ITE

GALURPOT GOMEZ LAZAN MARAÑAN TALINO

CRUD OPERATION (MVC FRAMEWORK – CODEIGNITER)

Start

ITE

INTRODUCTION

CRUD stands for Create, Read, Update, Delete, which are the four essential operations used to manage data in web applications.

CodeIgniter is a lightweight PHP framework that uses the Model–View–Controller (MVC) pattern to help developers build dynamic, organized, and secure web systems (Mandaviya et al., 2017).

Timmerman Industries

WHAT IS MVC?

Model – Handles data, database queries, and business logic.

View – Displays information to users (HTML pages, forms, tables).

Controller – Receives input, processes requests, and passes data between Model and View (Mandaviya et al., 2017).

ITE

WHAT IS CRUD?

- **Create**: Add new data
- **Read**: Display or retrieve existing data
- **Update**: Modify existing data
- **Delete**: Remove data
- **CRUD operations map to SQL commands**: INSERT, SELECT, UPDATE, DELETE (CollegeSideKick, 2020).

WHY CRUD MATTERS

CRUD is the foundation of all database-driven systems, including login systems, inventory, user accounts, and school systems (Ramakrishnan & Gehrke, Database Management Systems).

It ensures consistent data management and improves organization and structure of web applications.

ITE

WHY USE CODEIGNITER FOR CRUD?

- Simple folder structure
- Fast performance (lightweight framework)
- Clear separation using MVC
- Built-in security and validation features
- Query Builder makes database operations safer
- Easy to learn for beginners

CRUD WORKFLOW IN CODEIGNITER

- Model: Contains CRUD functions and communicates with the database
- Controller: Handles page requests, calls Model functions, and loads Views
- View: Shows forms for Create & Update, tables for Read, and buttons for Delete (Chapagain, CodeIgniter MVC CRUD Application)

ITE

HOW CRUD WORKS

Create: User submits a form → Controller → Model → saves to database

Read: Controller asks Model for data → View displays the list

Update: User edits a record → Controller → Model updates database

Delete: User clicks delete → Controller → Model removes record

ITE

ADVANTAGES OF MVC + CRUD

- Cleaner and more organized code
- Easier teamwork (Model, View, Controller handled separately)
- Faster development
- Easier troubleshooting
- Reusable components
- Scalable for large applications (Mandaviya et al., 2017)

CHALLENGES

- Requires proper planning of folder structure
- Must validate user input
- Must protect database from injection attacks
- Requires understanding of routing and controllers
- Large projects need modular organization (CodexWorld, CodeIgniter Guide)

CONCLUSION

CRUD operations are essential in almost all systems that rely on databases because they allow data to be created, viewed, updated, and removed efficiently.

Using the MVC architecture in CodeIgniter helps organize these CRUD processes by separating the logic, presentation, and control flow, making the application easier to maintain and improve.

This approach supports better readability, stronger security, and smoother scalability for real-world web development (Mandaviya et al., 2017).