**CRUD APP PROJECT REPORT**

**NAME-ADITYA SAINI**

**OBJECTIVE**

To create a simple GUI app that will interact with the MySQL database and help to perform CRUD (Create, Read, Update, Delete) Operations.

**LOGIC OF CODE-**

This Python code creates a simple CRUD (Create, Read, Update, Delete) application using the `tkinter` library for the GUI and `mysql.connector` for database interactions. Here's a breakdown of the logic:

**1. Importing Libraries**

- tkinter: Used to create the GUI application.

- messagebox: A submodule of `tkinter` that provides pop-up dialogs to display messages.

- mysql.connector: Allows the application to connect to a MySQL database.

**2. Database Operations**

- insertData(): Inserts a new employee record into the `empDetails` table. It checks if any of the fields (`id`, `name`, or `dept`) are empty before inserting. If not, it connects to the database, executes the `INSERT` SQL command, and then commits the changes. After insertion, it clears the input fields and shows all records.

- updateData(): Updates an existing employee record in the `empDetails` table. It checks if any of the fields are empty before updating. If not, it executes the `UPDATE` SQL command to modify the employee's name and department based on their ID.

- getData(): Fetches an employee's details from the database using their `empID`. It uses the `SELECT` SQL command to retrieve the data. If the employee exists, it fills the name and department fields in the GUI with the retrieved data.

- deleteData(): Deletes an employee's record from the `empDetails` table based on their `empID`. It uses the `DELETE` SQL command to remove the record from the database.

- show(): Retrieves all records from the `empDetails` table and displays them in the `Listbox` widget in the GUI. It is called after every database operation to update the displayed list of employees.

- resetFields(): Clears all input fields in the GUI.

**3. Creating the GUI**

- The GUI window is created using `tkinter.Tk()`.

- Labels, text entry fields, and buttons are placed in the window to interact with the user.

- Labels are used to indicate where to enter the employee ID, name, and department.

- Entry widgets allow users to input the employee details.

- Buttons trigger the various CRUD operations.

- A Listbox is used to display the list of employee records.

**4. Main Loop**

- `window.mainloop()` starts the application's main event loop, waiting for user interaction.

**5. Workflow**

- Users can insert, update, fetch, or delete employee records.

- After each operation, the list of records is updated and displayed.

- The application checks for input validation (e.g., ensuring that no fields are left empty) before performing database operations.

This code provides a basic framework for managing employee data in a MySQL database through a graphical interface.