**THE UNIVERSITY OF DODOMA**

**COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION**



**CIVE REPOSITORY PROJECT**

GROUP MEMBERS

|  |  |  |
| --- | --- | --- |
| SN | **NAMES** | **ADMISSION NUMBER** |
| 1 | ELIA WILLIAM MARIKI | T22-03-13063 |
| 2 | HAROON AHMED ABDALLAH | T22-03-11755 |
| 3 | FAUDHIA SHARIFU MAKOSA | T22-03-11647 |
| 4 | LEONIDA GERVAS LAURENT | T22-03- |
| 5 | JUMANNE SHABAN JUMA | T22-03-05988 |
| 6 | MYANO MALUGU NGUSA | T22-03-07219 |
| 7 | NEEMA GODFREY KISANGA | T22-02-03401 |
| 8 | NIMFA ALPHONCE MASUMBUKO | T22-03-14258 |
| 9 | SWEETBERTH SANDE MWAMPASHI | T22-03-07634 |
| 10 | RAMADHAN ABDALLAH SAID | T22-03-02494 |

**CIVE REPOSITORY**

The “CIVE REPOSITRY” is a web application designed to provide users(students) with easy access to a wide range of educational materials .This repository serves as a system where users can explore and contribute various resources such as lecture notes, past papers, relevant links and recommended books . It aims to enhance the learning experience by offering organized contents that aligns with the curriculum of the College of Informatics and virtual Education. Whether you are looking to study, prepare for exams, or share valuable resources, this repository enhance collaboration and knowledge exchange within the college community.

**FUNCTIONAL REQUIREMENTS**

System User(student) Functional Requirement:

1. User Registration and Authentication:

* Users can register with the system.
* Users can log in to the system using their credentials.
* During registration, users can input their full name, registration number, email, profile picture, and phone number.

2. User Profile Customization:

* Users can customize their profile settings.
* Users can change their profile picture.
* Users can change their password.

3. Content Upload and Viewing:

* Users can upload notes, books, links, and past papers.
* Users can upload documents in PDF format.
* Users can view uploaded materials such as notes, books, links, and past papers.

4. Comment Functionality:

* Users can send comments on uploaded materials.
* Users can read comments on uploaded materials.
* Users can delete their own comments.

5. Logout:

* Users can log out of the system after using it.

**System Admin Functional Requirement:**

1. Admin Post Management:

* Admins can set the status of a post as available or not.
* Admins can terminate a user's access to the system.
* Admins can view user information and their current status.
* Admins can create special accessory posts for student use.

2. Root Admin Control:

* Root admin can change the status of other admins, allowing or revoking their admin privileges.

3. Admin Post Creation:

* Admins can create posts within the system.

4. Admin Registration and Authorization:

* Admins can register with the system by providing full name, email, username, password, and role.
* Admin registration requires permission from the root admin.

5. Admin Login:

* Admins can log in to the system using their credentials.

6. Customization:

* Admins can customize their profile by changing their profile picture and password.

7. Logout:

* Admins can log out of the system after using it.

8. Post Approval/Rejection:

* Admins can approve or reject posts submitted by normal users (students).
* Approved posts become visible to all users.
* Rejected posts will not be seen by other users.

**System Functional Requirements:**

1. Search Functionality:

* Users can search for notes, books, and past papers.

1. Input:

* Users can log in by providing their email address and password.

1. Output:

* Users can view and access uploaded resources, including notes, books, links, and past papers.

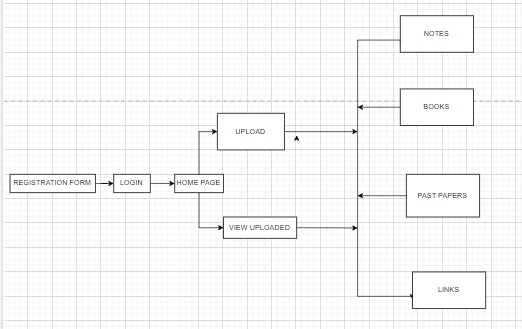
**Required System Inputs:**

I. The system requires the following to be uploaded:

* Notes
* Links
* Books
* Past papers

II. Who Can Upload:

* Admins can upload.
* Users can upload.

**A Functional Flow Block Diagram (FFBD)**

**Project Timeline (3 Weeks):**

WEEK 1:

**Milestone 1: design database and documentation of the whole system;**

Implementation Steps:

* Requirement gathering
  + User functional requirement (students)
  + System admin functional requirement
  + System requirement
* Database design and implementation
  + ER diagram
  + Logical schema
  + Physical schema
  + Data dictionary and documentation
* Functional flow block Diagram (FFBD)
* Designing of the user interface.
  + Design the user registration and login form with fields for full name, registration number, email, phone number, and password.
  + Design of the user dashboard and profile settings page.
  + Design the content upload interface for notes, books, links, and past papers in PDF format.
  + Design the comment section for uploaded materials.

WEEK 2:

**- Milestone 2: User Registration and Authentication ;**

Implementation Steps:

* Implement user registration back-end logic, including secure storage of user credentials.
* Create the login page with registration number and password fields.
* Develop authentication logic to verify user credentials and create session management.

- **Milestone 3: User Profile Customization**

Implementation Steps:

* Enable users to upload and change their profile pictures.
* Implement functionality for users to change their passwords.

- **Milestone 4: Content Upload and Viewing**

Implementation Steps:

* Implement back-end logic to handle document uploads securely.
* Create a page to display uploaded materials.

WEEK 3:

**- Milestone 5: Comment Functionality**

Implementation Steps:

* Implement back-end logic to store and retrieve comments associated with materials.
* Allow users to send, read, and delete their comments.

- **Milestone 6: Logout**

Implementation Steps:

* Implement the logout functionality to end the user's session securely.

PROGRESS TRACKING CHECKLIST:

|  |  |  |
| --- | --- | --- |
| SN | Task | status |
| 1 | Design database and documentation |  |
| 2 | User registration and authentication |  |
|  | User registration form design and implementation |  |
|  | Login page design and authentication logic |  |
| 3 | User profile customization |  |
|  | User profile settings design and functionality |  |
| 4 | Content upload and viewing |  |
|  | Content upload interface design and back-end logic |  |
|  | Display page for uploaded materials |  |
| 5 | Comment functionality |  |
|  | Comment section design and implementation |  |
| 6 | logout |  |
|  | Secure logout functionality |  |

**TASK SHARING PLAN FOR THE TEAM:**

* Designer:(RAMADHAN,NIMPHA,NEEMA,SWEETBERT)

Design user registration form, profile settings page, content upload interface, comment section, and logout page,data base.

* Front-end Developer:(FAUDHIA,HAROON,JUMANNE,Ngusa)

Implement front-end components for user registration, profile customization, content upload, comments, and logout.

* Back-end Developer:(ALL MEMBERS)

Implement back-end logic for user registration, authentication, content upload, comments, and logout.

* Tester:(LEONIDA)

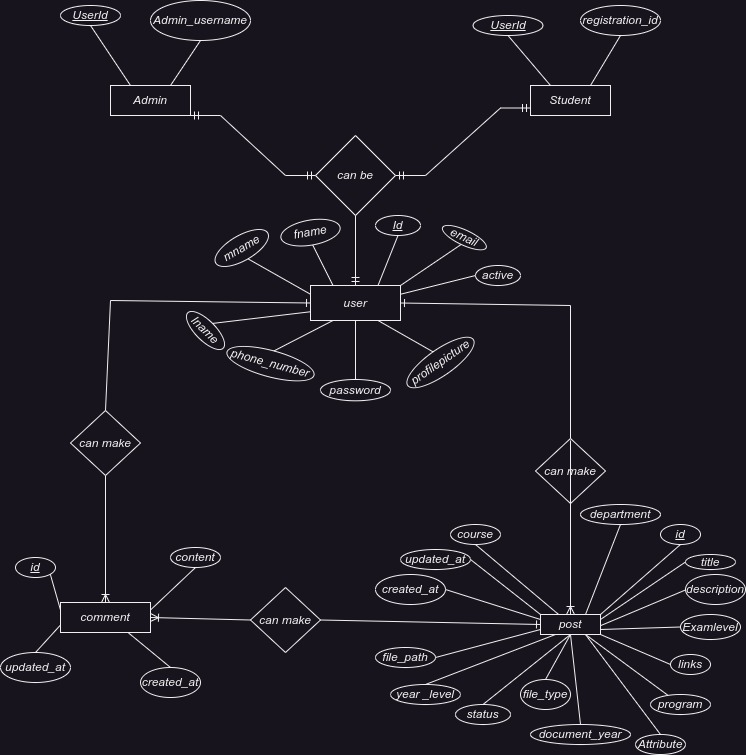
Test user registration, authentication, content upload, comments, and logout functionality.

* Project Manager:(ELIA WILLIAM MARIKI)

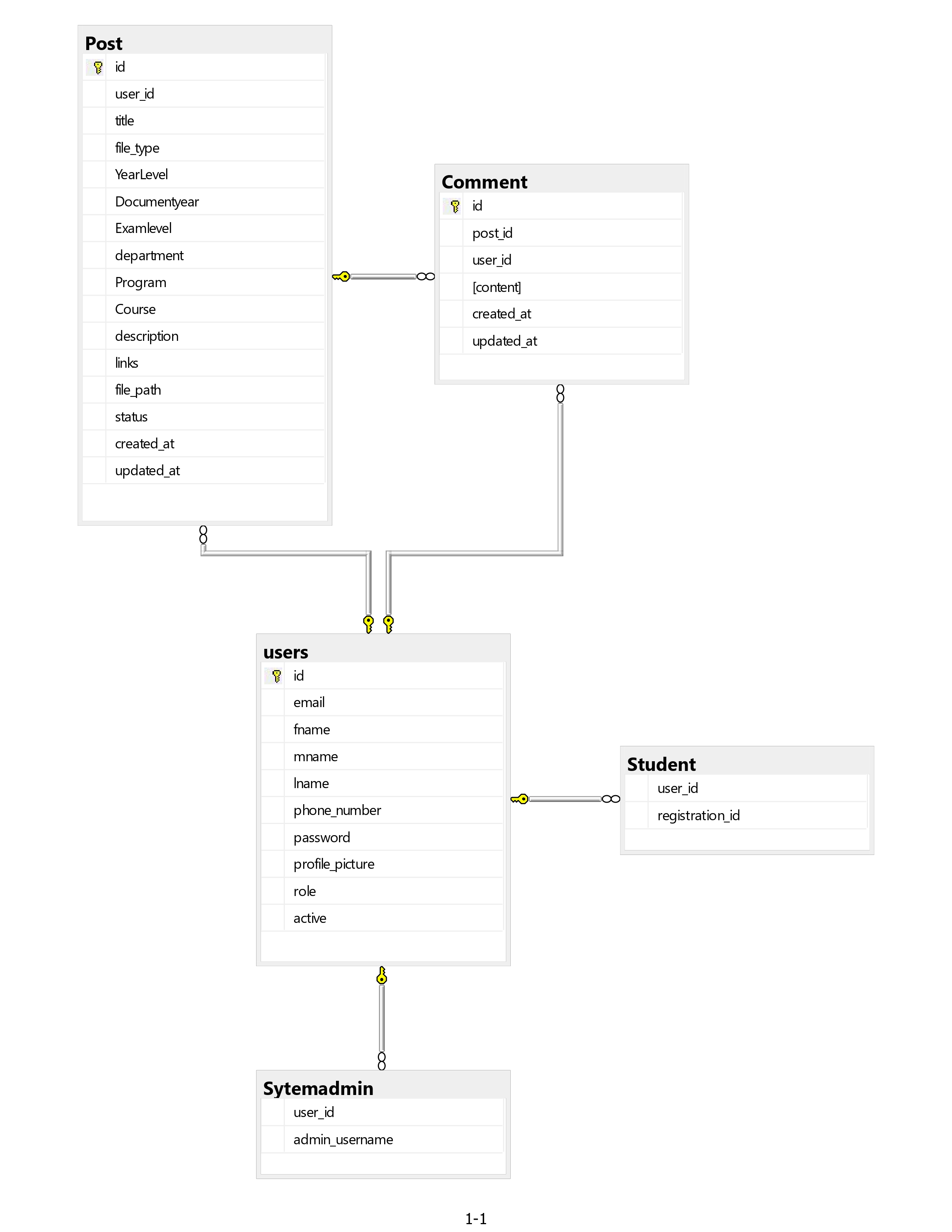
Oversee the project's progress, coordinate tasks, and ensure milestones are met.

**Database and documentation:**

**ER-DIAGRAM**

****

**LOGICAL SCHEMA:**

****

**PHYSICAL SCHEMA:**

CREATE DATABASE CIVE REPOSITORY;

USE REPOSITORY\_ONE;

CREATE TABLE A\_user(

id INT IDENTITY (1,1)NOT NULL,

email VARCHAR(50) NOT NULL,

fname VARCHAR(25) NOT NULL,

mname VARCHAR(25) NOT NULL,

lname VARCHAR(25) NOT NULL,

phone\_number INT NOT NULL,

password VARCHAR(20) NOT NULL,

profile\_picture varchar(255),

role VARCHAR(10) CHECK(role IN ('student', 'admin')),

active BIT,

PRIMARY KEY (id)

);

CREATE TABLE Student (

user\_id INT NOT NULL,

registration\_id VARCHAR (10) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES users (id)

);

CREATE TABLE Sytemadmin (

user\_id INT NOT NULL,

admin\_username VARCHAR(255) NOT NULL,

FOREIGN KEY (user\_id) REFERENCES users (id)

);

CREATE TABLE Post (

id INT IDENTITY (1,1) NOT NULL,

user\_id INT NOT NULL,

title VARCHAR(255) NOT NULL,

file\_type VARCHAR(255) NOT NULL, --((for pastpaper, notes, or program)

YearLevel int CHECK (YearLevel BETWEEN 1 AND 4),

Documentyear varchar(4) NOT NULL,

Examlevel varchar(20) CHECK (Examlevel IN ('Test 01', 'Test 02', 'UE')), --(for pastpaper only)

department VARCHAR(255),

Program varchar(100),

Course varchar(100),

description text, --(used for accessories,tutorials,installation,books)

links VARCHAR(255), --(used for books,tutorial,accessories,setup)

file\_path varchar(255),

status VARCHAR(255) NOT NULL, --(pending, approved, rejected)

created\_at DATETIME NOT NULL,

updated\_at DATETIME NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (user\_id) REFERENCES users (id)

);

CREATE TABLE Comment (

id INT IDENTITY (1,1)NOT NULL,

post\_id INT NOT NULL,

user\_id INT NOT NULL,

content TEXT NOT NULL,

created\_at DATETIME NOT NULL,

updated\_at DATETIME NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (post\_id) REFERENCES posts (id),

FOREIGN KEY (user\_id) REFERENCES users (id)

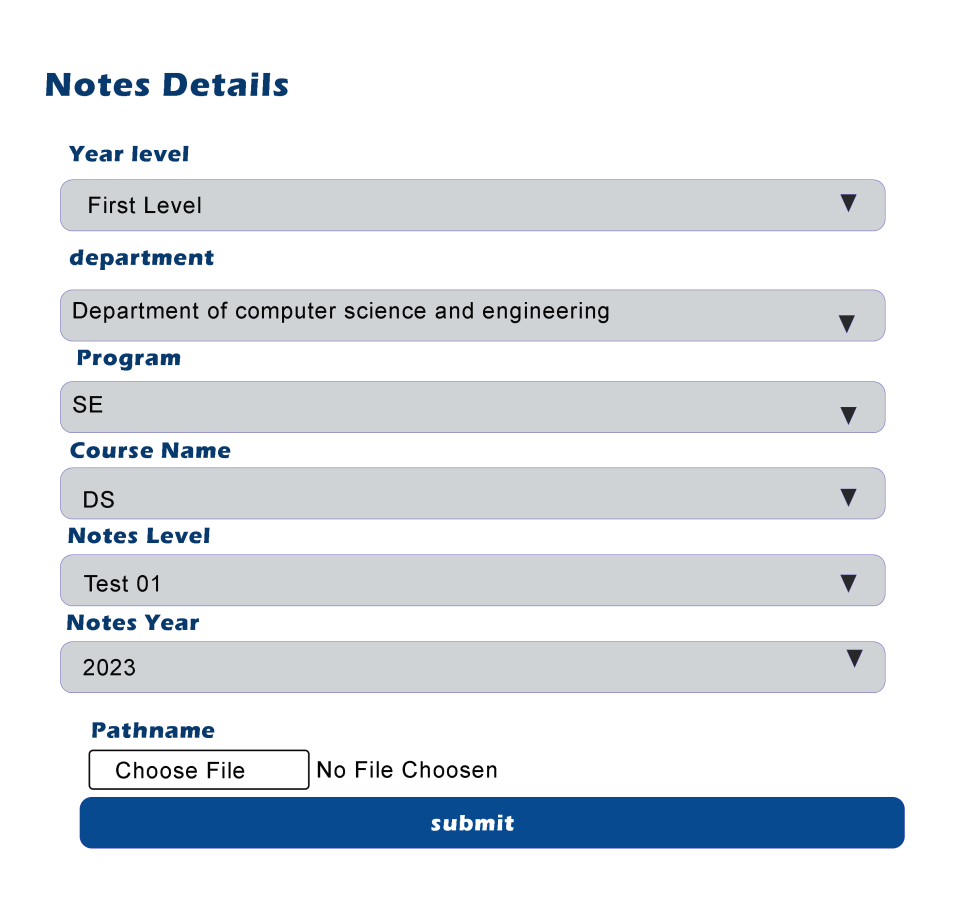
);

**DATA DICTIONARY**

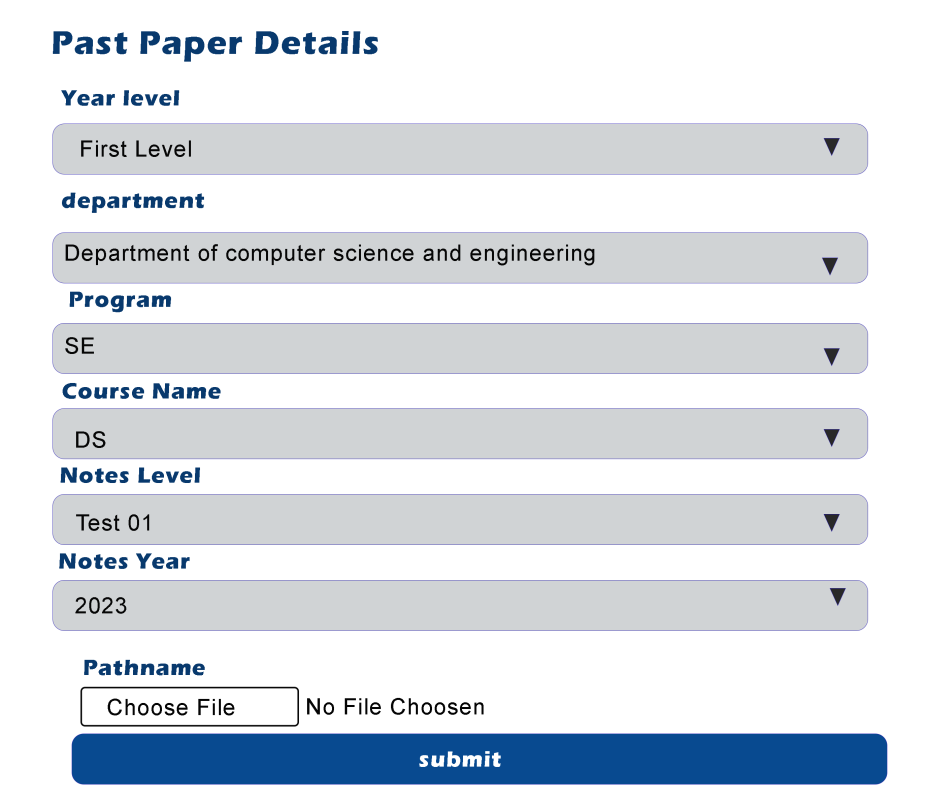
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TABLE NAME | ATTRIBUTE | CONSTRAINTS | SIZE | DATA TYPE | DESCRIPITION |
| USER | id | PRIMARY KEY |  | INT | NOT NULL EACH USER (ADMIN/STUDENT)MUST BE GIVEN AN UNIQUE ID |
|  | Email |  | 50 | VARCHAR | NOT NULL EACH USER (ADMIN/STUDENT)MUST HAVE A UNIQUE EMAIL |
|  | Fname |  | 25 | VARCHAR | NOT NULL EACH USER MUST ENTER HIS /HER FIRST NAME |
|  | Mname |  | 25 | VARCHAR | EACH USER MUST ENTER HIS /HER MIDDEL NAME |
|  | Lname |  | 25 | VARCHAR | NOT NULL EACH USER MUST ENTER HIS /HER LAST NAME |
|  | Phone\_number |  |  | INT | NOT NULL EACH USER MUST ENTER HIS /HER PHONE NUMBER |
|  | Password |  | 20 | VARCHAR | NOT NULL EACH USER MUST ENTER HIS /HER PASSWORD |
|  | Profile\_picture |  | 255 | VARCHAR | EACH USER MAY UPLOAD HIS/HER PROFILE PICTURE |
|  | Role |  | 10 | VARCHAR | CHECK(role in (student admin) |
|  | active |  |  | BIT | NOT NULL  THE USER MAY BE ACTIVE OR NOT |
| STUDENT | User\_id | FOREIGN KEY |  | INT | NOT NULL  EACH USER SHOULD HAVE HIS /HER ID |
|  | Registration\_ID |  | 25 | VARCHAR | EACH STUDENT SHOLD USE HIS /HER OWN ID TO LOGIN |
| ADMINS | User\_id | FOREIGN KEY |  | INT | NOT NULL  EACH USER(ADMIN) SHOULD HAVE HIS /HER ID |
|  | Admin\_username |  | 20 | VARCHAR | NOT NULL  ADMIN MUST ENTER HIS /HER VALID USER NAME TO LOGIN |
| POSTS | Id | PRIMARY KEY |  | INT | NOT NULL  EACH POST IS GIVEN A UNIQUE ID |
|  | Title |  | 150 | VARCHAR | NOT NULL  EACH POST MUST HAVE A TILE |
|  | File\_type |  | 50 | VARCHAR | NOT NULL  A FILE MAYBE A PAPER,NOTES,TUTORIAL AND BOOKS |
|  | YearLevel |  |  | INT | NOT NULL  CHECK(yearLevel between 1 and 4) |
|  | DocumentYear |  |  | INT | NOT NULL  THE UPLOADED DOCUMENT MUST HAVE A SPECIFIC YEAR |
|  | ExamLevel |  | 20 | VARCHAR | NOT NULL  CHECK(ExamLevel in (test1,test2 and UE)) |
|  | Department |  | 50 | VARCHAR | NOT NULL  USER MUST SPECIFY THE DEPARTMENT OF THE POSTS THAT HE/SHE POSTED |
|  | Program |  | 20 | VARCHAR | USER MUST SPECIFY THE PROGRAM OF THE POSTS THAT HE/SHE POSTED |
|  | Course |  | 20 | VARCHAR | USER MUST SPECIFY THE COURES OF THE POSTS THAT HE/SHE POSTED |
|  | Description |  |  | TEXT | USER PROVIDES INFORMATION ABOUT THE POSTED MATERAL |
|  | Links |  | 100 | VARCHAR | THE USER PROVIDES A LINK FOR BOOKS,TUTORIAL AND SETUPS |
|  | FilePath |  | 50 | VARCHAR | NOT NULL THE PASS OF THE UPLOADED FILE |
|  | Status |  | 10 | VARCHAR | NOT NULL THE STATUS OF THE POST IS EITHER PENDING,APPROVED,REJECTED |
|  | Created\_at |  |  | DATETIME | NOT NULL  EACH POST CREATED MUST HAVE A DATE AND TIME |
|  | Updated\_at |  |  | DATETIME | NOT NULL  UPDATED POST MUST HAVE A DATE AND TIME |
|  | User\_id | FOREIGN KEY |  | INT | NOT NULL  EACH POST MUST ATTAIN ID FROM THE USER |
| COMMENTS | id | PRIMARY KEY |  | INT | NOT NULL  EACH COMMENT SHOULD HAVE A UNIQUE ID |
|  | Post id | FOREIGN KEY |  | INT | NOT NULL EACH COMMENT MUST ATTAIN POST ID |
|  | User\_id | FOREIGN KEY |  | INT | NOT NULL EACH COMMENT MUST ATTAIN A USER ID |
|  | Created\_at |  |  | DATETIME | NOT NULL  EACH COMMENT CREATED MUST HAVE A DATE AND TIME |
|  | Updated\_at |  |  | DATETIME | NOT NULL  UPDATED COMMENT MUST HAVE A DATE AND TIME |

**Content upload interface design:**

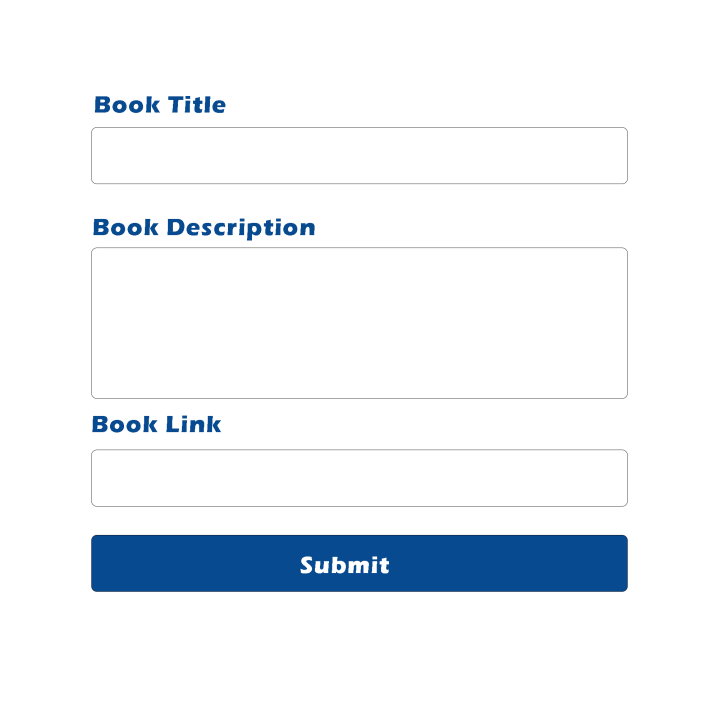
**1.Interface for notes Upload:**

****

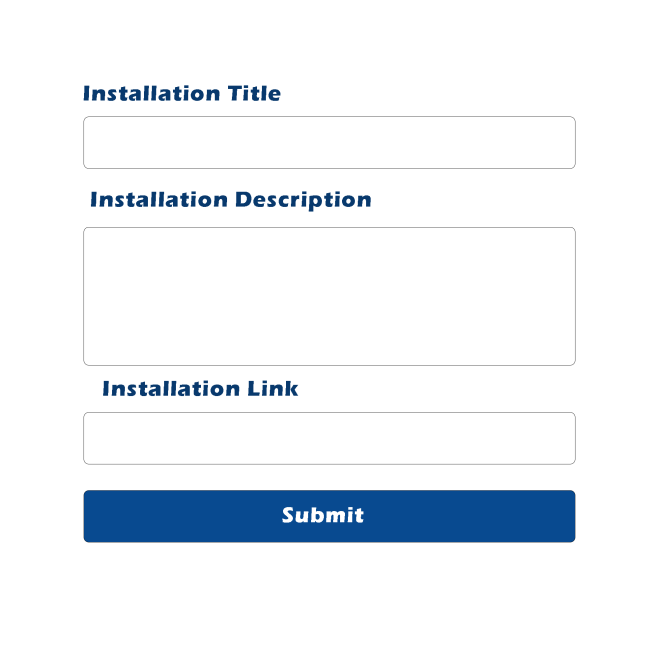
**2.Interface for pastpaper Upload:**

****

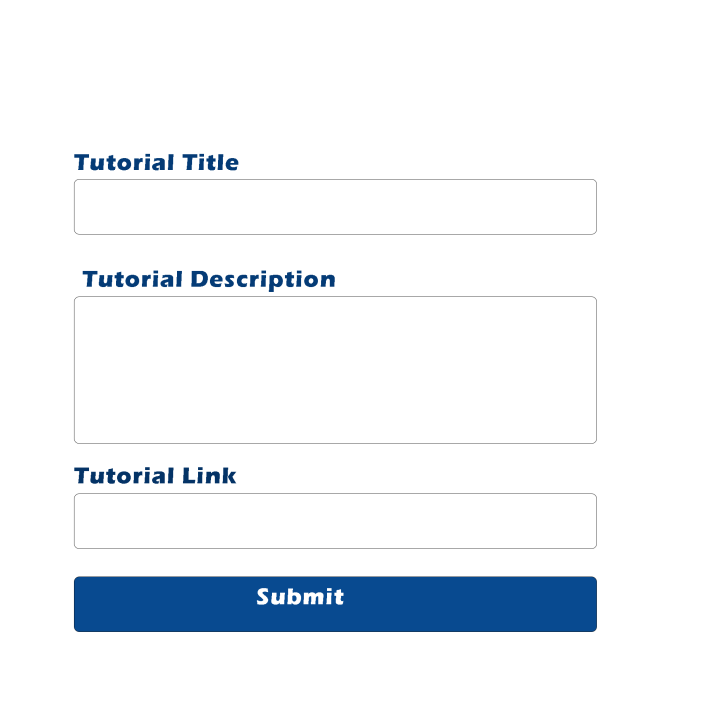
1. **Interface for books-upload**

****

1. **Interface for installation information upload**

****

1. **Interface for Tutorial information upload**

****