# Database Design & Project Workflow

## 1. Project Workflow Overview

1. Sign In / Sign Up  
2. Manage Users (Insert, Edit, Delete)  
3. Register Project (done by Supervisor)  
4. Explore Projects (view all registered projects)  
5. Select Project (Student selects with justification)  
6. Approve Project (Supervisor approves/rejects with comments)  
7. Submit Work Product (multiple versions per project)  
8. Give Feedback (Supervisor gives feedback as file or text)  
9. Receive Feedback (Student views feedback)

**2. Table: Projects**

|  |
| --- |
| Stores the list of registered projects. Supervisor registers projects. |
| |  |  |  | | --- | --- | --- | | Column Name | Data Type | Description | | project\_id | INT (PK, AUTO\_INCREMENT) | Primary key | | title | VARCHAR(100) UNIQUE | Project title | | description | TEXT | Project description | | justification | TEXT | Why this project is needed | | languages | TEXT | Languages required | | academic\_requirements | TEXT | Academic needs | | cgpa | DECIMAL(3,2) | Required CGPA | | credit\_hours | INT | Required credit hours | | submission\_date | DATETIME DEFAULT CURRENT\_TIMESTAMP | Auto-set submission date | |
|  |

## 3. Table: project selections

Stores project selections made by students and approval status.

|  |  |  |
| --- | --- | --- |
|  | | |
| Column Name | Data Type | Description |
| selection\_id | INT (PK, AUTO\_INCREMENT) | Primary key |
| project\_id | INT (FK) | Linked to projects table |
| student\_id | INT (FK) | Linked to users table |
| justification | TEXT | Student's justification |
| status | ENUM('pending','approved','rejected') | Selection status |
| supervisor\_comment | TEXT | Supervisor’s comment |

- project\_id is auto-generated and NOT part of the form.  
- Supervisor registers the project.  
- Student selects from existing projects.  
- Supervisor can comment during approval/rejection.  
- Use foreign key relationships to maintain consistency across screens.

**1. users – For Sign In / Sign Up / Manage User**

CREATE TABLE users

(

user\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(100) NOT NULL,

role ENUM('student', 'supervisor', 'admin') NOT NULL,

reg\_no VARCHAR(30) UNIQUE -- for students

);

* **Screens using this**: Sign in, Sign up, Manage user
* Admin can edit/delete users

**2. Projects – For Register Project & Explore Projects**

CREATE TABLE projects

(

project\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(100) NOT NULL UNIQUE,

description TEXT NOT NULL,

languages TEXT,

academic\_requirements TEXT,

supervisor\_id INT,

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

);

* Registered by **supervisor**.
* **Used in**: Register Project, Explore Project

**3. Project Selections – For Select Project (by student)**

CREATE TABLE project\_selections (

selection\_id INT PRIMARY KEY AUTO\_INCREMENT,

project\_id INT,

student\_id INT,

justification TEXT NOT NULL,

status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',

FOREIGN KEY (project\_id) REFERENCES projects(project\_id),

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

);

* **Used in**: Select Project screen, Approve Project screen

**4. Project\_versions – For Submit Work (multiple versions)**

CREATE TABLE project\_versions (

version\_id INT PRIMARY KEY AUTO\_INCREMENT,

selection\_id INT,

version\_name VARCHAR(50), -- e.g., 'Proposal', 'SRS', etc.

file\_path VARCHAR(255), -- store path to the uploaded file

submission\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (selection\_id) REFERENCES project\_selections(selection\_id)

);

* **Used in**: Submit Work screen

**5. Feedback – For Give Feedback and Receive Feedback**

CREATE TABLE feedback (

feedback\_id INT PRIMARY KEY AUTO\_INCREMENT,

version\_id INT,

Version\_name TEXT,

supervisor\_id INT,

feedback\_text TEXT,

feedback\_file VARCHAR(255),

feedback\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (version\_id) REFERENCES project\_versions(version\_id),

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

);

* Can include text and File
* **Used in**: Give Feedback, Receive Feedback