**ACCOUNT MANAGEMENT SYSTEM**

A Project Presented to the

Faculty of the College of Computer Studies Northern Bukidnon State College

In Partial Fulfillment

of the Requirements for the Degree Bachelor of Science in Information Technology

Ron Andre S. Manuel

IT26 Fundamentals of Database Systems

June 2025

**Page**

# 1



## TABLE OF CONTENTS

**Page**

|  |
| --- |
| **TITLE PAGE** |
| **TABLE OF CONTENTS 1** |

1. Introduction 2
2. ERD 3
3. SQL Join Integration 4
4. Implementation 5
5. Features 6
6. Manual 7

## INTRODUCTION

**Page**

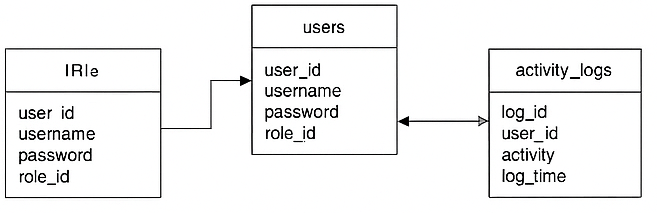
2



**1.1 Overview**

This project is a simple Login and Logout Web Application with a Dashboard, built using PHP and MySQL. It allows users to log in securely, view a dashboard populated with data using SQL joins, and perform basic CRUD (Create, Read, Update, Delete) operations on user records. This system demonstrates an understanding of relational databases, database manipulation, and secure session handling in web development.

## ENTITY RELATIONSHIP DIAGRAM

****

**Figure 1: ERD**

**Page**

# 3

The system uses three main tables:

* **Users**: stores user credentials and role references.
* **Roles**: stores role names (e.g., Administrator, User).
* **Activity Logs**: records login activities linked to user IDs.

**Relationships**:

* users.role\_id is a foreign key referencing roles.role\_id (many-to-one).
* activity\_logs.user\_id is a foreign key referencing users.user\_id (many-to-one).

These relationships demonstrate proper normalization and enable meaningful joins across the system.



**SQL Join Integration**

**Sample code showing the join used (put comment):**

**Sample SQL Join Query**:

-- Display users and their roles

SELECT u.user\_id, u.username, r.role\_name

FROM users u

LEFT JOIN roles r ON u.role\_id = r.role\_id;

**Comment**: This LEFT JOIN ensures all users are displayed, even if their role is null.

**Another Example (Activity Logs)**:

-- Show activity logs with usernames

SELECT l.log\_id, u.username, l.activity, l.log\_time

FROM activity\_logs l

INNER JOIN users u ON l.user\_id = u.user\_id;

**Comment**: This INNER JOIN shows only logs that have a valid user associated.

**Page**

# 4

## IMPLEMENTATION

**Page**

# 5



* 1. **Technologies Used**
* **PHP** (server-side scripting)
* **MySQL** (database)
* **HTML/CSS** (basic frontend)
* **XAMPP** (development environment)
  1. **Key Code Snippets and Functions**

**Login Authentication**:

$password = hash('sha256', $\_POST['password']);

$stmt = $conn->prepare("SELECT \* FROM users WHERE username=? AND password=?");

**CRUD - Insert User**:

$stmt = $conn->prepare("INSERT INTO users (username, password, role\_id) VALUES (?, ?, ?)");

$stmt->bind\_param("ssi", $username, $password, $role\_id);

**JOIN to Get User Roles**:

$sql = "SELECT u.user\_id, u.username, r.role\_name FROM users u LEFT JOIN roles r ON u.role\_id = r.role\_id";

## FEATURES

**Page**

# 6



* 1. **System Features**
* **Login Page**: Allows users to securely log in using SHA-256 password hashing.
* **Dashboard**: Displays user data and their assigned roles using SQL joins.
* **Create User**: Add new users with role assignments.
* **Update User**: Edit username and change roles.
* **Delete User**: Delete records with confirmation.
* **Activity Logging**: Logs every successful login attempt.
  1. **User Instructions**

## MANUAL

**Page**

# 7



* **Login**: Visit login.php, enter username and password, and click "Login".

A white background with black dots

AI-generated content may be incorrect.

* **Add a User**: Click on "Add User", fill out the form, and click submit.

A white background with black dots

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

* **Edit a User**: On the dashboard, click "Edit" next to the user to update info.



* **Delete a User**: Click "Delete" and confirm the action.

A black rectangular object with a blue button

AI-generated content may be incorrect.

* **Logout**: Click the "Logout" link on the dashboard.