جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Lecture: (6)

Subject: Student Records Management in Web Applications

Level: Third stage

Lecturer: Msc najwan thaeer ali

Introduction:

This lecture describes the implementation and testing of the Student Record Management System. Implementation refers to the process of converting the system design into a working software application. Testing is performed to ensure the system works correctly, efficiently, and meets the requirements. The Student Record Management System has been developed to manage student information in a computerized environment. The system allows administrators to easily add, update, delete, and view student records. It reduces manual work, saves time, and improves data accuracy. This lecture explains the tools and technologies used, system architecture, and database design.

In modern web applications, managing student records efficiently is a fundamental requirement. This includes updating, deleting, validating, and securing student data while ensuring smooth user interaction.

This lecture focuses on:

1. Database operations
2. Form handling
3. Data validation
4. User interaction
5. Security considerations

1-Database Operations (CRUD)

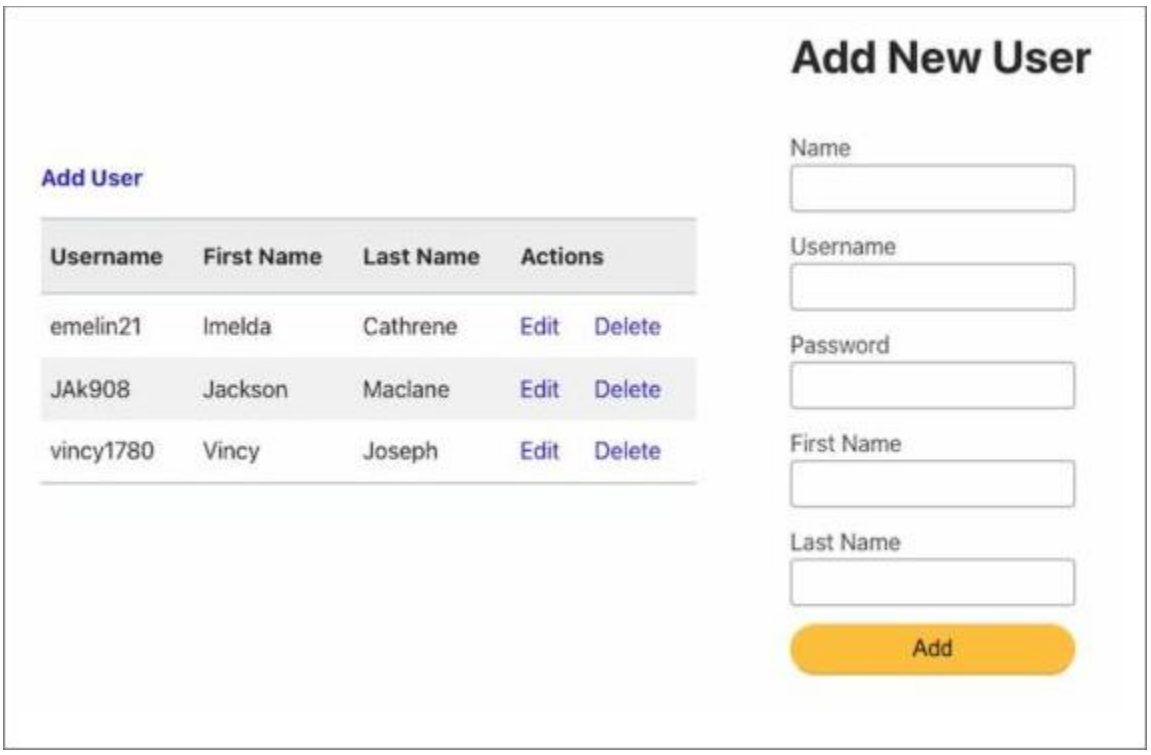
Database operations, commonly referred to as **CRUD**, represent the four fundamental actions performed on data stored in a database. These operations are essential for managing and manipulating data in any database-driven application.

- **Create:** The process of adding new records to the database.
- **Read:** Retrieving or viewing existing data from the database.
- **Update:** Modifying or editing existing records in the database.
- **Delete:** Removing records from the database.

Together, CRUD operations form the backbone of database interaction in modern web applications, enabling efficient data storage, retrieval, and management.

```sql
UPDATE students
SET name = 'Ali', age = 22
WHERE id = 1;

DELETE FROM students
WHERE id = 1;
```



https://helpdocs.quickadminpanel.com/create-panel/create-crud-string-fields

Connecting Web App to Database

Common technologies:

- PHP + MySQL
- Python (Flask / Django)
- Node.js

```
import sqlite3

conn = sqlite3.connect('students.db')
cursor = conn.cursor()

cursor.execute("SELECT * FROM students")
data = cursor.fetchall()
```

2-Form Handling (HTML + Backend)

Form handling refers to the process of collecting user input through an HTML form and processing that data on the server using a backend language such as PHP, Python, or Node.js.

It involves:

- Designing a form in HTML to capture user data (e.g., name, age, email).
- Sending the data to the server using methods like **GET** or **POST**.
- Processing, validating, and possibly storing the data in a database using backend code.

In simple terms, form handling is the bridge between the user interface (frontend) and the server logic (backend), allowing users to interact with web applications effectively.

```
<form method="POST">
  <input type="text" name="name">
  <input type="number" name="age">
  <button type="submit">Update</button>
</form>
```

3-Data Validation

Data Validation refers to the process of ensuring that the data entered by users is correct, complete, and follows the required format before it is processed or stored in a database.

It involves checking:

- Whether required fields are filled
- If the data type is correct (e.g., numbers, text)
- If the values are within acceptable ranges

- If the input format is valid (e.g., email format)
  To prevent errors, improve data quality, and protect the system from invalid or harmful input.

:Example

JavaScript

```javascript
if(name == "") {
  alert("Name is required");
}
```

:Python

Run

Python

```python
if not name:
    return "Invalid input"
```

Updating Student Records

Updating Student Records refers to the process of modifying or editing existing student information stored in a database.

This operation allows changes to be made to specific data fields such as:

- Student name
- Age
- Email
- Grades

The update is typically performed using a unique identifier (such as Student ID) to ensure that only the intended record is modified.
To keep student information accurate, current, and consistent within the system.

```
cursor.execute("UPDATE students SET name=? WHERE id=?", (name, id))
conn.commit()
```

4-User Interaction (UI/UX)

User Interaction (UI/UX) refers to how users communicate and interact with a web application through its interface, and how easy, efficient, and enjoyable that experience is.

- **UI (User Interface):** The visual elements of the system such as buttons, forms, menus, and layout.
- **UX (User Experience):** The overall experience the user has while using the system, including ease of use, clarity, and satisfaction.

Purpose:
To ensure that users can perform tasks easily and without complexity, such as:

- Entering data
- Updating information
- Deleting records

```
if(confirm("Are you sure?")) {
   // delete
}
```

5-Security Considerations

Security Considerations refer to the measures and practices taken to protect a system, its data, and users from unauthorized access, attacks, and potential threats.

These considerations include:

- Protecting data from hacking or misuse
- Preventing attacks such as SQL Injection and XSS
- Ensuring only authorized users can access or modify data
- Validating and sanitizing user inputs

Purpose:
To maintain the confidentiality, integrity, and availability of the system and its data.

```
cursor.execute("SELECT * FROM students WHERE id=?", (id,))
```

كامل شرح .
CRUD

https://www.youtube.com/watch?v=9ylj9NR0Lcg