**:**

**🔧 Technology Stack Justification**

**1. HTML (HyperText Markup Language)**

* **Purpose: Structure and layout of web pages.**
* **Justification: Acts as the skeleton of the web interface, providing a semantic and responsive user interface for forms like login, registration, and student dashboards.**

**2. JavaScript (JS)**

* **Purpose: Frontend interactivity and dynamic behavior.**
* **Justification: Handles form submissions, API calls to the backend, and updates the UI based on JSON responses from Node.js server.**

**3. Node.js**

* **Purpose: JavaScript runtime for server-side development.**
* **Justification: Provides a fast, scalable backend using asynchronous operations. Ideal for building RESTful APIs that interact with the MySQL database.**

**4. Express.js**

* **Purpose: Web application framework for Node.js.**
* **Justification: Simplifies backend routing, request handling, and middleware integration. Makes building APIs cleaner and more modular.**

**5. MySQL**

* **Purpose: Relational Database Management System (RDBMS).**
* **Justification: Provides strong data consistency, foreign key relationships, and efficient query handling for structured data like students, courses, marks, and grades.**
* **And in our system it is based on user so it requires security for accessing the data.**

**📊 ER Diagram Explanation**

**The Entity-Relationship Diagram (ERD) visually represents the database structure and relationships between entities:**

**Entities & Attributes:**

* **User: General table for all users (admin, student, instructor) with common attributes like name, email, and role.**
* **Student & Instructor: Specialized tables linked to User via foreign keys to distinguish roles and hold extra details (e.g., semester, department).**
* **Admin: Generates results for student without modifying the marks.**
* **Course: Managed by instructors and identified by course\_id.**
* **Enrollment: Many-to-many link between students and courses, holds registration details.**
* **Marks: Stores assessment scores per enrollment.**
* **Grades: Stores letter grade and grade point per enrollment.**

**Key Relationships:**

* **One User can be either a Student or an Instructor (1:1).**
* **One Instructor teaches many Courses (1:M).**
* **One Student can enroll in many Courses via Enrollment (M:N through junction table).**
* **Each Enrollment has one set of Marks and Grades (1:1).**