ABSTRACT:

This project aims to modernize student information management in educational institutions by developing a sophisticated web application using the CodeIgniter framework. The focus is on creating an efficient system with seamless handling of student details through a user-friendly interface. By implementing CRUD operations, the project seeks to establish a secure, scalable, and intuitive platform for storing and managing student records.

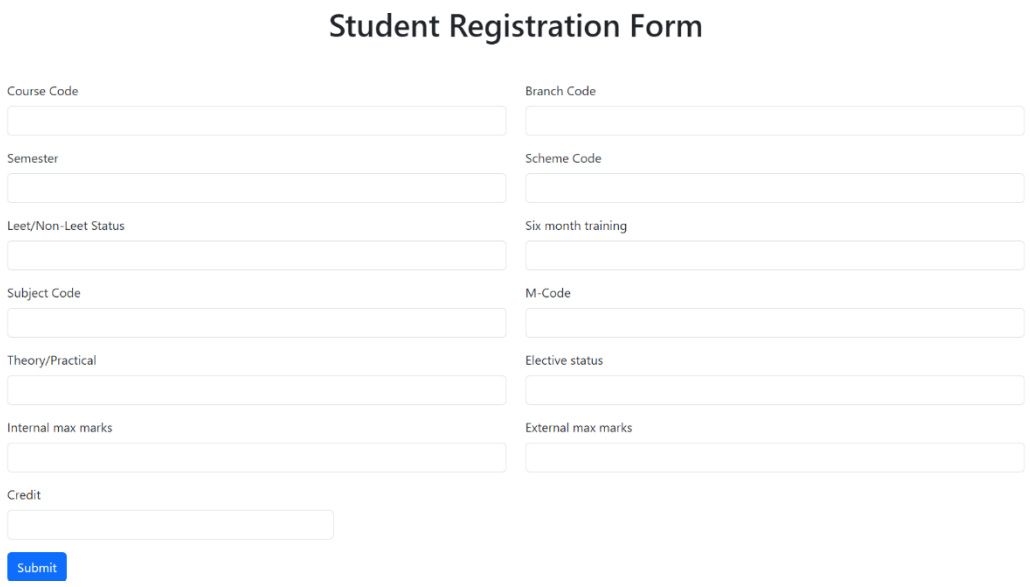
In the realm of educational information systems, the project leverages the capabilities of the CodeIgniter framework to build an innovative solution tailored to the dynamic needs of educational institutions. The primary goal is to construct a comprehensive platform adept at efficiently managing student details while ensuring data integrity, security, and user-friendliness.

Given the pivotal role of efficient student information management in today's educational landscape, this project addresses challenges posed by manual record-keeping, disparate spreadsheets, and fragmented databases. The aim is to engineer a revolutionary solution that streamlines student details management, enhancing administrative efficiency and adapting to evolving educational requirements.

INTRODUCTION:

The efficient management and organization of student information serve as foundational pillars in facilitating seamless administrative operations and fostering academic excellence. However, the prevalent reliance on archaic, paper-based record-keeping methods and disjointed digital spreadsheets presents formidable challenges. These challenges encompass data redundancy, inconsistent data integrity, limited accessibility, and an inability to swiftly adapt to the dynamic and evolving needs of educational institutions. Recognizing these limitations, this project embarks on a transformative journey towards architecting an innovative solution poised to revolutionize the management of student details within educational ecosystems

Form



Objectives:

- Develop a comprehensive database schema for efficient storage of student information.
- Implement CRUD functionality to facilitate the seamless management of student records.
- Design an intuitive and user-friendly interface tailored for administrators to interact with and navigate student data effectively.
- Ensure robust data security measures and implement validation protocols to enhance the accuracy and reliability of user inputs.

Implementation Steps:

Database Design:

- Established a MySQL database incorporating tables to meticulously store student details, including but not limited to ID, name, contact information, and enrolled courses.
- Ensured normalization and delineated relationships among tables to optimize data retrieval efficiency.

CodeIgniter Setup:

- Installed and configured CodeIgniter on the server to lay the foundation for the web application.
- Set up essential configuration files, with a specific focus on the database configuration to establish seamless connectivity.

Model-View-Controller (MVC) Structure:

- Implemented the MVC architecture to ensure an organized and modular development approach.
- Delegated database interactions to models, entrusted user interface representation to views, and orchestrated application flow through controllers.

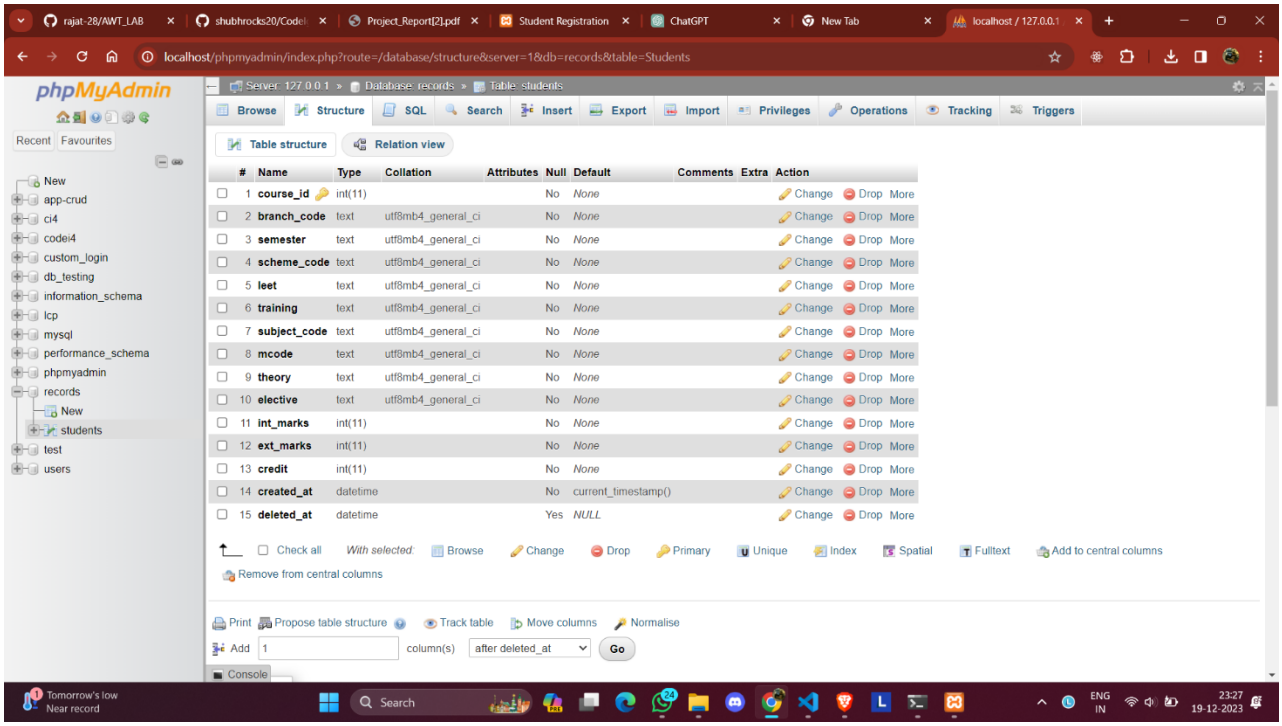
CRUD Operations:

- Developed functions/methods within controllers and models to enable Create, Read, Update, and Delete operations on student records.
- Crafted user-friendly forms and interfaces to empower administrators with the ability to effortlessly add, view, edit, and delete student records.

User Interface Design:

- Designed an aesthetically pleasing and user-friendly interface using HTML and CSS, enhancing the overall user experience.
- Implemented robust validation mechanisms for user inputs to safeguard data integrity and enhance the reliability of the system.

Database Structure



Data Visualisation

localhost/CURRENT_PROJECT/Home/show

Student URN	Course ID	Branch Code	Semester	Scheme Code	Leet	Training	Subject Code	M Code	Theory	Elective	Int Max Marks	Ext Max Marks	Credit	Created at	Action
1	0	M-45	22	eee	eee	cdd	ddf	fdd	fdfg	ddfg	21	23	12	2023-12-19 23:31:16	Delete Edit
2	0	IT104	5	2023	Non Leet	Yes	PCIT-110	11	theory	True	90	1000	12	2023-12-19 23:37:07	Delete Edit
12	22	2232	2	2232	leet	yes	pcit-1220	1002	theory	yes	20	58	21	2023-12-19 23:40:01	Delete Edit
13	0	branch-1	1	scheme-1	leet-1	training-1	subject-1	mcode-1	theory-1	elective-1	75	80	3	2023-12-19 23:42:18	Delete Edit
14	0	branch-2	2	scheme-2	leet-2	training-2	subject-2	mcode-2	theory-2	elective-2	80	85	4	2023-12-19 23:42:18	Delete Edit
15	0	branch-1	1	scheme-1	leet-1	training-1	subject-3	mcode-3	theory-3	elective-3	70	75	3	2023-12-19 23:42:18	Delete Edit
16	0	branch-2	2	scheme-2	leet-2	training-2	subject-4	mcode-4	theory-4	elective-4	85	90	4	2023-12-19 23:42:18	Delete Edit
17	0	branch-1	1	scheme-1	leet-1	training-1	subject-5	mcode-5	theory-5	elective-5	78	82	3	2023-12-19 23:42:18	Delete Edit
18	0	branch-2	2	scheme-2	leet-2	training-2	subject-6	mcode-6	theory-6	elective-6	88	92	4	2023-12-19 23:42:18	Delete Edit

Soft Delete in CodeIgniter:

Database Schema:

- Add a nullable **deleted_at** column to track soft deletions.

Model Setup:

- Set **$soft_delete = true;** in the CodeIgniter model.

Soft Delete Process:

- Use **softDelete** instead of **delete** to mark records with the current timestamp.

Query Scope:

- Exclude soft-deleted records by default in queries.

Benefits:

- Retains historical data for auditing.
- Provides recovery options for accidental deletions.

rn	course_id	branch_code	semester	scheme_code	leet	training	subject_code	mcode	theory	elective	int_marks	ext_marks	credit	created_at	deleted_at
1	0	M-45	22	eee	eee	cdd	ddf	fdd	fdfg	ddfg	21	23	12	2023-12-19 23:31:16	NULL
2	0	IT104	5	2023	Non Leet	Yes	PCIT-110	11	theory	True	90	1000	12	2023-12-19 23:37:07	NULL
12	22	2232	2	2232	leet	yes	pcit-1220	1002	theory	yes	20	58	21	2023-12-19 23:40:01	NULL
13	0	branch-1	1	scheme-1	leet-1	training-1	subject-1	mcode-1	theory-1	elective-1	75	80	3	2023-12-19 23:42:18	NULL
14	0	branch-2	2	scheme-2	leet-2	training-2	subject-2	mcode-2	theory-2	elective-2	80	85	4	2023-12-19 23:42:18	NULL
15	0	branch-1	1	scheme-1	leet-1	training-1	subject-3	mcode-3	theory-3	elective-3	70	75	3	2023-12-19 23:42:18	NULL
16	0	branch-2	2	scheme-2	leet-2	training-2	subject-4	mcode-4	theory-4	elective-4	85	90	4	2023-12-19 23:42:18	NULL
17	0	branch-1	1	scheme-1	leet-1	training-1	subject-5	mcode-5	theory-5	elective-5	78	82	3	2023-12-19 23:42:18	NULL
18	0	branch-2	2	scheme-2	leet-2	training-2	subject-6	mcode-6	theory-6	elective-6	88	92	4	2023-12-19 23:42:18	NULL
19	0	branch-1	1	scheme-1	leet-1	training-1	subject-7	mcode-7	theory-7	elective-7	72	78	3	2023-12-19 23:42:18	2023-12-19 18:13:27

Results and Discussion:

The objectives of the project have been met, resulting in several notable achievements:

- **Functional CRUD Operations:**
  - The web interface enables administrators to seamlessly create, read, update, and delete student records.
- **Efficient Database Management:**
  - The implemented database schema ensures the efficient storage and management of student details, promoting data integrity and scalability.
- **Intuitive User Interface:**
  - Users experience a clean and responsive interface, enhancing the overall interaction and usability for administrators.

Conclusion:

The implementation of the student management system using CodeIgniter showcases its effectiveness in handling student details. The user-friendly interface empowers administrators to manage student records securely. Future enhancements may include

features such as user authentication, advanced search capabilities, and the ability to generate reports.

References:

- CodeIgniter Documentation: https://codeigniter.com/user_guide/
- Bootstrap Documentation: https://getbootstrap.com/

GitHub Repository Link

🔗 Awt Minor Project(https://github.com/shubhrocks20/AWT_MINOR_PROJECT)