REGISTER FORM REPORT

*Project Title: Developing a Student Website with Course Registration Functionality*

Lecturer: Pham The Bao

Students: Le Phuong Hanh

Le Viet Ha

Mai Ngoc Diem My

Nguyen Ba Khanh Huyen

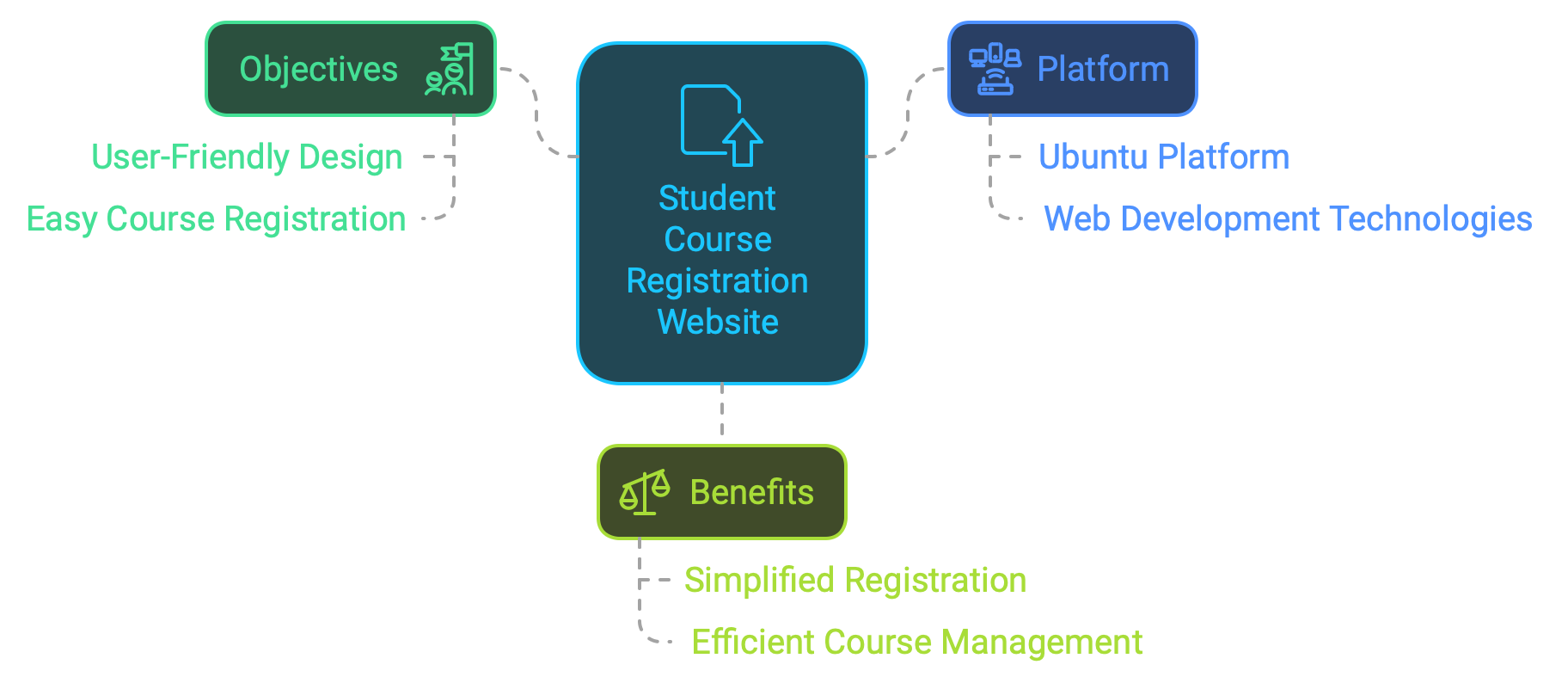
Nguyen Ngoc Gia Vinh

Nguyen Quang Nhat

# **Project overview**

**Objective:**

* The aim of this project is to design and implement a user-friendly website for students to register for courses through a website built on the ubuntu platform. This project will help students register and manage their course registrations easily.



*This website is designed to aid students register for courses.*

**Project Scope:**

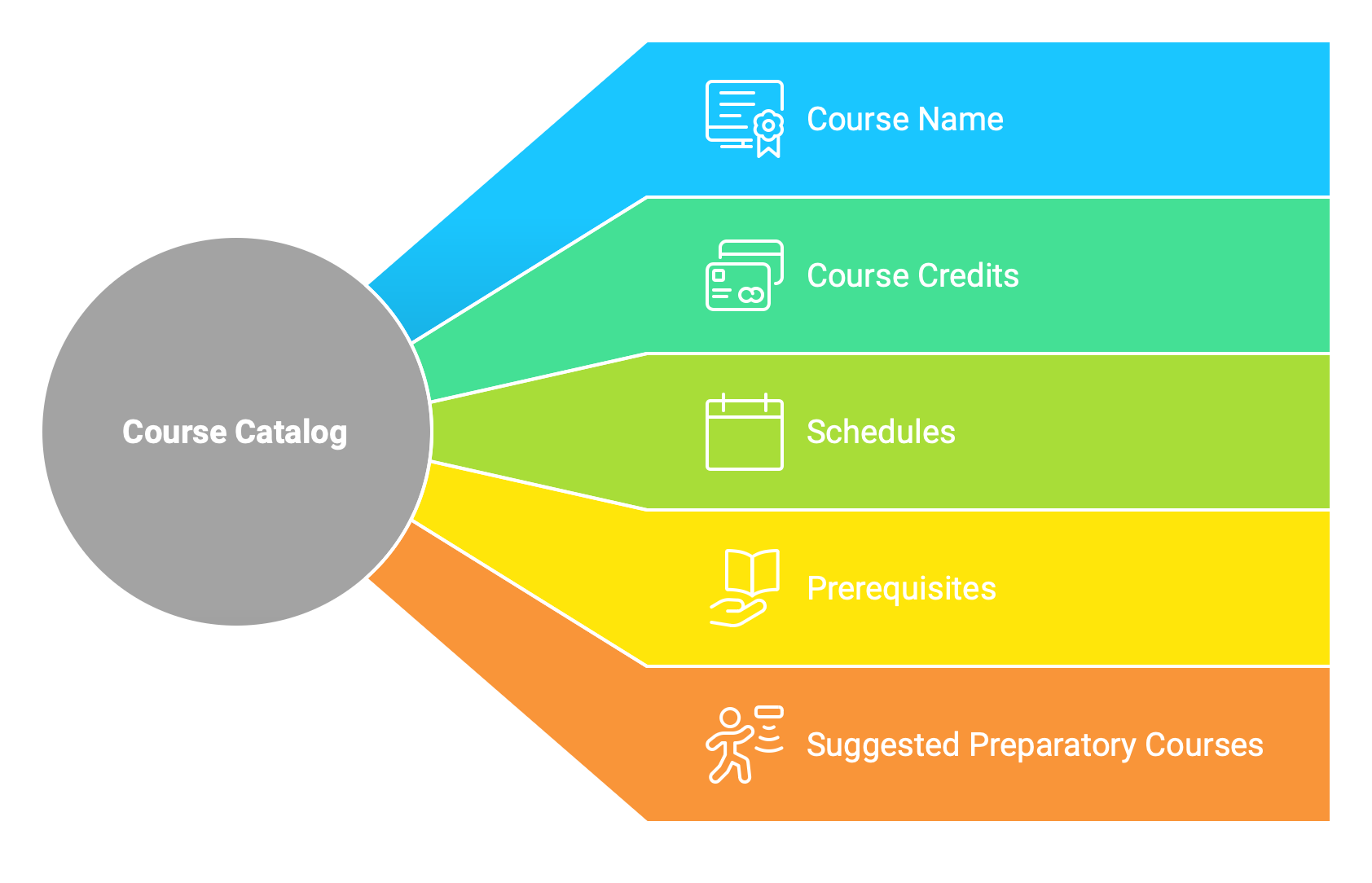
The website will include the following key features:

* *Student Registration and Login:*
* Secure registration and login system to manage student records.
* Integrated encryption to securely handle sensitive data.



*Secure system and integrated encryption.*

* *Course Catalog:*
* A dynamic catalog of available courses, providing detailed information about each course details such as course name, course credits, schedules, prerequisites, and suggested preparatory courses for enrollment.



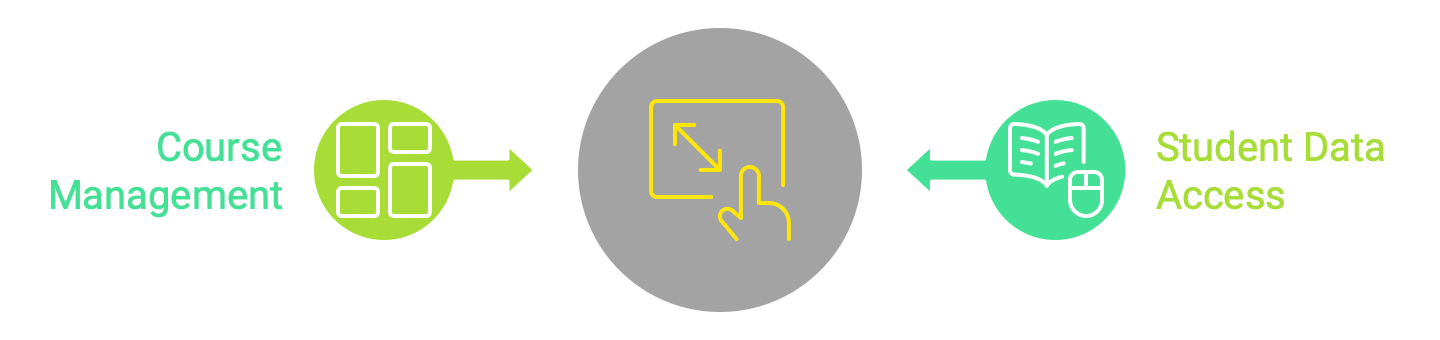
*The data contained in the course catalogue.*

* *Course Registration Form:*
* Intuitive and responsive form for students to select and register for their desired courses.
* Authentication to ensure accurate and complete data.



*Course registration process.*

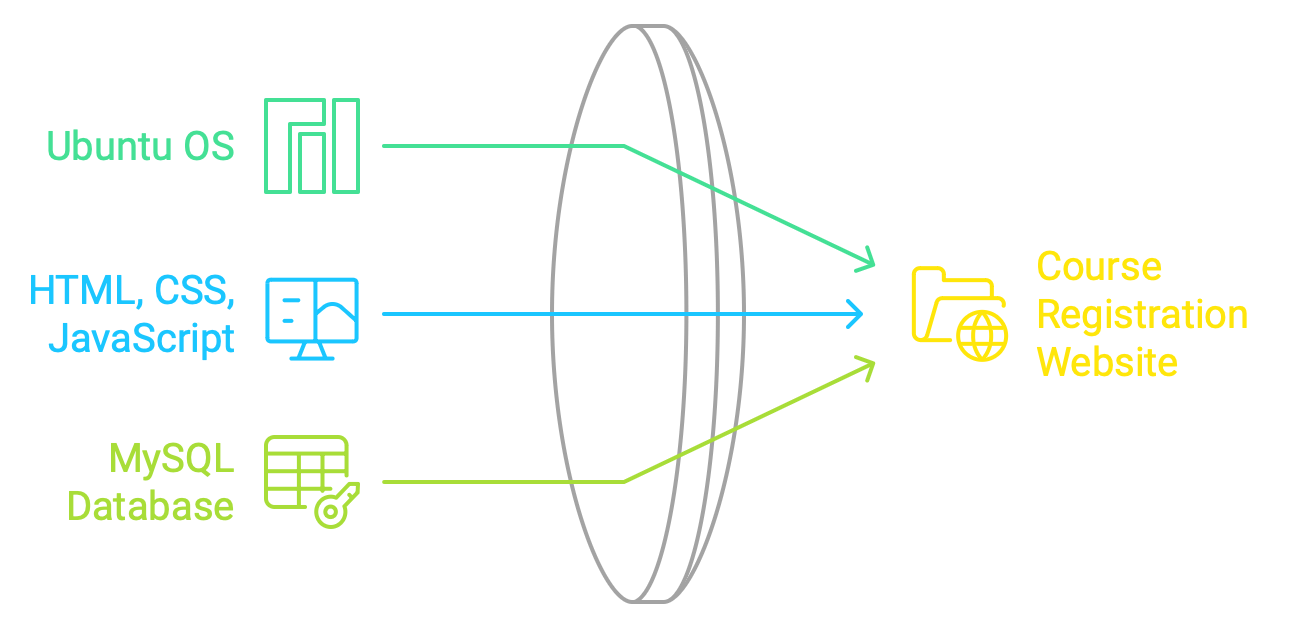
* *Admin Dashboard:*
* Management system for administrators to add, edit or delete courses.
* Access student registration data and analytics to make effective decisions.



*Effective Courses Management System.*

**Tools used:**

* Operating system: Ubuntu (to take advantage of flexibility and developer-friendly ecosystem).
* Programming languages: HTML, CSS, JavaScript (frontend);
* Database management: MySQL for managing data (backend).



*Technical Stack Overview.*

# **System outline and task list**

## **Register system outline**

**User system:**

**❖ Users:**

➢ Students

➢ Manager

➢ Register office

* Business Administration Admin.
* Information Technology Admin.
* Hospitality Administration Admin.

**❖ Departments:**

➢ Business Administration.

➢ Information Technology.

➢ Hospitality Administration.

**❖ Course Types:**

➢ Major-specific courses (BA, IT, HA).

➢ General subjects (G).

**Database Design:**

**❖ Manager**

**❖ Register office**

**❖ Department**

**❖ Next semester course**

**❖ Student Table:**

➢ Student\_ID (primary key)

**➢** Student\_name

➢ Student\_email

➢ Password

➢ Department ID (foreign key)

**❖ Departments Table:**

➢ Department\_ID (primary key)

➢ Department\_Name

**❖ Term Table**

➢ Term\_ID (primary key)

➢ Term\_name

➢ Term\_starttime

➢ Term\_endtime

**❖ Courses Table:**

➢ Course\_ID (primary key)

➢ Course\_name

➢ Course\_credits

➢ Prerequisites\_course\_ID

➢ Suggested\_course\_ID

➢ Department\_ID (foreign key)

**❖ Student\_Courses Table:**

➢ Term\_ID (primary key, foreign key)

➢ Student\_ID (primary key, foreign key)

➢ Course\_ID

**❖ Term\_Courses Table:**

➢ Term\_ID (primary key, foreign key)

➢ Course\_ID (primary key, foreign key)

➢ Professor

➢ Time

➢ Room

**❖ Course roster**

**User Roles and Permissions:**

**❖ Supervision Admin:**

➢ View and manage all student registrations.

➢ Generate course schedules.

➢ Assign teachers to courses.

➢ Manage user accounts.

**❖ Departmental Admins (BA, IT, HA):**

➢ View and manage student registrations for their specific department.

➢ Generate course schedules for their department.

➢ Assign teachers to courses in their department.

**❖ Student:**

➢ View course catalog.

➢ Register for courses.

➢ View registered courses.

**System Functionality:**

* **Student Registration:**
* Students can log in and view the course catalog.
* They can select courses based on prerequisites and suggestions.
* Upon successful registration, the system updates the Student Courses table.
* **Course Scheduling:**
* Admins can create course schedules, considering factors like room availability, teacher availability, and course prerequisites and suggestions.
* **Security and Privacy:**
* Implement strong authentication and authorization mechanisms.
* Encrypt sensitive data like passwords and student information.
* Adhere to data privacy regulations.
* **User Interface:**
* Design intuitive and user-friendly interfaces for different user roles.
* Provide clear instructions and helpful error messages.

## **Task list**

**Setup server, web service, database:**

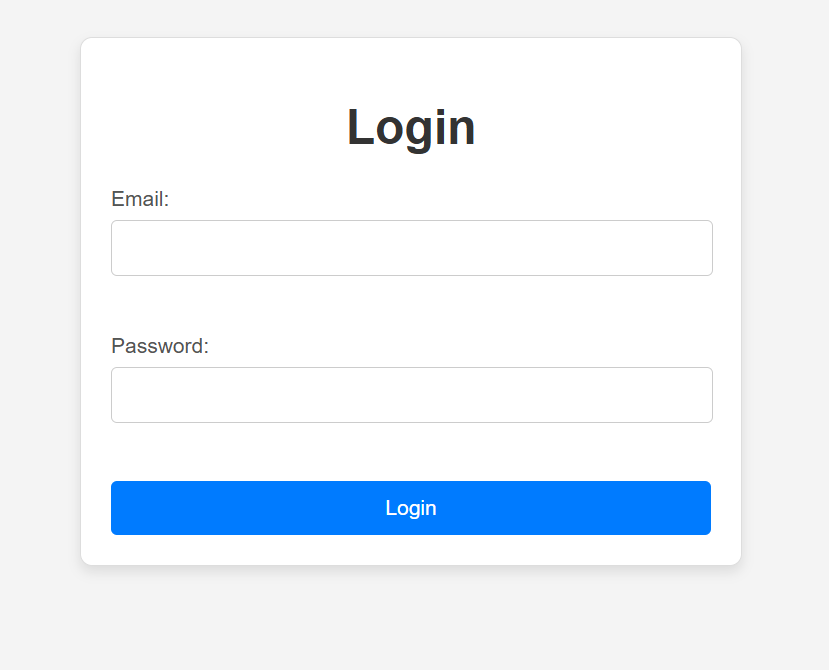
1. Setup ubuntu server on PC or Vmware.
2. Setup ssh server (sshd) for remote login.
3. Setup ftp server (ftpd base on vsftpd).
4. Setup Apache (web service).
5. Setup MySQL.
6. Samba.

**Front-end: Create Register Form Web using HTML, CSS, Javascript:**

**Front-End Development:**

* The front-end of the student website is designed to offer a user-friendly interface, ensuring an intuitive and seamless experience for users. This was achieved using HTML and CSS, which together define the structure and styling of the web pages. HTML was used to create a well-organized layout, while CSS was employed to enhance visual appeal and maintain consistency across different screens and devices.
* To add interactivity and dynamic features, JavaScript was used to implement various functions, such as automatically filling in students' personal information upon logging into the website, dynamic course selection, and providing feedback on the information they have registered. JavaScript also enabled the seamless integration of user inputs with the back-end, ensuring a smooth and responsive user experience.

**Login Form Functionality:**

****

*Login student interface.*

* The login form serves as a secure gateway for users to access the website. It is designed to collect the email and password credentials provided to each user (student and admin) by the university. The system will authenticate the user's role. If the user is an admin, the database will be partitioned into column data 1. If the user is a student, it will be partitioned into column data 2.

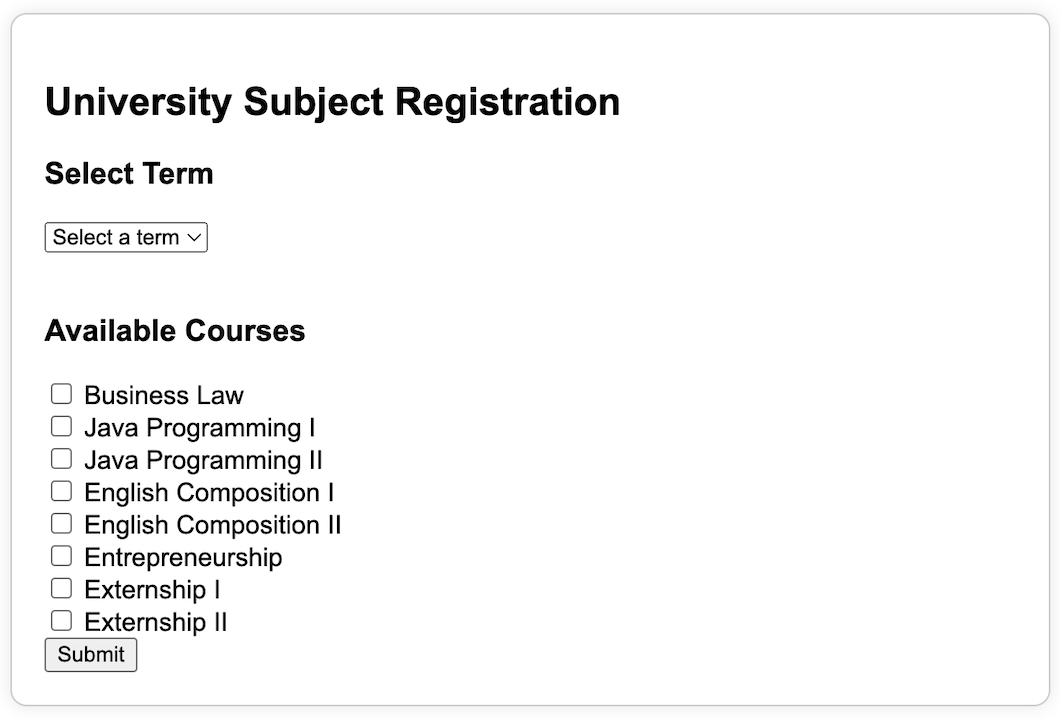
*Column 1:*

* Administrators have the authority to modify courses within a specific semester and set the activation time for the student course registration form.

*Column 2:*

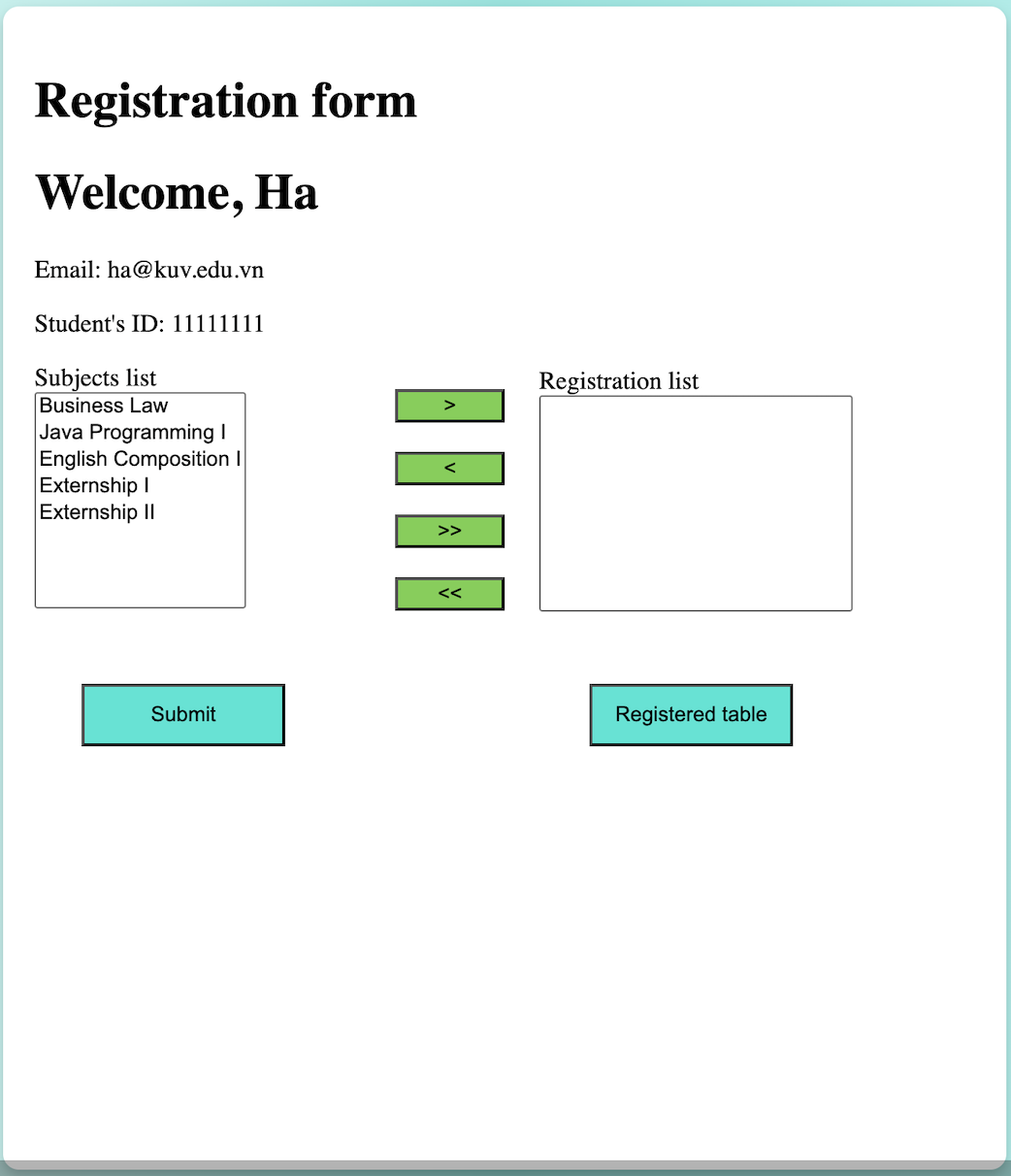
* Upon successful authentication, the login process is seamlessly linked to the student's personal information. This integration automatically retrieves and displays key details such as the student's name and student ID, ensuring a personalized and efficient user experience for each student register form.

**Admin setup courses form:**

*****Admin course registration setup form.*

* The Admin Management Form is designed to facilitate the efficient management of course registration periods by the university's registration office. This form empowers administrative users to select the term and associated courses for each period. By doing so, the registration office can define the academic offerings for each term in an organized and streamlined manner.
* One of the key features of this form is its ability to display the start and end dates for the selected term. These dates define the period during which students can access and submit their course registration forms. This functionality ensures that registration activities are conducted within the designated time frame, providing a clear structure for both administrators and students.

**Register form for student:**

*****Student register form.*

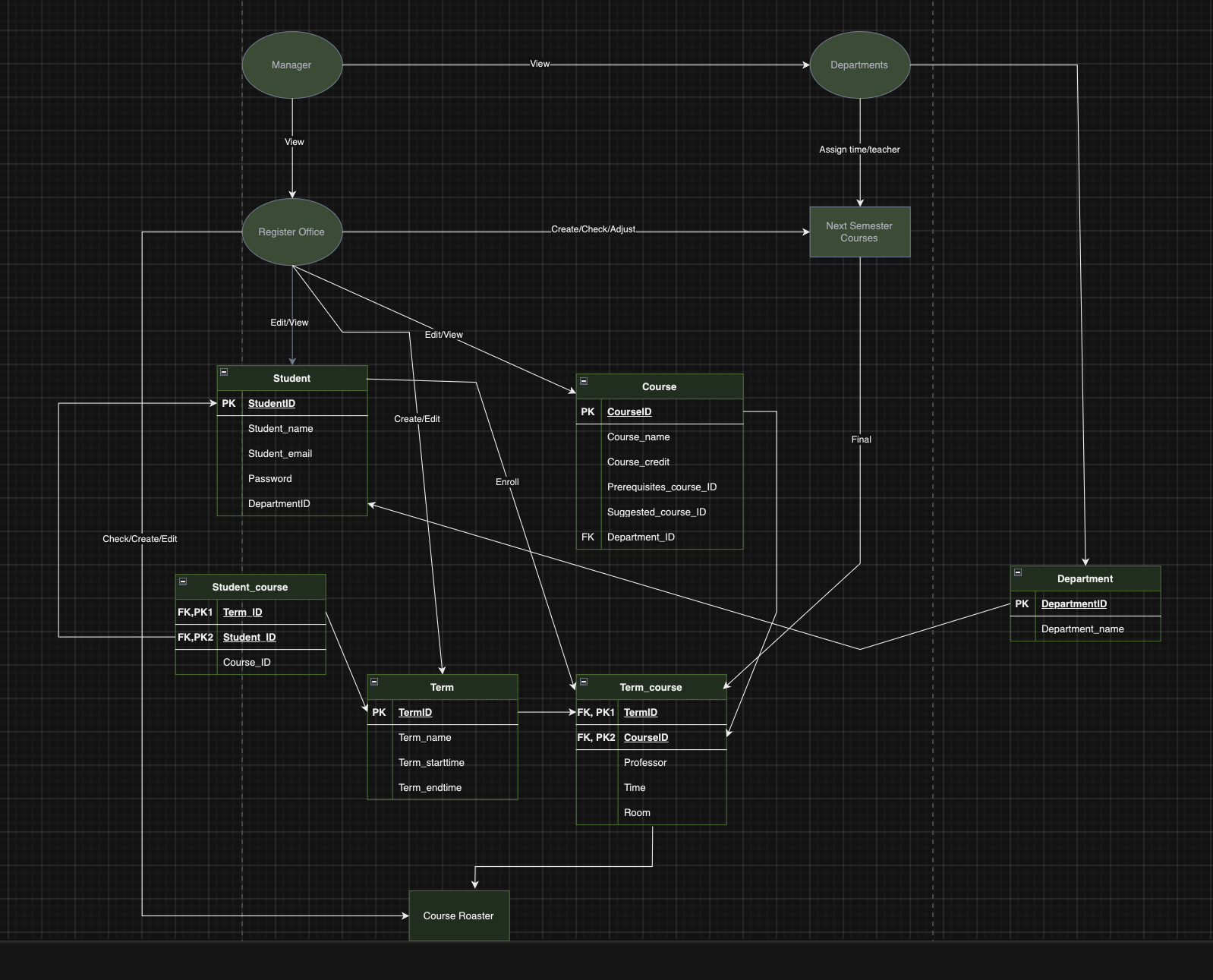
* The Student Registration Form is designed to provide students with an efficient and intuitive interface for enrolling in courses. This form includes essential student information, such as the student's name and student ID, which are pre-filled for convenience and accuracy.
* A dynamic course selection feature allows students to easily browse and select courses for the current term. The form also incorporates a registration table that displays the details of the information selected by the student. This table provides a clear overview of the registered information, enabling students to verify their choices.

**Back-end: Create Database and Fetch Data (mysql):**

**Data and Database Management:**

* The system's database is meticulously designed to uphold data consistency, integrity, and security, while facilitating efficient access and seamless updates.
* Key data entities include:
* **Student Data:** This includes information such as student names, IDs, email addresses, passwords, and linked course registrations.
* **Department Data:** This includes information about department ID, department name.
* **Course Data:** Information about courses, including course IDs, names, prerequisites and suggested, credit hours, and department ID of that course.
* **Term Data:** Details about each academic term, including term names, start and end dates, and associated courses.
* The database is relational, with clear relationships between tables for students, courses, and terms. This structure ensures that changes to one entity are reflected dynamically across the system.

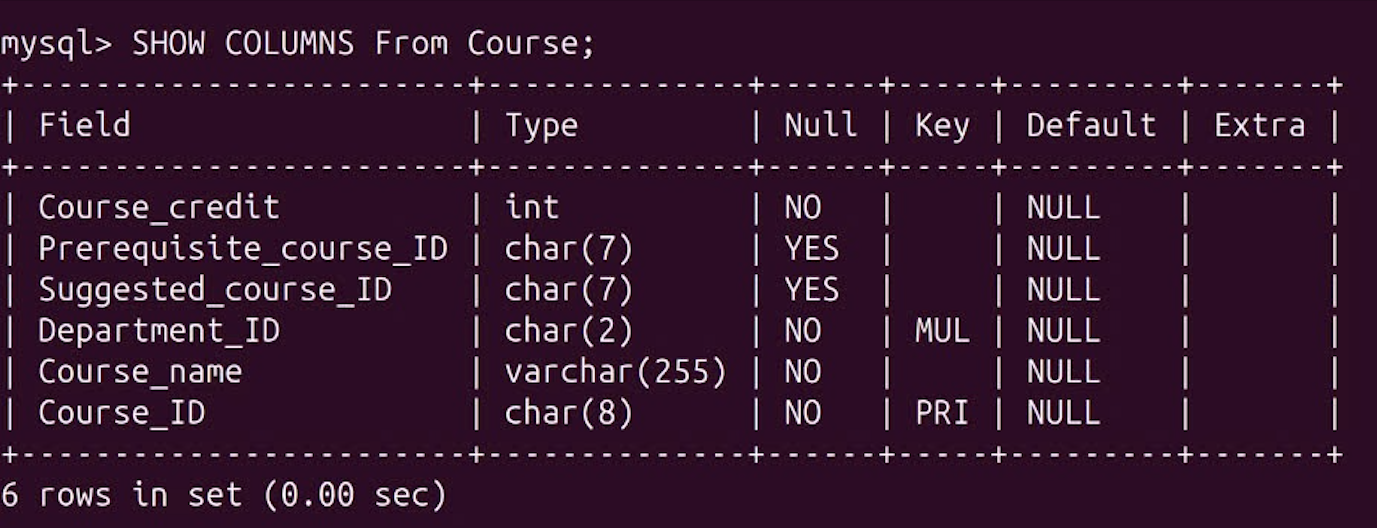
**Entity-Relationship (ER) Diagram:**

****

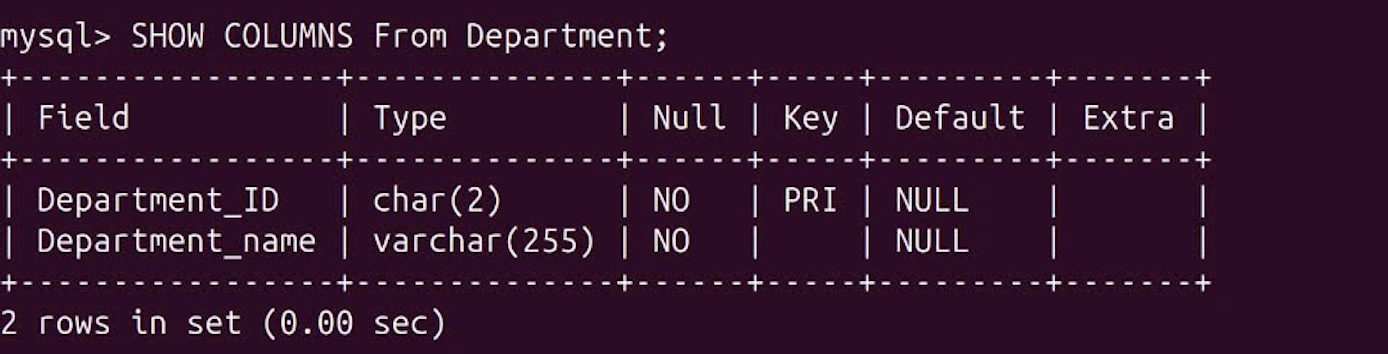
*Entity Relationship Diagram.*

* The Entity-Relationship (ER) Diagram illustrates a visual representation of this system's database structure, illustrating the relationships between various tables and entities.
* Key features of the ER diagram include:
* The diagram highlights the relationships between core entities, such as students, courses, terms, and administrative tasks, showing how data flows and interacts across the system.
* It visually represents the roles and responsibilities of different users, such as students and admin staff, and their interactions with the system's data.
* The diagram clearly identifies the primary keys for each table, which uniquely define records, and the foreign key, which establishes relationships between different entities. This structure ensures data integrity and supports relational operations.

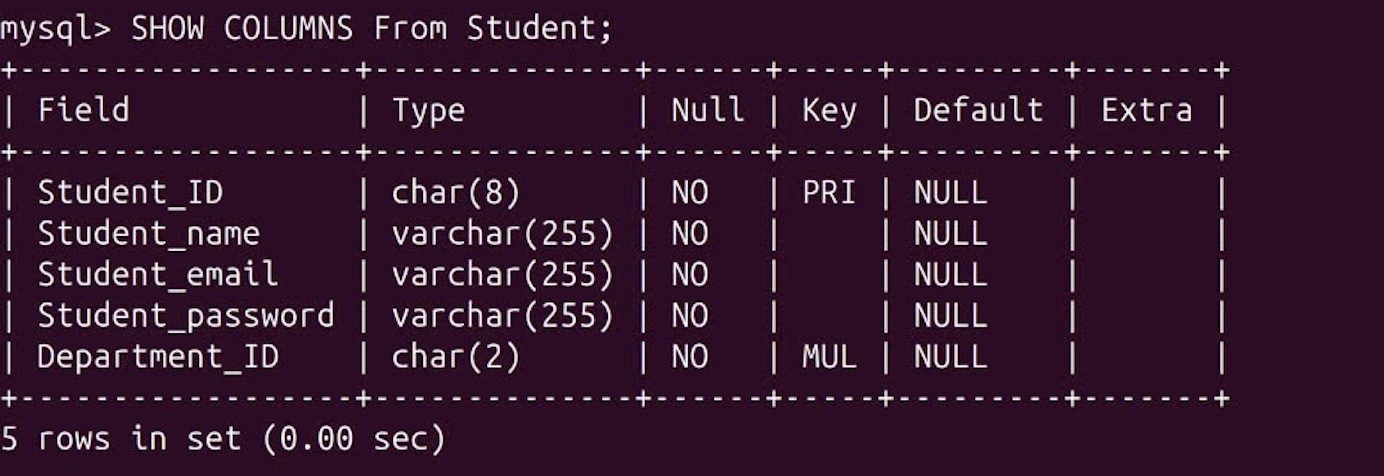
**Database Implementation with MySQL on Ubuntu:**

**

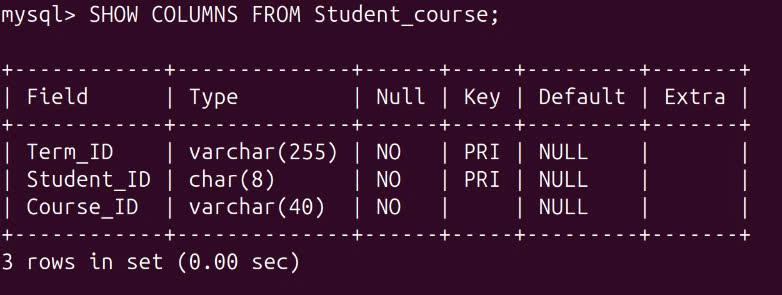
*Course table.*



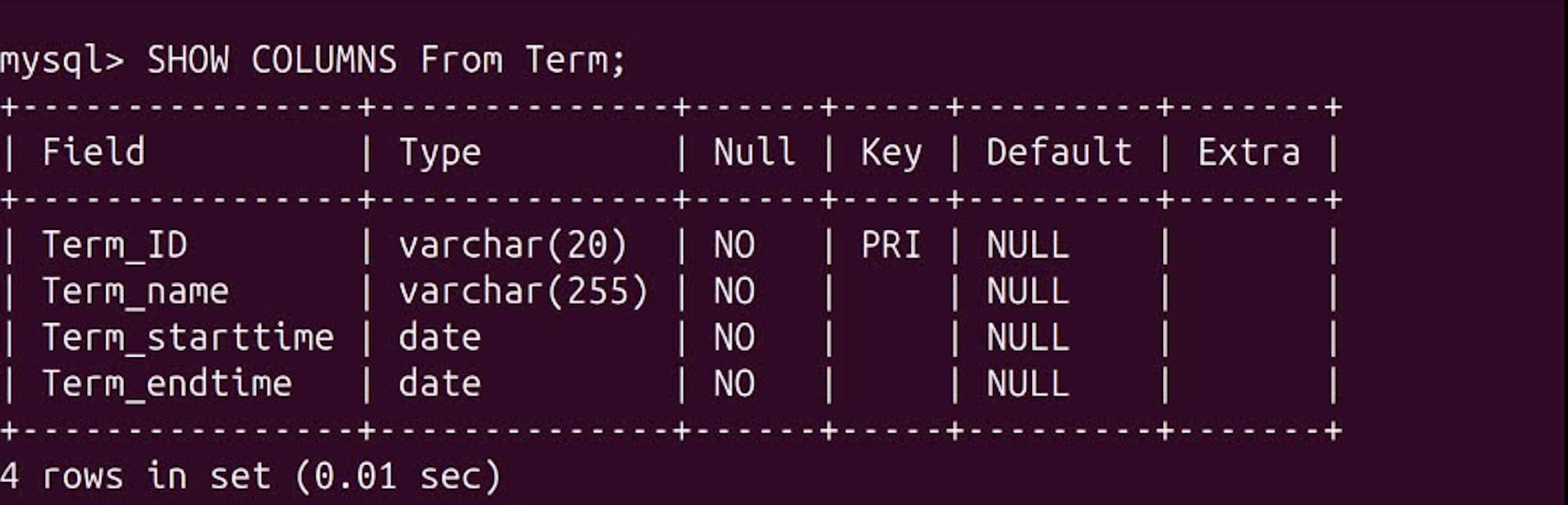
*Department table.*

**

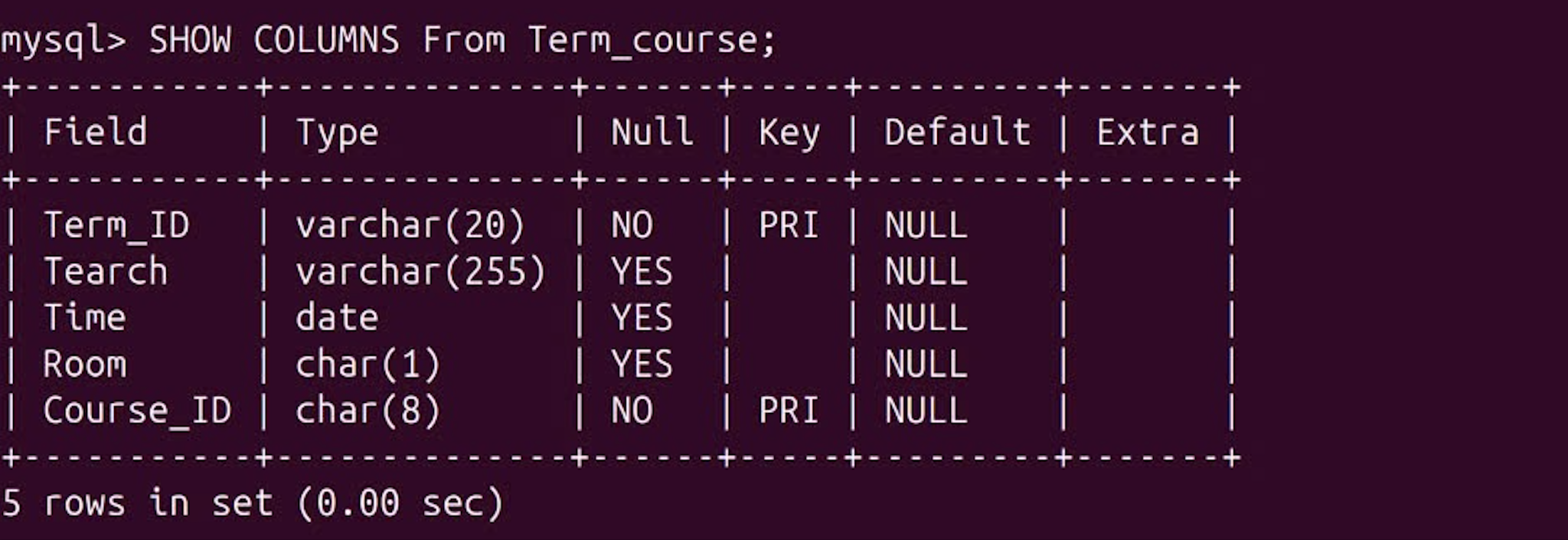
*Student table.*



*Student\_course table.*



*Term table.*



*Term\_course table.*

* The database for this system was developed using **MySQL** and deployed on an **Ubuntu** server.
* Key aspects of the database implementation include:

1. **Environment Setup**: MySQL was installed and configured on an Ubuntu server, leveraging the stability and performance of the Linux operating system. Proper configurations were applied to optimize database performance and security.
2. **Schema Design:** The database schema was carefully designed based on the requirements outlined in the ER diagram. Tables for students, courses, terms, and user roles were created with appropriate primary and foreign keys to ensure data integrity and efficient query execution.
3. **Data Management:** MySQL's features, such as indexing, were utilized to enhance query performance, especially for frequently accessed data like student information and course details.
4. **Security:** User authentication and access control mechanisms were implemented to restrict database access to authorized users only. Regular backups and updates were also scheduled to maintain data safety and system reliability.
5. **Integration:** The MySQL database seamlessly integrates with the front-end and back-end components of the system, enabling dynamic data retrieval and updates in real time.

# **Conclusion**

**Streamlined Course Registration and Management System:**

* The goal of this project is to create an intuitive course registration website using the Ubuntu platform. It streamlines course enrollment for students with secure login, dynamic course catalogs, and intuitive registration forms while providing administrators with robust tools to manage courses, schedules, and student records.
* The system guarantees a safe, effective, and expandable platform in the future by utilising front-end technologies like HTML, CSS, and JavaScript, as well as MySQL for database administration. Effectively improving academic management and user experience, this project streamlines procedures for administrators and students.

**Test case for registration form:**

| Test Case ID | Test Case Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
| --- | --- | --- | --- | --- | --- | --- |
| TU01 | Check user login when valid email ID and password are entered | 1. Open the login page.  2. Enter valid email and password. 3. Click submit. | Email: abc@gmail.com Password:  abc@123 | Users should be successfully logged in and redirected to the registration page. | Like expected result | Completed |
| TU02 | Check user login with invalid email ID or password. | 1. Open the login page.  2. Enter invalid email and password. 3. Click submit. | Email: xyz@gmail.com Password:  xyz@123 | Login attempts should fail, and an error message like "Invalid email or password" should be displayed. | Like expected result | Completed |
| TU03 | Verify registration form submission with all fields filled. | 1. Login with valid credentials. 2. Fill out all required fields. 3. Select subjects. 4. Submit the form. | Student ID: 123456 Name: James Doe Date of Birth:  2002-05-10 Gender: Male Selected Subjects: Introduction to Programming, Web Authoring | The form should be submitted successfully, and the data should appear in the table or modal as expected. | Like expected result | Completed |
| TU04 | Verify form submission with missing required fields. | 1. Login with valid credentials. 2. Leave one or more required fields empty. 3. Attempt to submit the form. | Student ID:(empty) Name:(empty) Date of Birth:  (empty)  Gender:(empty) | Submission should fail, and a validation message indicating the required fields should be displayed. | Like expected result |  |
| TU05 | Check addition of subjects to the registration list. | 1. Login. 2. Select one subject from the list on the left. 3. Click the > button. 4. Check if the subject appears in the registration list. | Subject selected: Introduction to Programming | The subject should be successfully added to the registration list. | Like expected result | Completed |
| TU06 | Check removal of subjects from the registration list. | 1. Login. 2. Add a subject to the registration list. 3. Select the subject in the registration list. 4. Click the < button to remove it. | Subject added: Web Authoring | The subject should be successfully removed from the registration list. | Like expected result | Completed |
| TU07 | Verify the registration list can handle multiple selections. | 1. Login. 2. Select multiple subjects from the list on the left. 3. Click the >> button. 4. Check the registration list for all subjects. | Subjects selected: Client-Side Scripting, Programming I, Project Management | All selected subjects should be added to the registration list. | Like expected result | Completed |
| TU08 | Check the modal displays accurate information after submission. | 1. Fill out the registration form. 2. Click the "Registered Table" button. 3. Verify the modal information. | Student ID: 456789 Name: Jane Smith Date of Birth: 2000-03-15 Gender: Female Selected Subjects: Programming I, Network+ | The modal should display the student information and a list of registered subjects accurately. | Like expected result | Completed |

**Tasks of members:**

| **Task list** | **Students** |
| --- | --- |
| Setup server, web service, database. | Hanh, My, Ha, Huyen, Vinh |
| Front-end: Create Register Form Web using HTML, CSS, Javascript. | Hanh, My, Ha |
| Back-end: Create Database and Fetch data. | Huyen, Vinh |
| Write a report. | Hanh, My, Ha |