Design Manual

Project 2: DB backend web application

University management system

Team: A

Swati kar

Poorna Raavi

Shiva Prasad Reddy Govindagari

Sowjanya Butukuru

Vardhan Kumar Pokala

Introduction:

Welcome to the University Management System (UMS) design manual. This manual serves as a comprehensive guideline on how the system is built. It contains three major sections :

- Design Overview
- E-R diagram and database schemas
- Security assurance process

Design Overview:

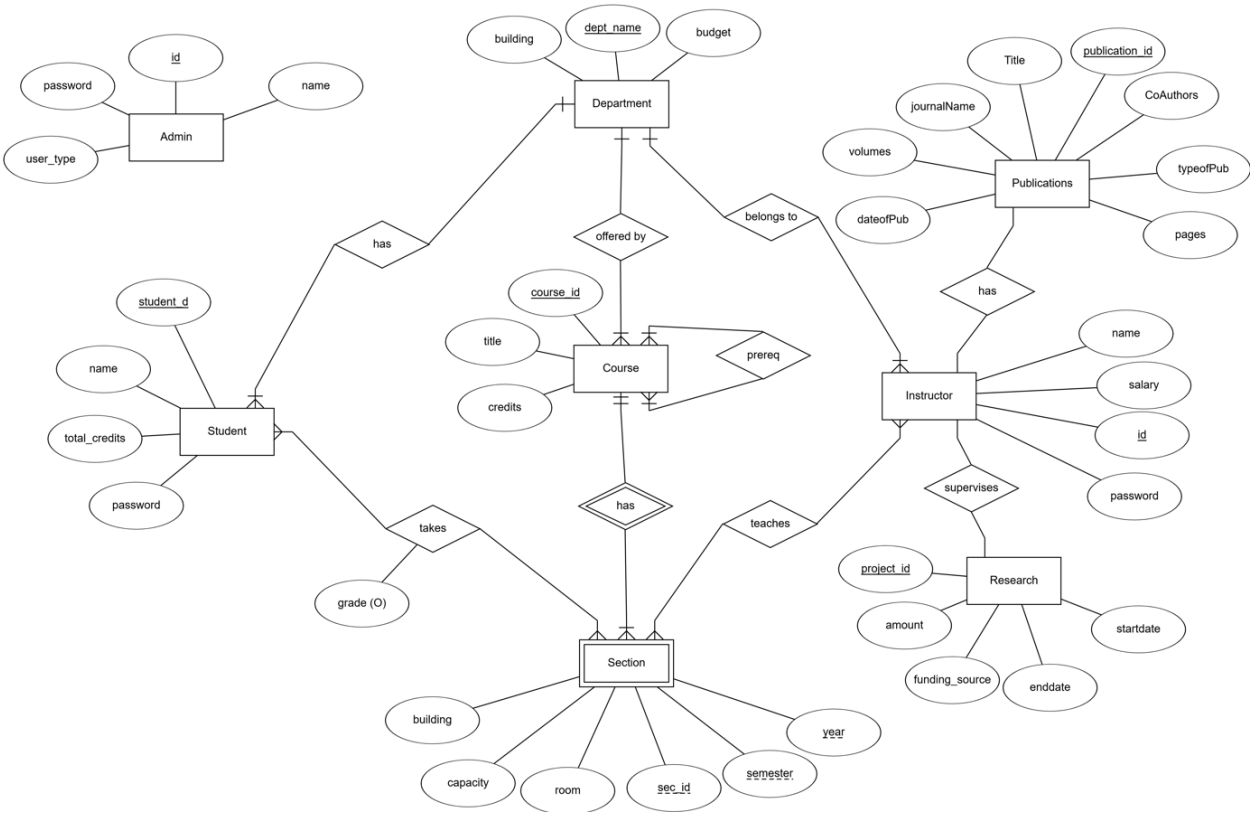
System goals and required functionalities

- University Management System with authorized credentials
- Easy access for end user
- Fast and efficient system
- Prone to security attack
- Six features (F1-F6) fully implemented

Design Process

- Connect database and Django using settings.py
- Add new tables that are required for some of the features
- Update model.py so that it can be compatible with all six features
- Generate URLs for every features
- Update view.py to process input and execute SQL queries
- Design Django templates for user interface
- Implemented security checks
- Design queries that are needed for features

ER Diagram



Database Schema

For our project, we have used ‘University’ database that was created in the class. Along with the existing tables, we have made some updates:

- Added ‘Admin’ table
- Added ‘Publication’ table
- Added ‘Research’ table
- Added ‘password’ column in the existing ‘Instructor’ table
- Added ‘password’ column in the existing ‘Student’ table

In addition to these tables, Django framework has created its own internal table schema for several functionalities.

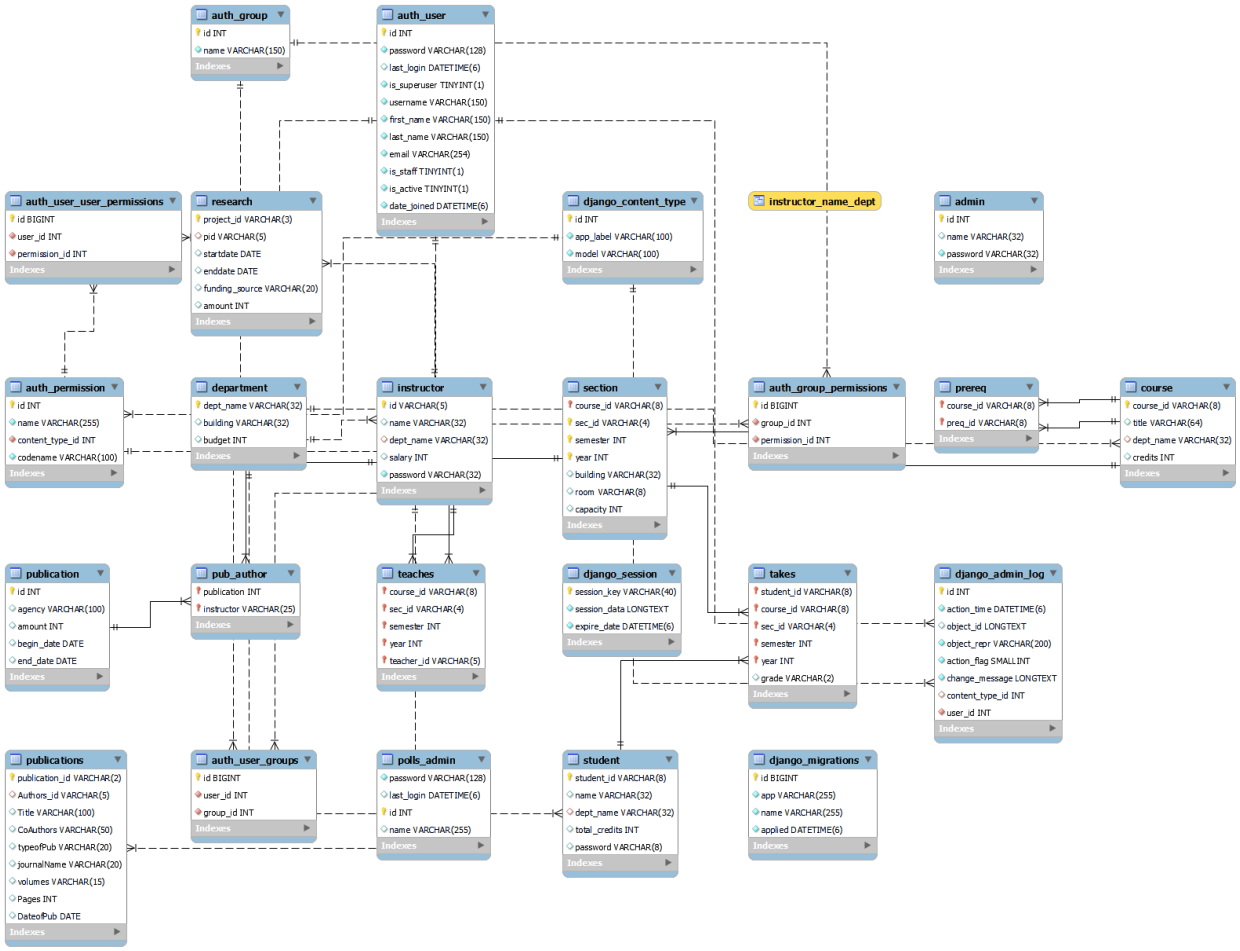
Below is the description of the newly created tables:

```
mysql> describe admin;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int         | NO   | PRI | NULL    | auto_increment |
| user_type | varchar(32) | YES  |     | NULL    |                |
| name      | varchar(32) | YES  |     | NULL    |                |
| password  | varchar(32) | NO   |     | 1234    |                |
+-----------+-------------+------+-----+---------+----------------+
```

```
mysql> describe publications;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| publication_id | varchar(2)   | NO   | PRI | NULL    |       |
| Authors_id     | varchar(25)  | YES  | MUL | NULL    |       |
| Title          | varchar(100) | YES  |     | NULL    |       |
| CoAuthors      | varchar(50)  | YES  |     | NULL    |       |
| typeofPub      | varchar(20)  | YES  |     | NULL    |       |
| journalName    | varchar(20)  | YES  |     | NULL    |       |
| volumes        | varchar(15)  | YES  |     | NULL    |       |
| Pages          | int          | YES  |     | NULL    |       |
| DateofPub      | date         | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

```
mysql> describe Research;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| project_id     | varchar(3)  | NO   | PRI | NULL    |       |
| pid            | varchar(5)  | YES  | MUL | NULL    |       |
| startdate      | date        | YES  |     | NULL    |       |
| enddate        | date        | YES  |     | NULL    |       |
| funding_source | varchar(20) | YES  |     | NULL    |       |
| amount         | int         | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

Schema Diagram



Security Assurance Process:

Cross-Site Request Forgery (CSRF) protection

CSRF is a type of security vulnerability in web applications where an attacker tricks a user's browser into executing unwanted actions on a web site to which the user is authenticated. To mitigate this risk, sites use CSRF tokens, which are random, hard-to-guess values that are generated by the server and passed to the client. They are usually included in forms as hidden inputs. When a form is submitted, the server checks if the CSRF token received from the client matches the one it expects. This confirmation helps ensure that the request is legitimate and originated from the site itself, not an external source.

Below is the screenshot of using CSRF token:

```
<form class="login-form" method="post" action="{% url 'login' %}">
    {% csrf_token %}
    <div class="form-group">
        <select class="form-control" id="login-as" name="login_as">
            <option value="">Select Login As</option>
            <option value="admin">Admin</option>
            <option value="instructor">Instructor</option>
            <option value="student">Student</option>
        </select>
    </div>
    <div class="form-group">
        <input type="text" class="form-control" id="admin_name" name="admin_name" placeholder="ID" required>
    </div>
    <div class="form-group">
        <input type="password" class="form-control" id="password" name="password" placeholder="Password" required>
    </div>
    <div class="checkbox">
        <label>
            <input type="checkbox" name="remember"> Remember me
        </label>
    </div>
    <button type="submit" class="login-button btn-block">LOG IN</button>
```

User Authentication and Session Management:

The application uses Django's built-in authentication mechanisms to handle user logins, session management, and logouts. This ensures that user credentials are verified before granting access to sensitive functionality within the application.

Below is the screenshot of using user authentication and session management:

```
if login_as == 'admin':
    try:
        admin = Admin.objects.get(name=admin_name)
        if admin.password == admin_password:
            request.session["login_as"] = login_as
            request.session["admin_name"] = admin_name
```