Building a Comprehensive Student Management System

This presentation outlines the core modules and functionalities required to develop a robust and efficient student management system.

Core Modules Overview

Course Management

Create, update, delete, and view course details.

Student Enrollment

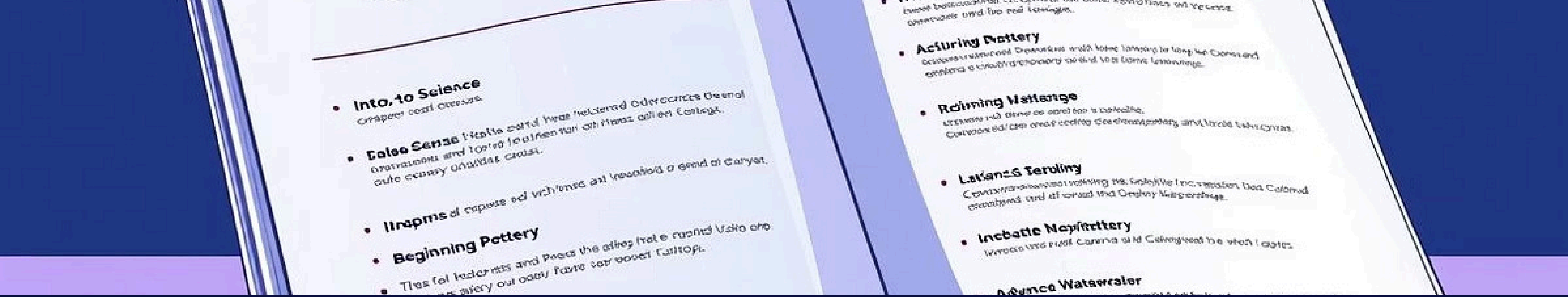
Manage student enrollment in multiple courses.

Marks & Results

Store and manage student marks and generate reports.

Attendance Tracking

Monitor student attendance per course and date.



Course Management Module

Key Functions

- Create, update, delete, and view courses.
- Each course includes ID, Name, Duration, and Fees.

Database Schema

```
CREATE TABLE courses (
 id INT PRIMARY KEY AUTO_INCREMENT,
 name VARCHAR(100),
 duration VARCHAR(50),
 fees DOUBLE
);
```

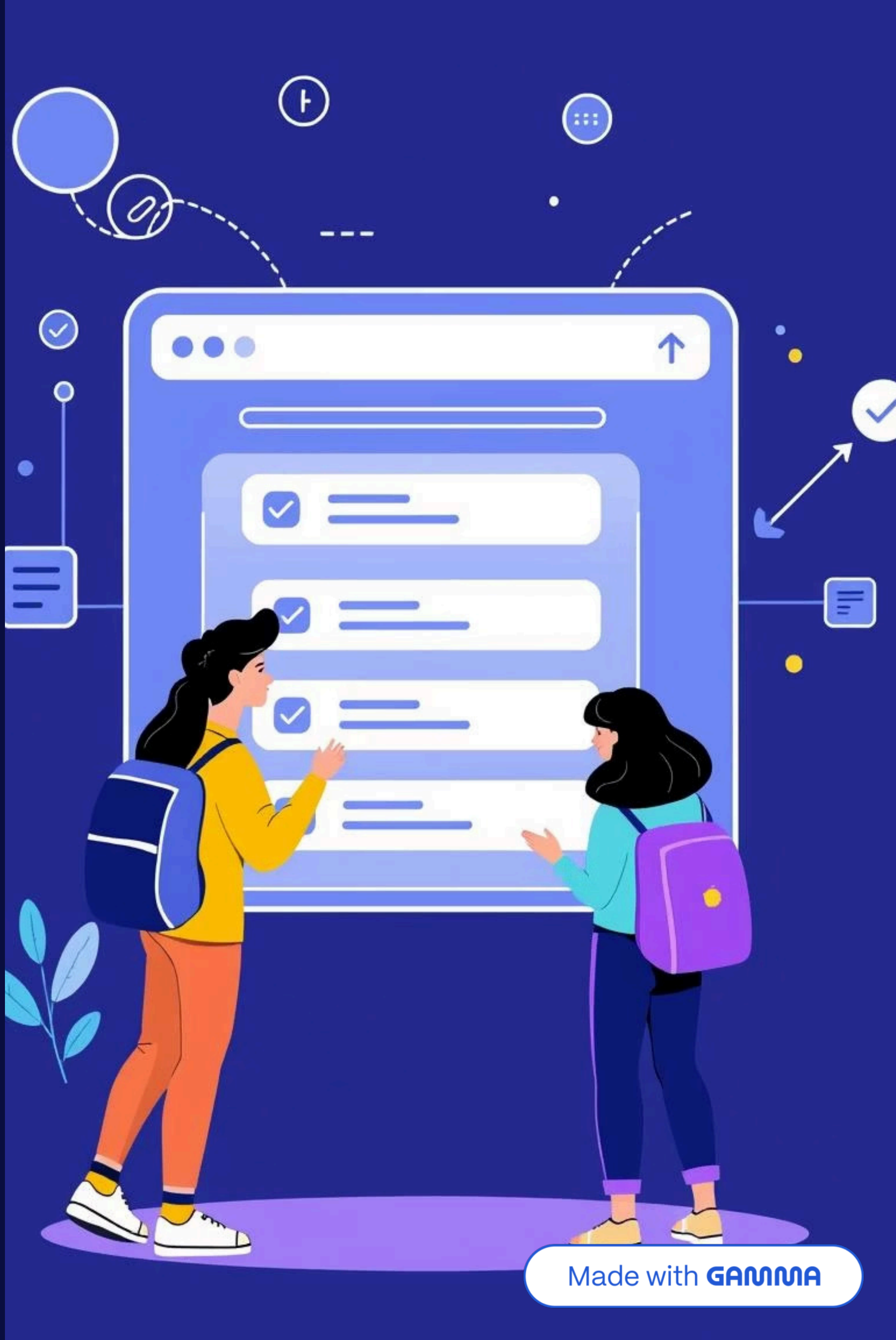
Student-Course Enrollment Module

Relationship & Operations

- Many-to-Many relationship between students and courses.
- Students can enroll in multiple courses.
- Operations: Enroll, View enrolled courses, Remove enrollment.

Join Table Schema

```
CREATE TABLE enrollments (
 student_id INT,
 course_id INT,
 PRIMARY KEY (student_id,
course_id),
 FOREIGN KEY (student_id)
REFERENCES students(id),
 FOREIGN KEY (course_id)
REFERENCES courses(id)
);
```



Marks / Result Module

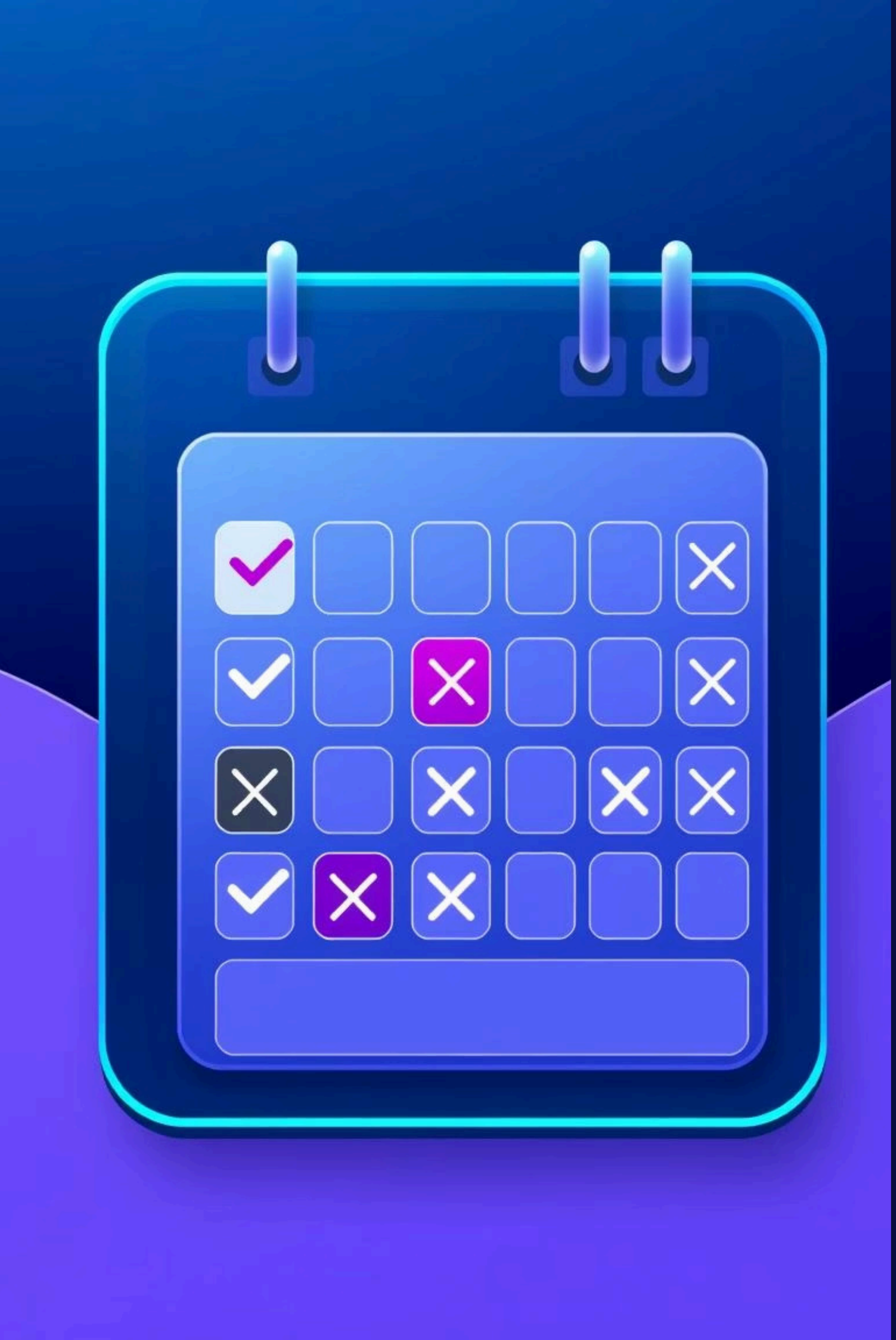
Functionality

- Store and manage marks per student per course.
- Add/update marks.
- Generate result reports (total, average, grade).

Database Schema

```
CREATE TABLE results (
 student_id INT,
 course_id INT,
 marks INT,
 PRIMARY KEY (student_id,
course_id),
 FOREIGN KEY (student_id)
REFERENCES students(id),
 FOREIGN KEY (course_id)
REFERENCES courses(id)
);
```

Made with GAMMA



Attendance Module

Tracking & Reporting

- Track attendance per student per course per date.
- Mark attendance (Present/Absent).
- View attendance summary and generate monthly reports.

Database Schema

```
CREATE TABLE attendance (
 id INT PRIMARY KEY
AUTO_INCREMENT,
 student_id INT,
 course_id INT,
 date DATE,
 status VARCHAR(10), --
Present/Absent
 FOREIGN KEY (student_id)
REFERENCES students(id),
 FOREIGN KEY (course_id)
REFERENCES courses(id)
);
```

User Login Module

Secure Access

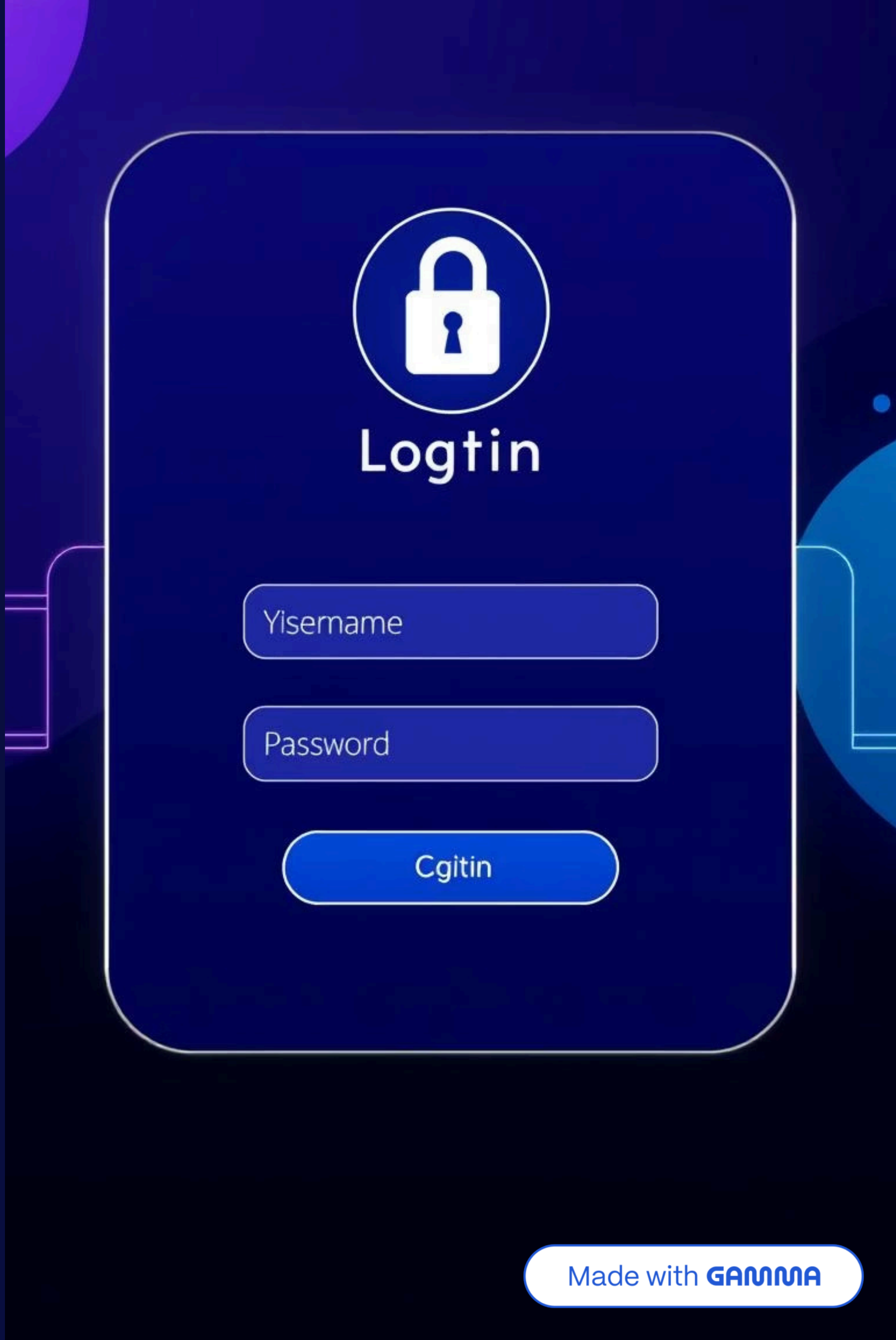
Add basic user login system for admin/teacher roles.

Authentication

Secure login using hashed passwords.

Role-Based Access

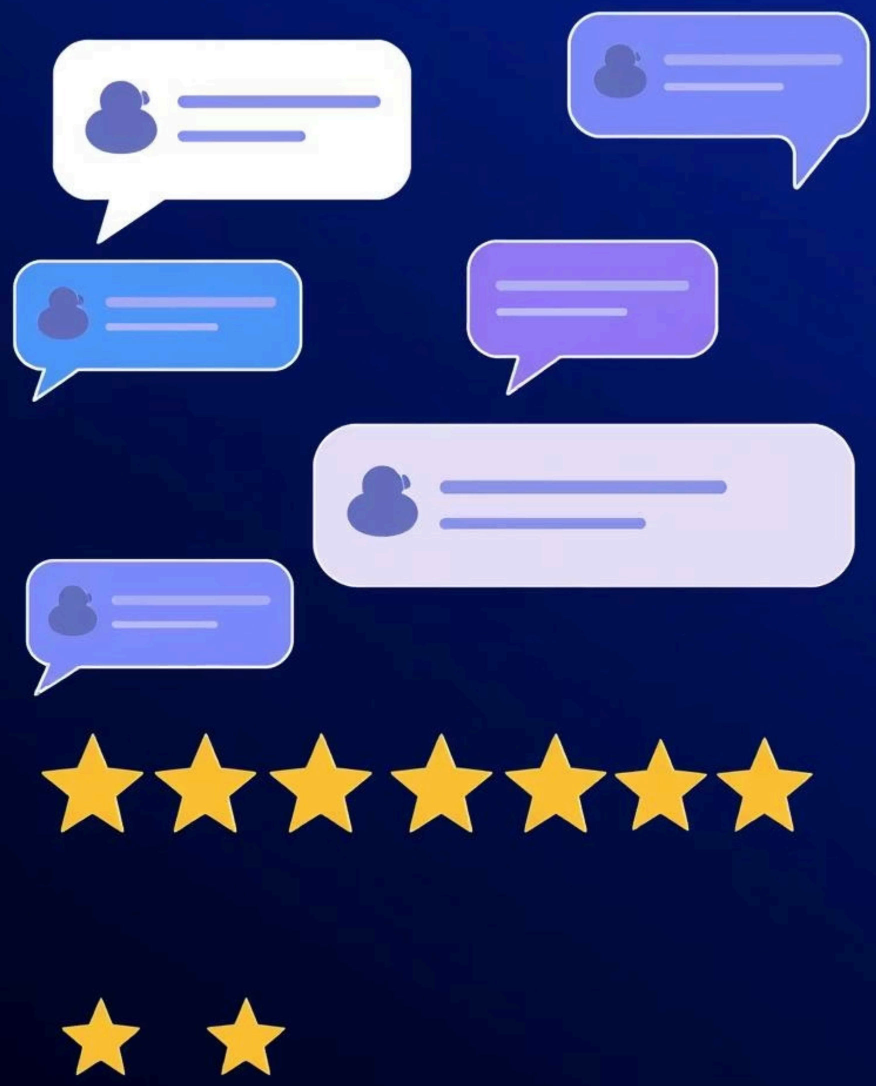
Optional role-based access control for different user types.



User Login Module: Schema

The **users** table stores user credentials and roles, ensuring secure and authenticated access to the system.

```
CREATE TABLE users (
 id INT PRIMARY KEY AUTO_INCREMENT,
 username VARCHAR(50) UNIQUE,
 password VARCHAR(255),
 role VARCHAR(20) -- admin, teacher
);
```



Feedback Module

Student Feedback

- Allows students to submit feedback for courses or teachers.
- Captures feedback text and submission date.

Database Schema

```
CREATE TABLE feedback (
 id INT PRIMARY KEY
AUTO_INCREMENT,
 student_id INT,
 course_id INT,
 feedback_text TEXT,
 submitted_on DATE,
 FOREIGN KEY (student_id)
REFERENCES students(id),
 FOREIGN KEY (course_id)
REFERENCES courses(id)
);
```

Report Generation Module

Performance Reports

Generate detailed student performance reports.

Course-wise Results

Summarize results for individual courses.

Attendance Summaries

Create comprehensive attendance summaries.

Top Performers

Identify and highlight top-performing students.