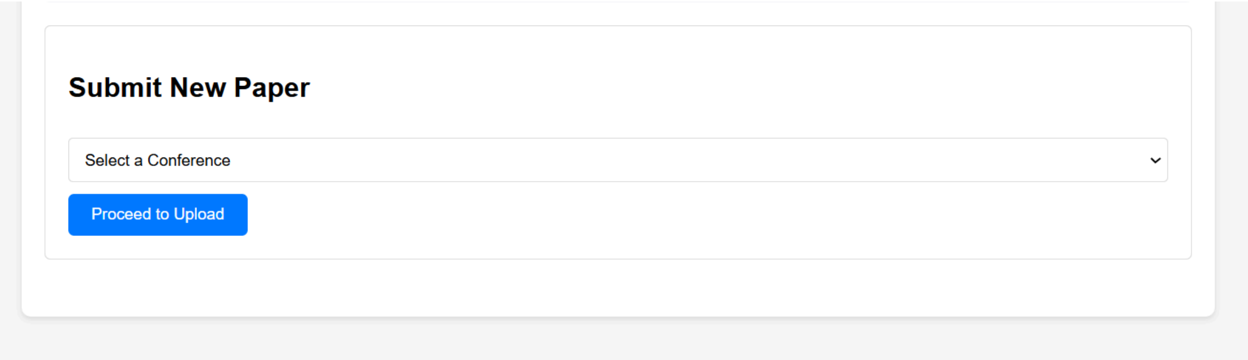


All Submissions

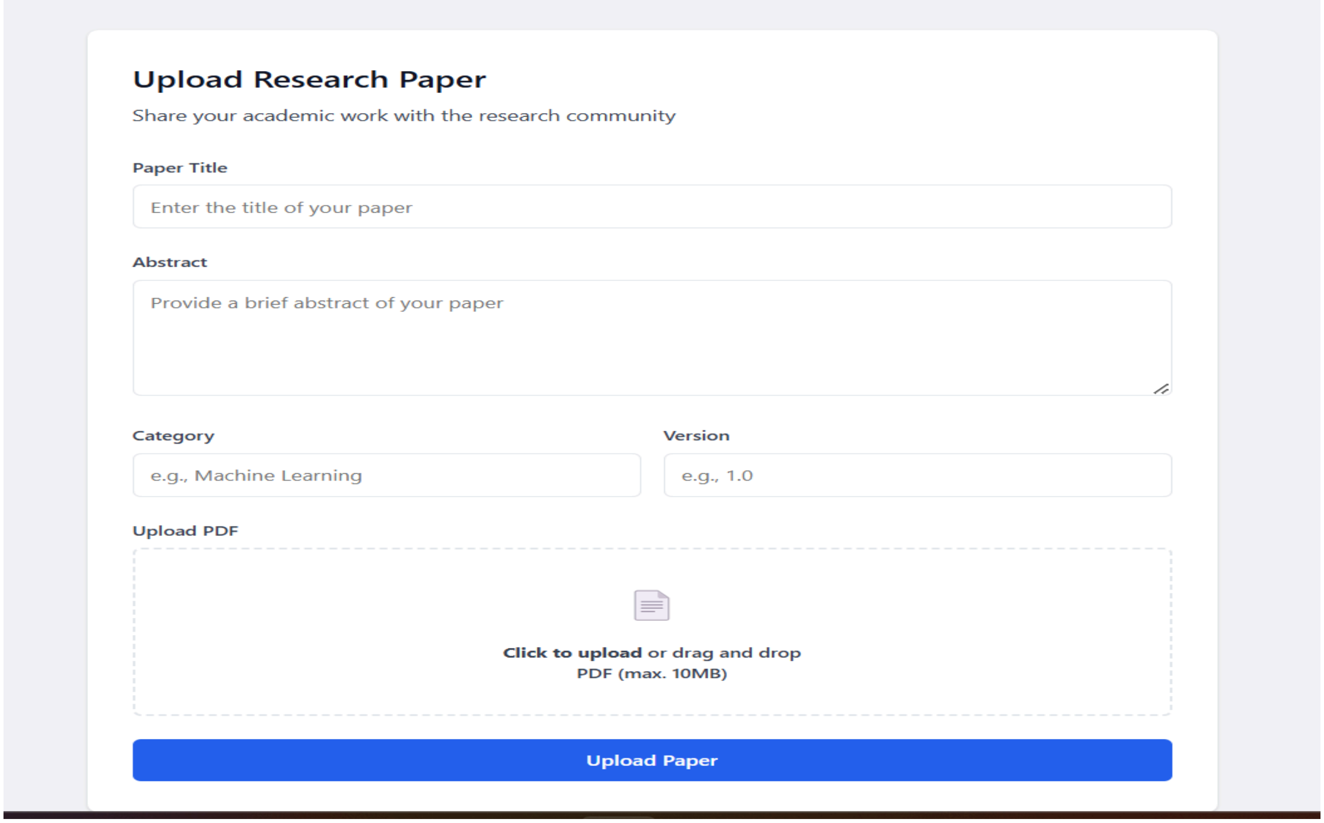
- **Paper ID:** P012
  Title: DBMS_project
  Submission Date: 2024-11-15
  View PDF
- **Paper ID:** P013
  Title: Sample
  Submission Date: 2024-11-15
  View PDF
- **Paper ID:** P014
  Title: Trial2
  Submission Date: 2024-11-15
  View PDF
- **Paper ID:** P015
  Title: Trail3
  Submission Date: 2024-11-15
  View PDF

2. Upload Research Papers:

- Authors can select any available conference with dates greater than the current date for submission.
- Authors can upload PDFs of their research paper manuscripts, submitting them to the chosen conference.



Submit New Paper



Upload Research Paper

Share your academic work with the research community

Paper Title

Enter the title of your paper

Abstract

Provide a brief abstract of your paper

Category

e.g., Machine Learning

Version

e.g., 1.0

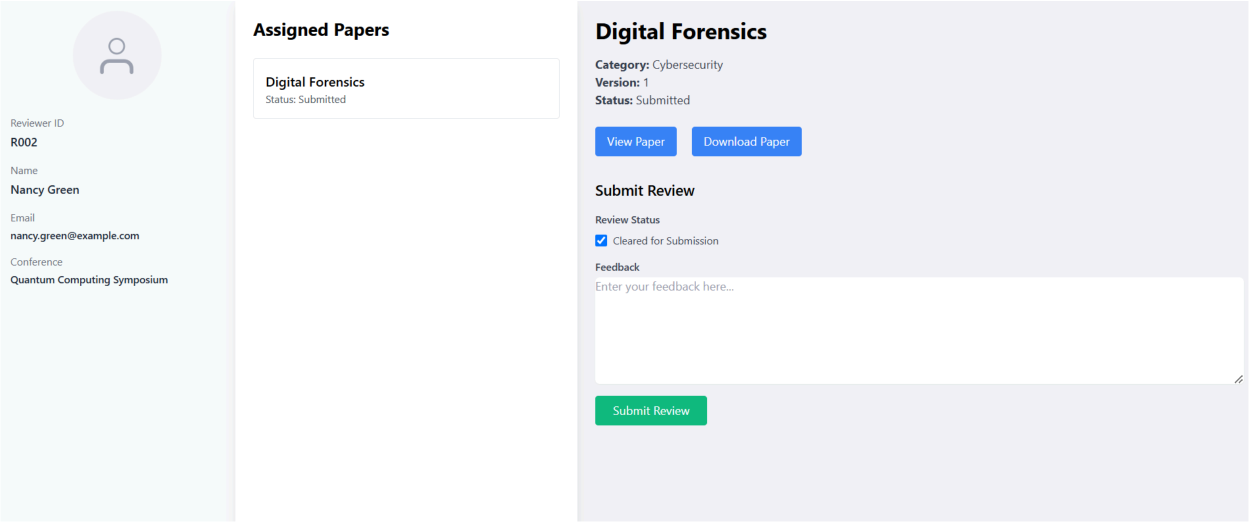
Upload PDF

Click to upload or drag and drop

PDF (max. 10MB)

REVIEWER FUNCTIONALITIES

- The backend algorithm maps all papers submitted to a conference to reviewers affiliated with the conference.
- Reviewers can access their profile page to view their personal details.
- Reviewers can check the papers assigned to them and provide feedback for each.
- Reviewers can mark the papers as cleared for publishing by checking the respective box after reviewing.



Reviewer ID
R002

Name
Nancy Green

Email
nancy.green@example.com

Conference
Quantum Computing Symposium

Assigned Papers

Digital Forensics
Status: Submitted

Digital Forensics

Category: Cybersecurity
Version: 1
Status: Submitted

Submit Review

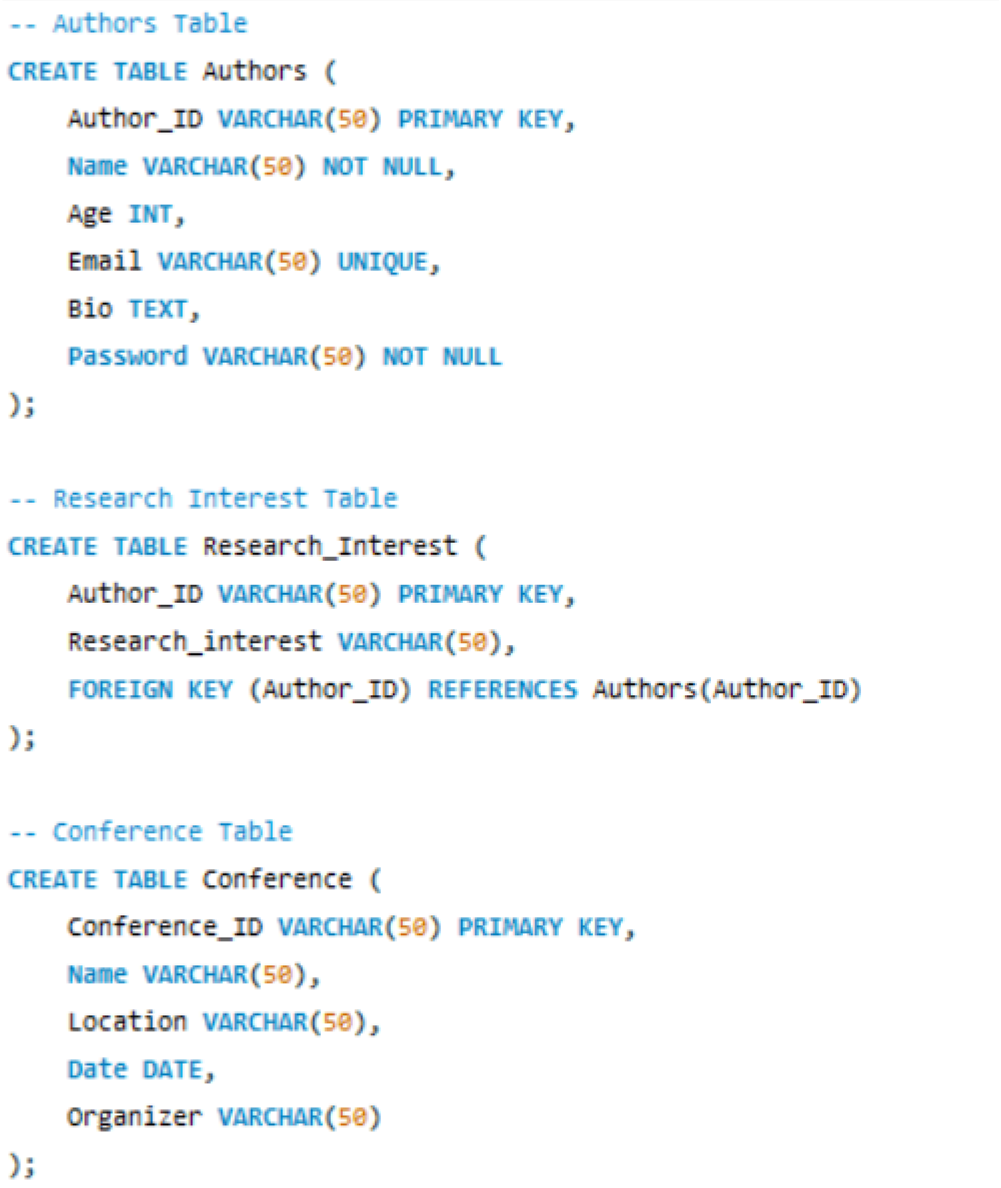
Review Status

☑ Cleared for Submission

Feedback

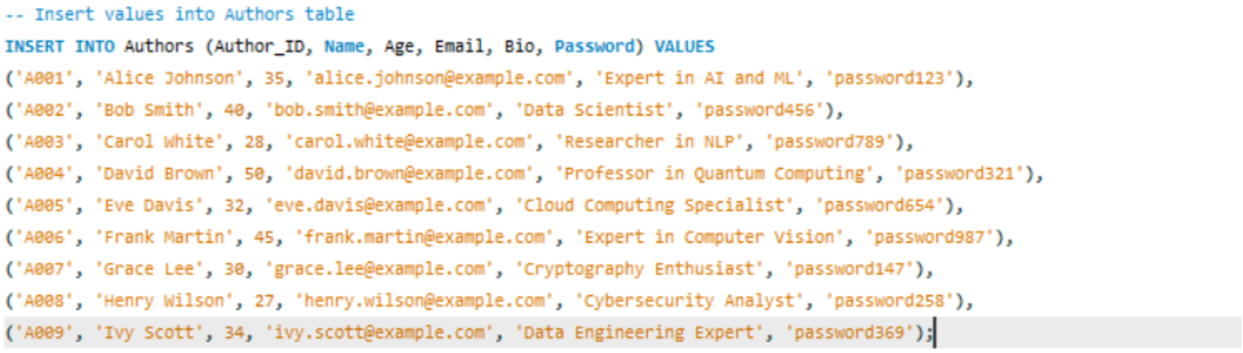
Enter your feedback here...

Create operations :



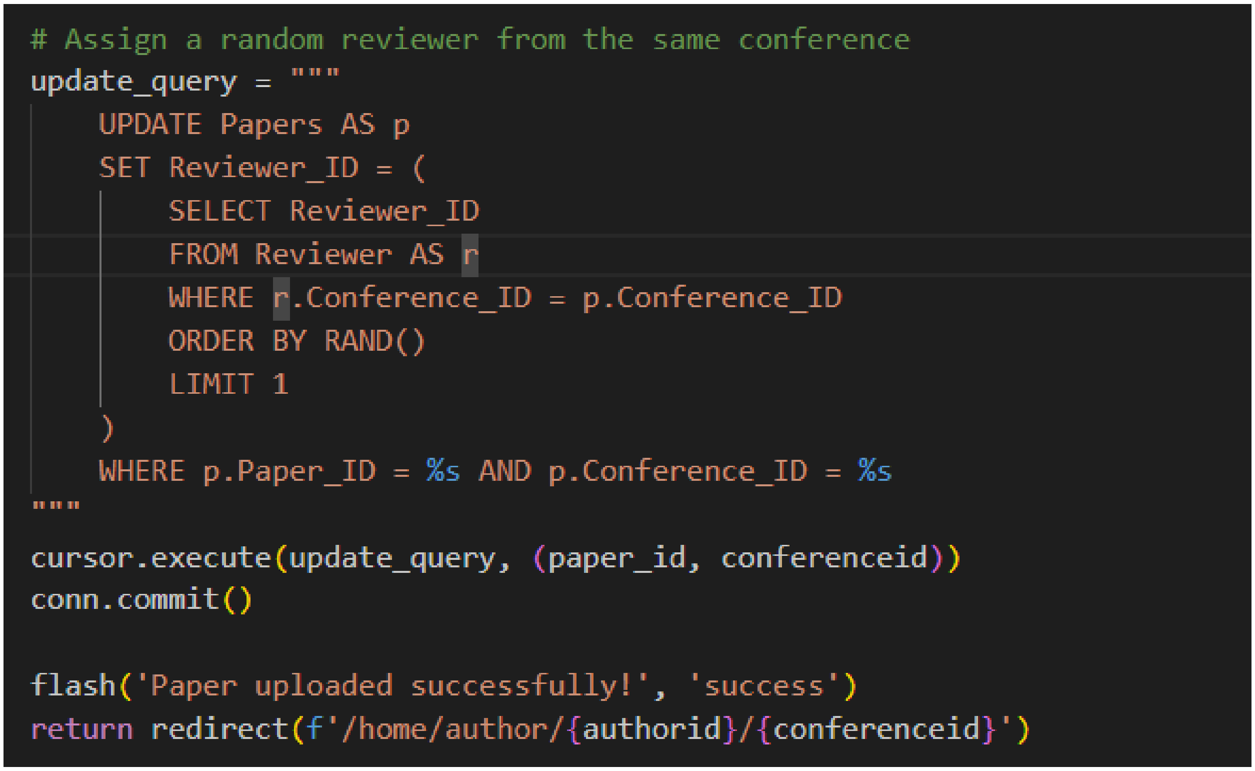
```
-- Authors Table
CREATE TABLE Authors (
    Author_ID VARCHAR(50) PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Age INT,
    Email VARCHAR(50) UNIQUE,
    Bio TEXT,
    Password VARCHAR(50) NOT NULL
);

-- Research Interest Table
CREATE TABLE Research_Interest (
    Author_ID VARCHAR(50) PRIMARY KEY,
    Research_interest VARCHAR(50),
    FOREIGN KEY (Author_ID) REFERENCES Authors(Author_ID)
);

-- Conference Table
CREATE TABLE Conference (
    Conference_ID VARCHAR(50) PRIMARY KEY,
    Name VARCHAR(50),
    Location VARCHAR(50),
    Date DATE,
    Organizer VARCHAR(50)
);
```

Insert Operations :



```sql
-- Insert values into Authors table
INSERT INTO Authors (Author_ID, Name, Age, Email, Bio, Password) VALUES
('A001', 'Alice Johnson', 35, 'alice.johnson@example.com', 'Expert in AI and ML', 'password123'),
('A002', 'Bob Smith', 40, 'bob.smith@example.com', 'Data Scientist', 'password456'),
('A003', 'Carol White', 28, 'carol.white@example.com', 'Researcher in NLP', 'password789'),
('A004', 'David Brown', 50, 'david.brown@example.com', 'Professor in Quantum Computing', 'password321'),
('A005', 'Eve Davis', 32, 'eve.davis@example.com', 'Cloud Computing Specialist', 'password654'),
('A006', 'Frank Martin', 45, 'frank.martin@example.com', 'Expert in Computer Vision', 'password987'),
('A007', 'Grace Lee', 30, 'grace.lee@example.com', 'Cryptography Enthusiast', 'password147'),
('A008', 'Henry Wilson', 27, 'henry.wilson@example.com', 'Cybersecurity Analyst', 'password258'),
('A009', 'Ivy Scott', 34, 'ivy.scott@example.com', 'Data Engineering Expert', 'password369');
```

Update Operations :

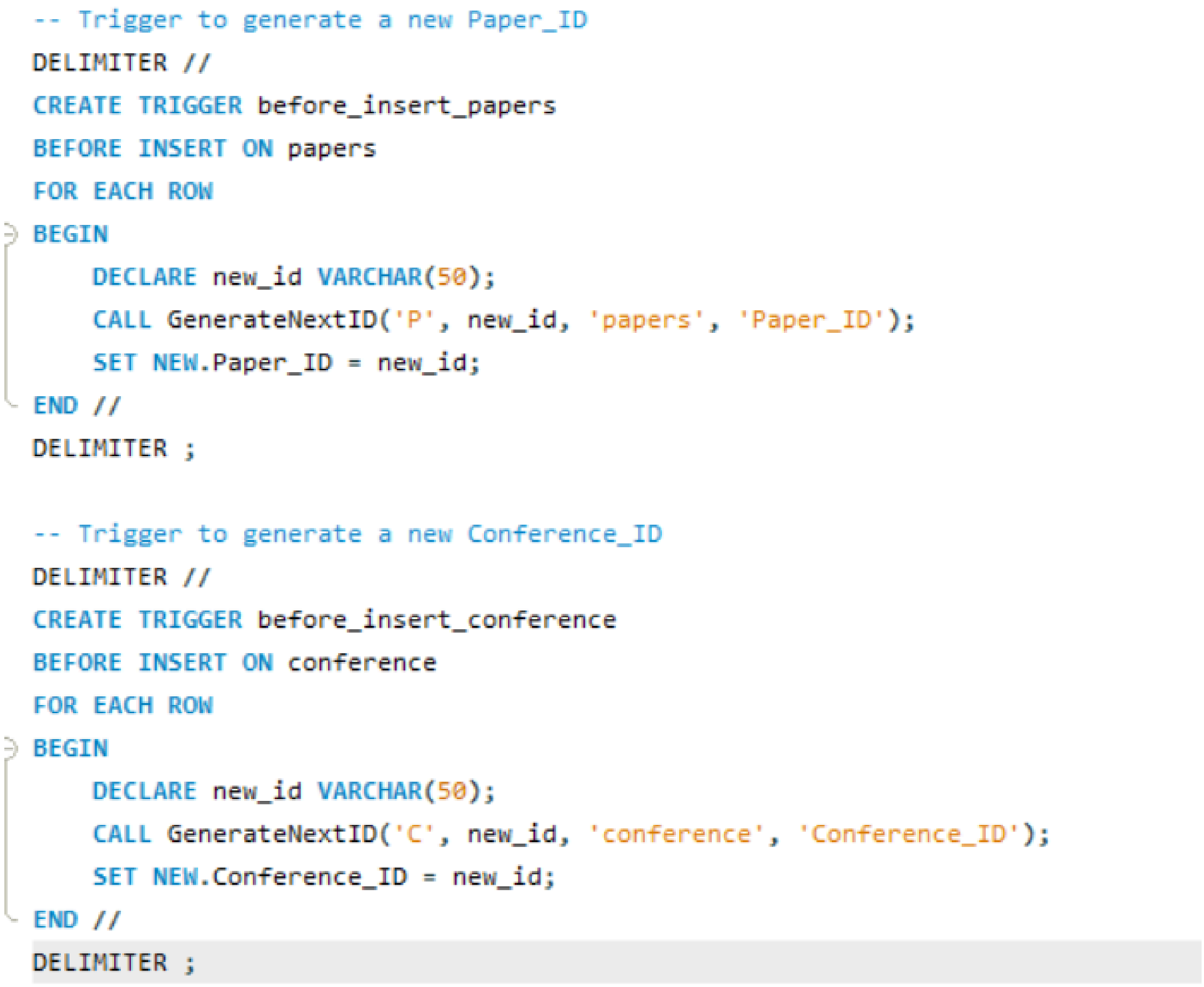


```python
# Assign a random reviewer from the same conference
update_query = """
    UPDATE Papers AS p
    SET Reviewer_ID = (
        SELECT Reviewer_ID
        FROM Reviewer AS r
        WHERE r.Conference_ID = p.Conference_ID
        ORDER BY RAND()
        LIMIT 1
    )
    WHERE p.Paper_ID = %s AND p.Conference_ID = %s
"""
cursor.execute(update_query, (paper_id, conferenceid))
conn.commit()

flash('Paper uploaded successfully!', 'success')
return redirect(f'/home/author/{authorid}/{conferenceid}')
```

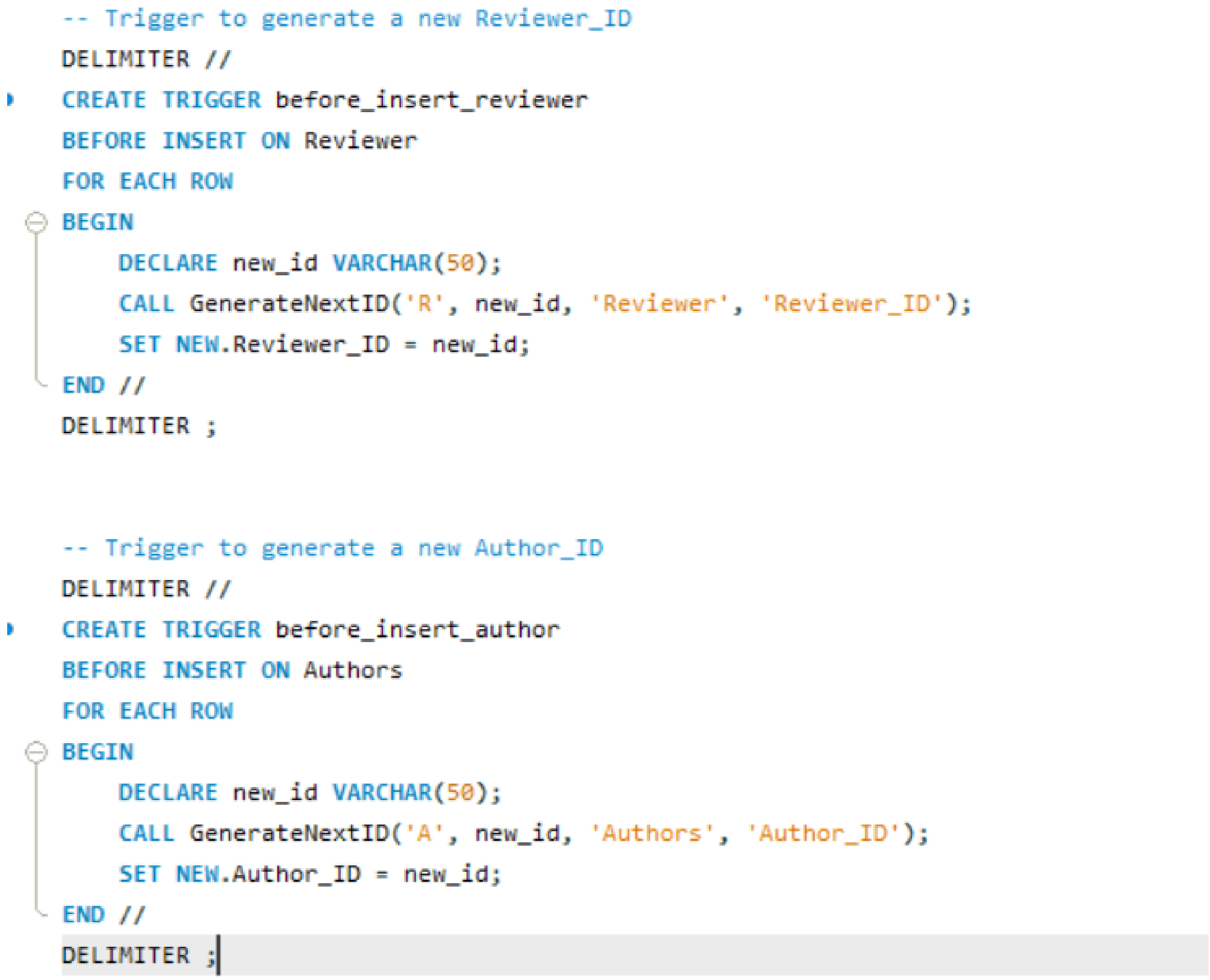
Queries based on Rubrics :

Triggers :

All primary keys in our tables are of type VARCHAR, which means auto-increment cannot be used. To address this, instead of manually generating the next key in the backend for every table, we developed a procedure that automatically generates the next ID. We then set up triggers that call this procedure, ensuring that the primary key is dynamically updated whenever an insert operation takes place. This approach streamlines the process of key generation and maintains consistency across the database.

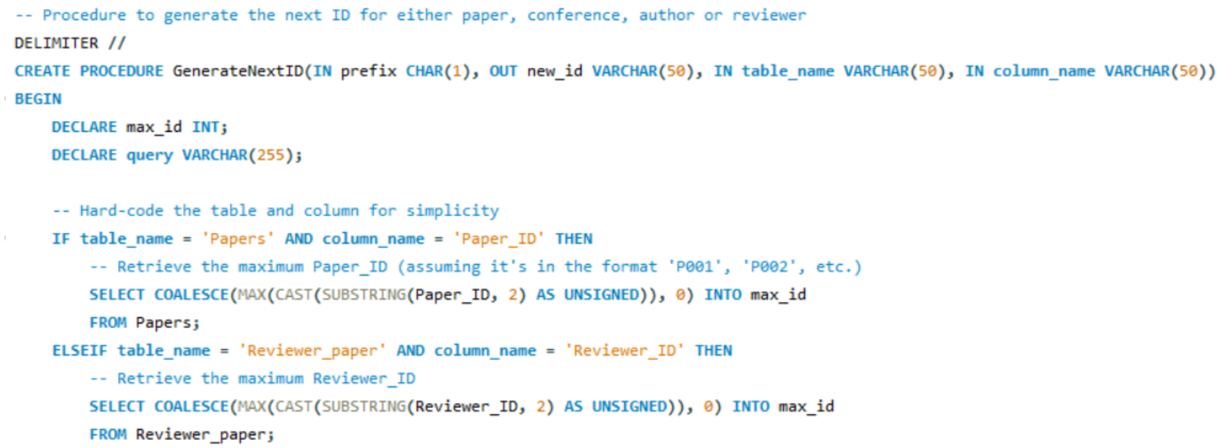


```sql
-- Trigger to generate a new Paper_ID
DELIMITER //
CREATE TRIGGER before_insert_papers
BEFORE INSERT ON papers
FOR EACH ROW
BEGIN
    DECLARE new_id VARCHAR(50);
    CALL GenerateNextID('P', new_id, 'papers', 'Paper_ID');
    SET NEW.Paper_ID = new_id;
END //
DELIMITER ;

-- Trigger to generate a new Conference_ID
DELIMITER //
CREATE TRIGGER before_insert_conference
BEFORE INSERT ON conference
FOR EACH ROW
BEGIN
    DECLARE new_id VARCHAR(50);
    CALL GenerateNextID('C', new_id, 'conference', 'Conference_ID');
    SET NEW.Conference_ID = new_id;
END //
DELIMITER ;
```

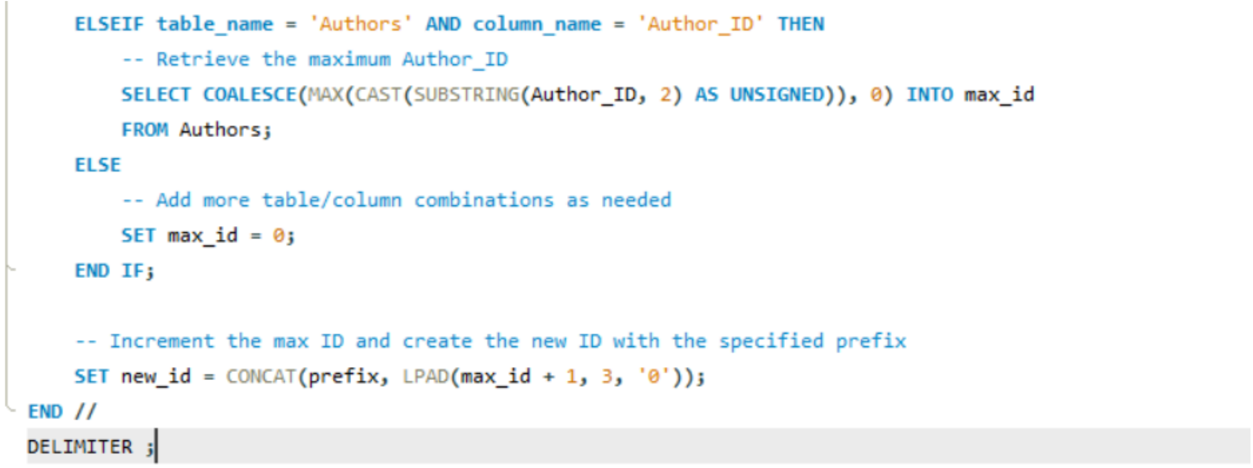


```
-- Trigger to generate a new Reviewer_ID
DELIMITER //
CREATE TRIGGER before_insert_reviewer
BEFORE INSERT ON Reviewer
FOR EACH ROW
BEGIN
    DECLARE new_id VARCHAR(50);
    CALL GenerateNextID('R', new_id, 'Reviewer', 'Reviewer_ID');
    SET NEW.Reviewer_ID = new_id;
END //
DELIMITER ;


-- Trigger to generate a new Author_ID
DELIMITER //
CREATE TRIGGER before_insert_author
BEFORE INSERT ON Authors
FOR EACH ROW
BEGIN
    DECLARE new_id VARCHAR(50);
    CALL GenerateNextID('A', new_id, 'Authors', 'Author_ID');
    SET NEW.Author_ID = new_id;
END //
DELIMITER ;
```

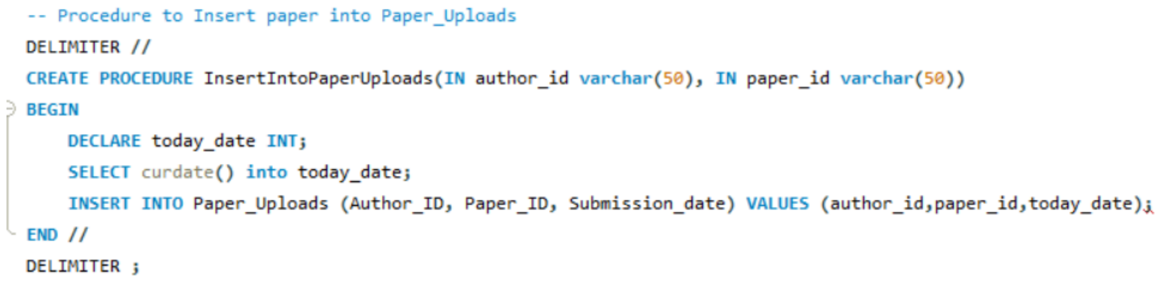
Procedures :



```
-- Procedure to generate the next ID for either paper, conference, author or reviewer
DELIMITER //
CREATE PROCEDURE GenerateNextID(IN prefix CHAR(1), OUT new_id VARCHAR(50), IN table_name VARCHAR(50), IN column_name VARCHAR(50))
BEGIN
    DECLARE max_id INT;
    DECLARE query VARCHAR(255);

    -- Hard-code the table and column for simplicity
    IF table_name = 'Papers' AND column_name = 'Paper_ID' THEN
        -- Retrieve the maximum Paper_ID (assuming it's in the format 'P001', 'P002', etc.)
        SELECT COALESCE(MAX(CAST(SUBSTRING(Paper_ID, 2) AS UNSIGNED)), 0) INTO max_id
        FROM Papers;
    ELSEIF table_name = 'Reviewer_paper' AND column_name = 'Reviewer_ID' THEN
        -- Retrieve the maximum Reviewer_ID
        SELECT COALESCE(MAX(CAST(SUBSTRING(Reviewer_ID, 2) AS UNSIGNED)), 0) INTO max_id
        FROM Reviewer_paper;
```

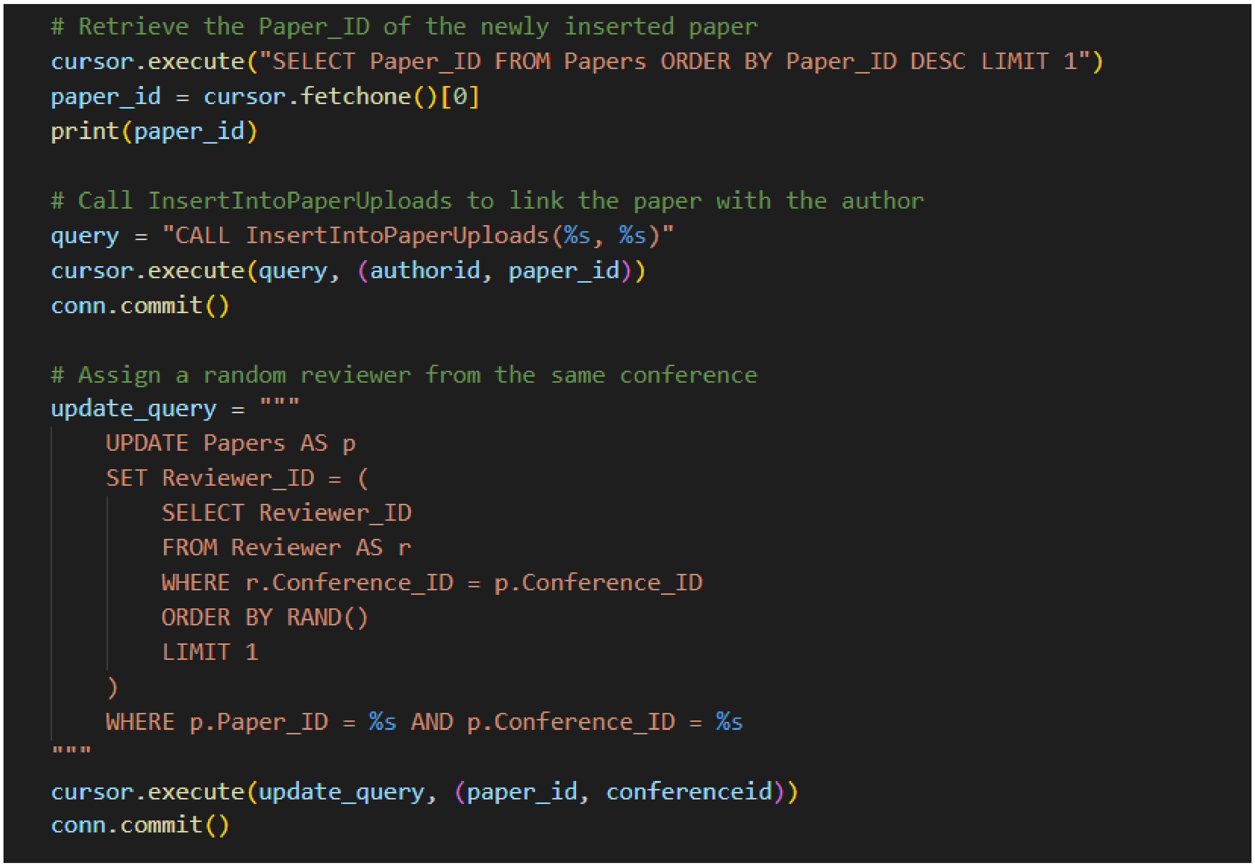


```
    ELSEIF table_name = 'Authors' AND column_name = 'Author_ID' THEN
        -- Retrieve the maximum Author_ID
        SELECT COALESCE(MAX(CAST(SUBSTRING(Author_ID, 2) AS UNSIGNED)), 0) INTO max_id
        FROM Authors;
    ELSE
        -- Add more table/column combinations as needed
        SET max_id = 0;
    END IF;

    -- Increment the max ID and create the new ID with the specified prefix
    SET new_id = CONCAT(prefix, LPAD(max_id + 1, 3, '0'));
END //
DELIMITER ;
```



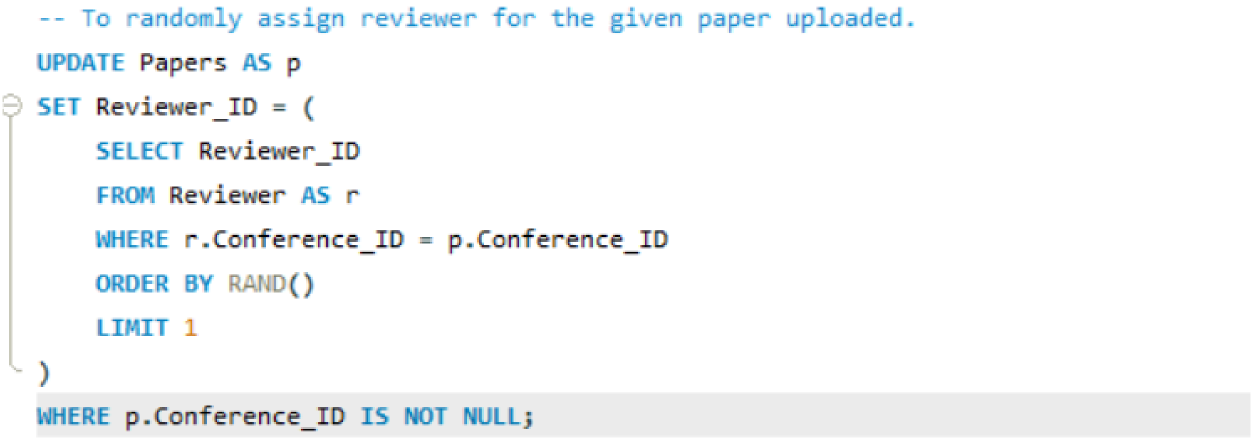
```
-- Procedure to Insert paper into Paper_Uploads
DELIMITER //
CREATE PROCEDURE InsertIntoPaperUploads(IN author_id varchar(50), IN paper_id varchar(50))
BEGIN
    DECLARE today_date INT;
    SELECT curdate() into today_date;
    INSERT INTO Paper_Uploads (Author_ID, Paper_ID, Submission_date) VALUES (author_id,paper_id,today_date);
END //
DELIMITER ;
```

Calling procedure InsertIntoPaperUploads in the code :



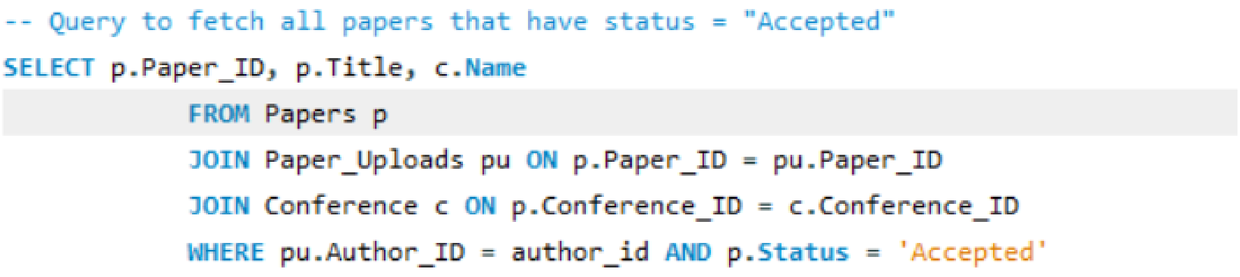
```
# Retrieve the Paper_ID of the newly inserted paper
cursor.execute("SELECT Paper_ID FROM Papers ORDER BY Paper_ID DESC LIMIT 1")
paper_id = cursor.fetchone()[0]
print(paper_id)

# Call InsertIntoPaperUploads to link the paper with the author
query = "CALL InsertIntoPaperUploads(%s, %s)"
cursor.execute(query, (authorid, paper_id))
conn.commit()

# Assign a random reviewer from the same conference
update_query = """
    UPDATE Papers AS p
    SET Reviewer_ID = (
        SELECT Reviewer_ID
        FROM Reviewer AS r
        WHERE r.Conference_ID = p.Conference_ID
        ORDER BY RAND()
        LIMIT 1
    )
    WHERE p.Paper_ID = %s AND p.Conference_ID = %s
"""
cursor.execute(update_query, (paper_id, conferenceid))
conn.commit()
```

Nested Query :

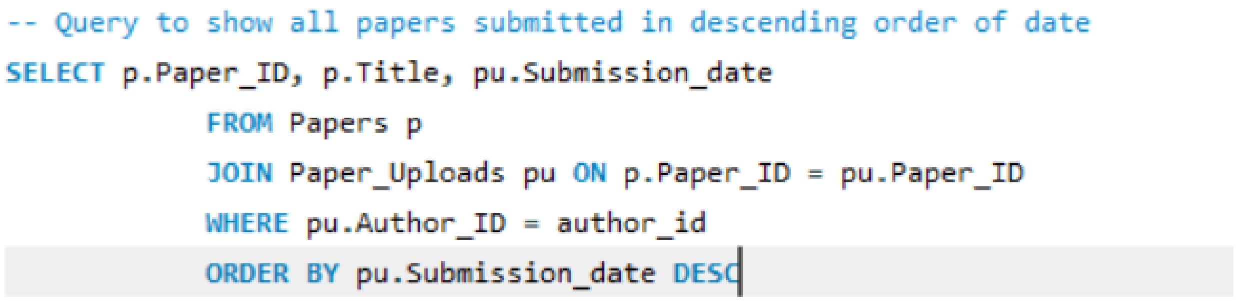


```
-- To randomly assign reviewer for the given paper uploaded.
UPDATE Papers AS p
SET Reviewer_ID = (
    SELECT Reviewer_ID
    FROM Reviewer AS r
    WHERE r.Conference_ID = p.Conference_ID
    ORDER BY RAND()
    LIMIT 1
)
WHERE p.Conference_ID IS NOT NULL;
```

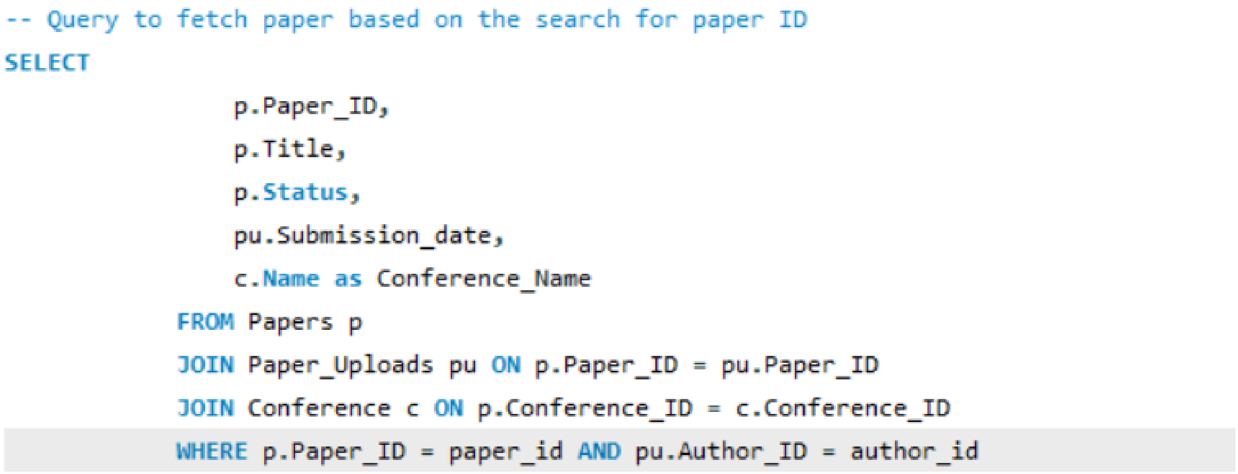
Join :



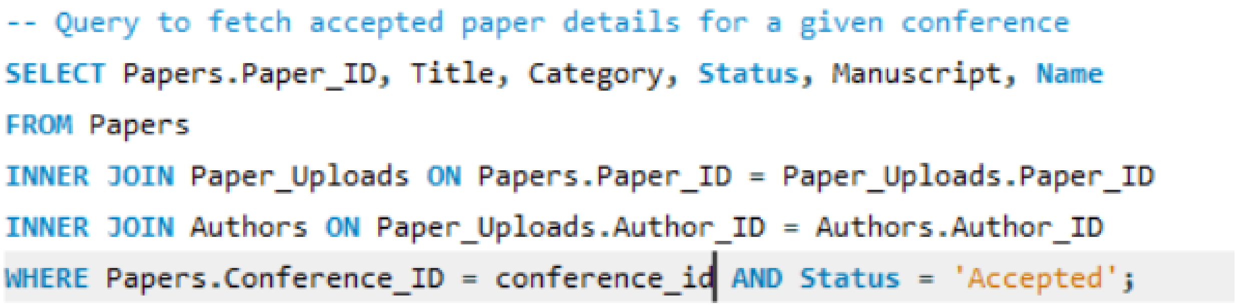
```
-- Query to fetch all papers that have status = "Accepted"
SELECT p.Paper_ID, p.Title, c.Name
            FROM Papers p
            JOIN Paper_Uploads pu ON p.Paper_ID = pu.Paper_ID
            JOIN Conference c ON p.Conference_ID = c.Conference_ID
            WHERE pu.Author_ID = author_id AND p.Status = 'Accepted'
```



```
-- Query to show all papers submitted in descending order of date
SELECT p.Paper_ID, p.Title, pu.Submission_date
            FROM Papers p
            JOIN Paper_Uploads pu ON p.Paper_ID = pu.Paper_ID
            WHERE pu.Author_ID = author_id
            ORDER BY pu.Submission_date DESC
```

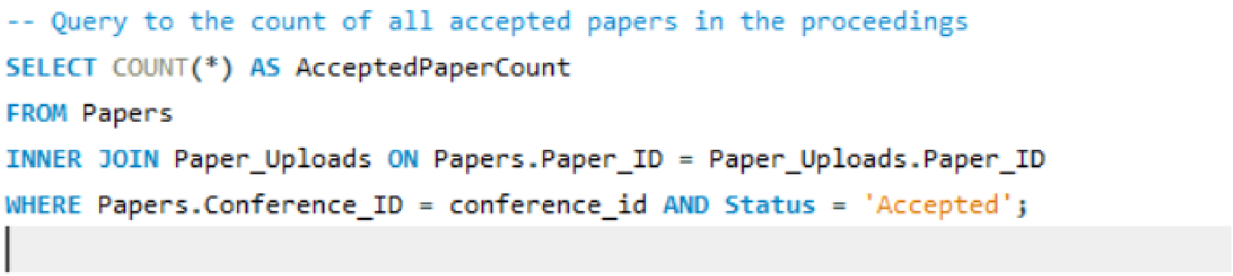


```
-- Query to fetch paper based on the search for paper ID
SELECT
                p.Paper_ID,
                p.Title,
                p.Status,
                pu.Submission_date,
                c.Name as Conference_Name
            FROM Papers p
            JOIN Paper_Uploads pu ON p.Paper_ID = pu.Paper_ID
            JOIN Conference c ON p.Conference_ID = c.Conference_ID
            WHERE p.Paper_ID = paper_id AND pu.Author_ID = author_id
```



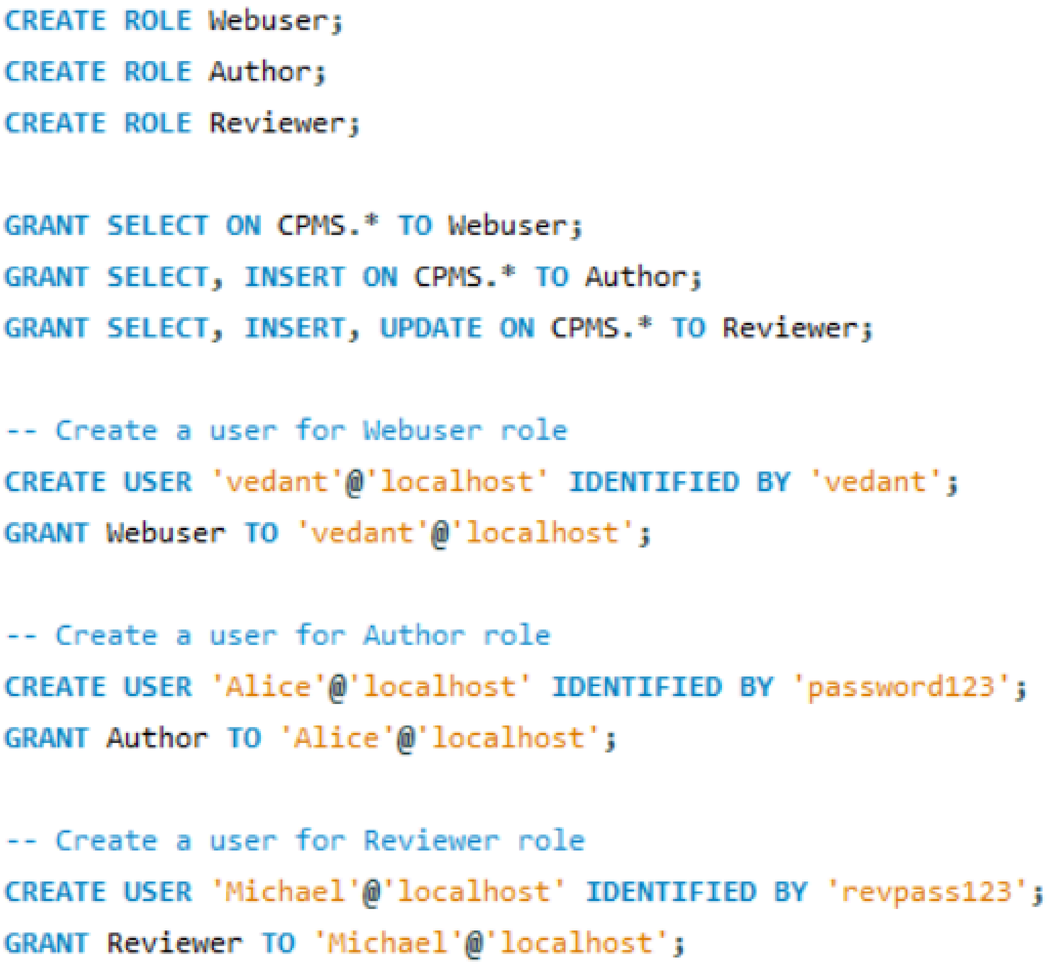
```
-- Query to fetch accepted paper details for a given conference
SELECT Papers.Paper_ID, Title, Category, Status, Manuscript, Name
FROM Papers
INNER JOIN Paper_Uploads ON Papers.Paper_ID = Paper_Uploads.Paper_ID
INNER JOIN Authors ON Paper_Uploads.Author_ID = Authors.Author_ID
WHERE Papers.Conference_ID = conference_id AND Status = 'Accepted';
```

Aggregate queries :

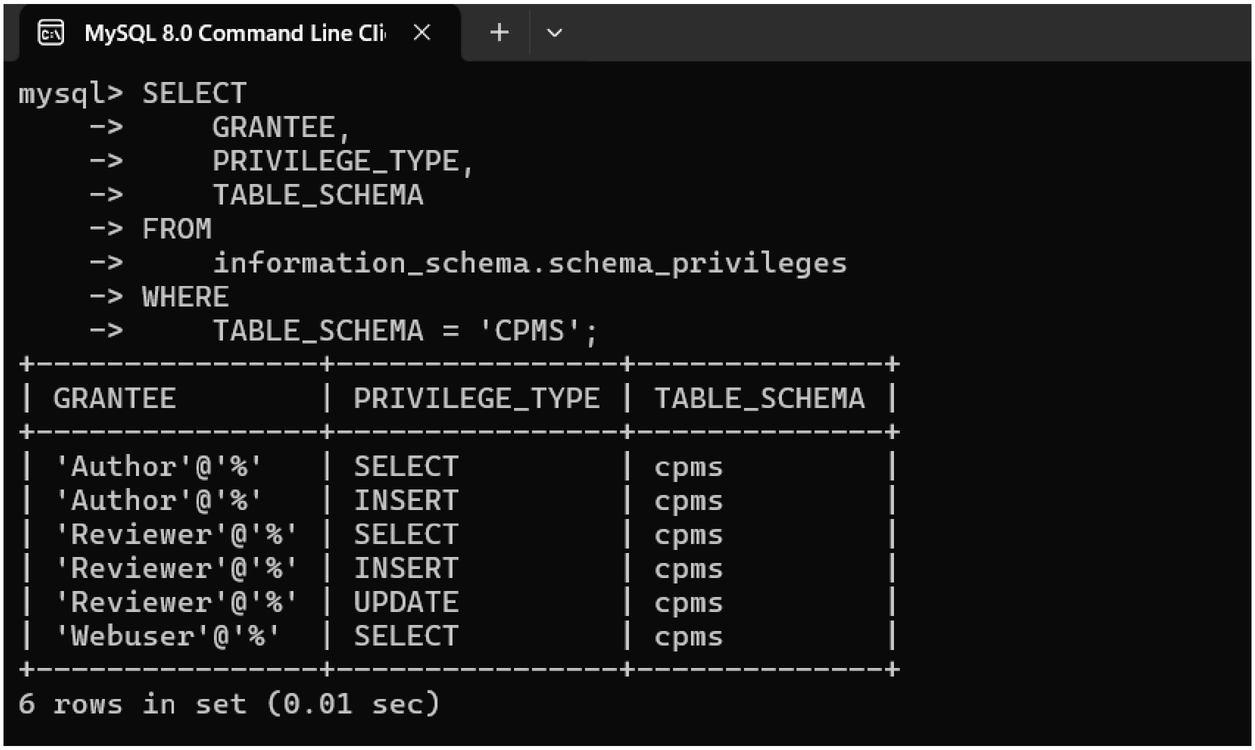


```
-- Query to the count of all accepted papers in the proceedings
SELECT COUNT(*) AS AcceptedPaperCount
FROM Papers
INNER JOIN Paper_Uploads ON Papers.Paper_ID = Paper_Uploads.Paper_ID
WHERE Papers.Conference_ID = conference_id AND Status = 'Accepted';
```

Create Roles / Users :

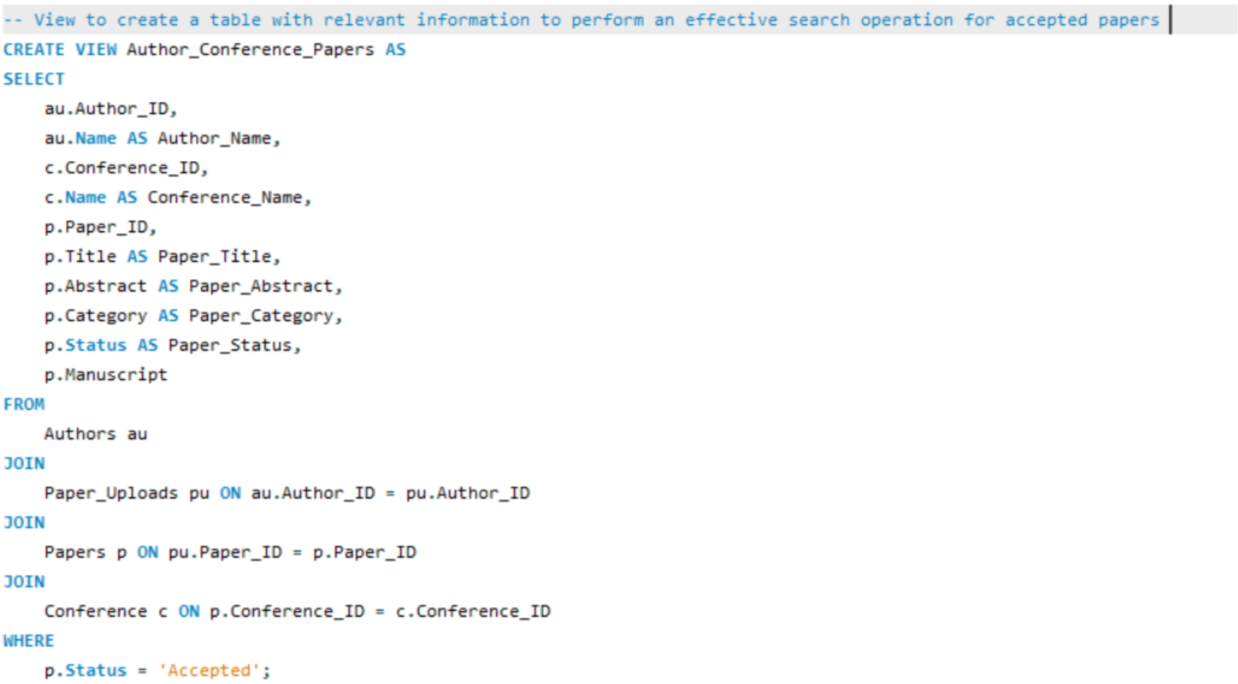


```
CREATE ROLE Webuser;
CREATE ROLE Author;
CREATE ROLE Reviewer;

GRANT SELECT ON CPMS.* TO Webuser;
GRANT SELECT, INSERT ON CPMS.* TO Author;
GRANT SELECT, INSERT, UPDATE ON CPMS.* TO Reviewer;

-- Create a user for Webuser role
CREATE USER 'vedant'@'localhost' IDENTIFIED BY 'vedant';
GRANT Webuser TO 'vedant'@'localhost';

-- Create a user for Author role
CREATE USER 'Alice'@'localhost' IDENTIFIED BY 'password123';
GRANT Author TO 'Alice'@'localhost';

-- Create a user for Reviewer role
CREATE USER 'Michael'@'localhost' IDENTIFIED BY 'revpass123';
GRANT Reviewer TO 'Michael'@'localhost';
```



```
mysql> SELECT
    ->     GRANTEE,
    ->     PRIVILEGE_TYPE,
    ->     TABLE_SCHEMA
    -> FROM
    ->     information_schema.schema_privileges
    -> WHERE
    ->     TABLE_SCHEMA = 'CPMS';
+----------------+----------------+--------------+
| GRANTEE        | PRIVILEGE_TYPE | TABLE_SCHEMA |
+----------------+----------------+--------------+
| 'Author'@'%'   | SELECT         | cpms         |
| 'Author'@'%'   | INSERT         | cpms         |
| 'Reviewer'@'%' | SELECT         | cpms         |
| 'Reviewer'@'%' | INSERT         | cpms         |
| 'Reviewer'@'%' | UPDATE         | cpms         |
| 'Webuser'@'%'  | SELECT         | cpms         |
+----------------+----------------+--------------+
6 rows in set (0.01 sec)
```

View (Virtual Table) :



```sql
-- View to create a table with relevant information to perform an effective search operation for accepted papers
CREATE VIEW Author_Conference_Papers AS
SELECT
    au.Author_ID,
    au.Name AS Author_Name,
    c.Conference_ID,
    c.Name AS Conference_Name,
    p.Paper_ID,
    p.Title AS Paper_Title,
    p.Abstract AS Paper_Abstract,
    p.Category AS Paper_Category,
    p.Status AS Paper_Status,
    p.Manuscript
FROM
    Authors au
JOIN
    Paper_Uploads pu ON au.Author_ID = pu.Author_ID
JOIN
    Papers p ON pu.Paper_ID = p.Paper_ID
JOIN
    Conference c ON p.Conference_ID = c.Conference_ID
WHERE
    p.Status = 'Accepted';
```