##### Project Proposal: "Student Management System

This project will be a beginner-friendly yet practical application designed to help you understand

and implement the concepts of forms, CRUD operations, and template management effectively.

**Objective**

Create a simple web-based system to manage students' records, including adding, updating,

viewing, and deleting information, while utilizing Django features like HTML forms, model forms,

class-based views, and template inheritance.

**Key Features**

**1. Student Registration**

Use HTML Forms to collect basic details such as:

Name, Email, Phone Number, Course.

Apply HTML Built-in Validation (e.g., required fields, proper email format).

**2. Student List Page**

Display all registered students using a template.

Create reusable templates for listing CRUD data.

**3. CRUD Operations**

Create: Add a new student using Model Forms.

Read: Display student details on a detailed view page.

Update: Edit student details using GET & POST requests.

Delete: Remove a student using a DELETE request.

**4. Template Features**

Implement Template Inheritance to structure pages (base.html for navbar, footer, etc.).

Use Template Blocks for customizing sections like forms, lists, and messages.

**5. Message Framework**

Show success/error messages (exmp., "Student added successfully", "Record deleted").

Tools & Technologies

Frontend: HTML, CSS (optional for styling).

Backend: Django (Focus on forms, views, templates).

Database: SQLite (default Django database).

**Learning Outcomes**

Understand HTML forms and their validation.

Learn how to use Django Model Forms for database integration.

Master CRUD operations with GET, POST, UPDATE, DELETE requests.

Create dynamic, reusable templates for forms and CRUD pages.

Submission Guideline:

Submit the GitHub Repository Link of this project. Otherwise, you will get 0 (ZERO)