**Course Registration System — Technical Description**

This project implements a **web-based course registration system** built with PHP and MySQL to facilitate efficient management of student course enrollments within an academic institution. The system follows a **client-server architecture** with a MySQL relational database backend and PHP-driven server-side logic.

**System Components and Features:**

* **User Authentication & Session Management:**  
  Students log in using their unique studentID and password, validated against stored credentials in the students table. PHP sessions handle user state and authorization, restricting access to registration features to authenticated users only.
* **Database Structure and Design:**  
  The relational database course\_registration\_db includes the following key tables:
  1. **students**
     + **studentID** (VARCHAR(20), Primary Key): Unique identifier for each student.
     + **studentName** (VARCHAR(100)): Full name of the student.
     + **studentPassword** (VARCHAR(100)): Password for authentication (currently stored in plain text; ideally hashed for security).
  2. **courselist**
     + **courseCode** (VARCHAR(10), Primary Key): Unique code representing each course (e.g., "SE101").
     + **courseName** (VARCHAR(100)): Descriptive course title.
     + **courseCredit** (INT): Number of credit hours assigned to the course.
  3. **registeredcourse**
     + **studentID** (VARCHAR(20), Foreign Key referencing students.studentID): Identifier of the student who registered the course.
     + **courseCode** (VARCHAR(10), Foreign Key referencing courselist.courseCode): Identifier of the registered course.
     + **Composite Primary Key:** The combination of studentID and courseCode enforces uniqueness, ensuring a student cannot register the same course multiple times.

This design implements a **many-to-many relationship** between students and courses via the registeredcourse junction table, maintaining data integrity through foreign key constraints and primary keys.

* **Course Registration Workflow:**  
  Upon successful login, the system queries the courselist table for courses not yet registered by the student by using a subquery excluding course codes present in registeredcourse. Adding a course executes an INSERT IGNORE operation into registeredcourse to prevent duplicates, while removal deletes the corresponding record.
* **Security and Data Validation:**  
  Prepared statements (mysqli::prepare with bind\_param) guard against SQL injection attacks. Output data is sanitized with htmlspecialchars() to prevent cross-site scripting (XSS). Password storage currently lacks hashing and should be improved to follow security best practices.
* **User Interface and Usability:**  
  The front-end offers responsive tables listing available and registered courses. A client-side JavaScript search function enables real-time filtering by course name or code for ease of navigation without additional server load.
* **Session and Logout Management:**  
  Logout functionality securely destroys the PHP session and redirects users to the login page, mitigating unauthorized session reuse.

This system showcases practical application of web development principles, relational database design, and secure data handling to support academic course enrollment processes.