A Project report on

EMPLOYEES SKILLS AND PROJECTS ALLOCATION SYSTEMS

Submitted in the partial fulfilment of requirements for the award of the Courses in

Database Management System

submitted by

PINNINTI THANUJA AF0256413

Under the esteemed guidance of

CHAITANYA MADAM

Lecturer in Anudip Foundation

Life. Transformed.

ANUDIP FOUNDATION

INDEX

ABSTRACT

The **Employee Skill Management and Project Allocation System** is designed to streamline the management of employee skills and efficiently allocate employees to projects based on their expertise. The system maintains comprehensive data on employees, including personal details, job titles, skills, and project history. Additionally, it tracks the ongoing and completed projects, ensuring that the right employees are assigned to projects that match their skill sets and availability.

Key components of the system include:

- **Employee Management**: A detailed database storing employee information such as names, job roles, department, experience, and contact details.
- **Skill Tracking**: The system categorizes and records employees' skills, maintaining proficiency levels for each skill.
- **Project Allocation**: Projects are recorded with details like department, location, and duration, and employees are assigned roles in these projects based on their skills.
- **Workload Management**: The system helps monitor employee allocation to prevent overburdening and ensure balanced resource distribution.
- **Query Functionality**: The system allows queries to find available employees, check project requirements, list allocated employees, and review an employee's project load.

The system aims to enhance resource management by ensuring that employees are appropriately assigned to projects according to their skill set, which increases productivity and project success rates.

INTRODUCTION

In the fast-evolving landscape of modern business, effective management of employee skills and project allocation is vital for enhancing productivity and achieving organizational goals. The Employee Skill Management and Project Allocation System is designed to optimize these processes, enabling organizations to match employees with suitable projects based on their skills and competencies.

3.1 Overview of the Project

The Employee Skill Management and Project Allocation System is a comprehensive database application that facilitates the tracking and management of employee skills while optimizing project assignments. This system is structured into several key components:

Employee Management: Centralized storage of employee information, including names, job titles, departments, and contact details.

Skill Tracking: A detailed framework for defining and recording the various skills that employees possess, along with their proficiency levels.

Project Management: Maintenance of project details, including project names, departments, start and end dates, and locations.

Allocation System: A mechanism for assigning employees to projects based on their skills, ensuring that the right talent is utilized effectively.

This project aims to streamline the skill management process, reduce administrative overhead, and improve project outcomes through better resource allocation.

3.2 Importance of Skill Management and Project Allocation

Effective skill management and project allocation are crucial for several reasons:

Optimized Resource Utilization: By accurately matching employee skills with project requirements, organizations can maximize the use of their workforce, leading to increased efficiency and productivity.

Enhanced Employee Satisfaction: Proper alignment of skills with projects can lead to higher job satisfaction among employees, as they are more likely to work on tasks that interest them and align with their strengths.

Improved Project Outcomes: Projects are more likely to succeed when team members possess the necessary skills and expertise. This alignment fosters collaboration and innovation, leading to better results.

Career Development: Skill management allows organizations to identify gaps in employee skills, facilitating targeted training and development initiatives. This supports employees' career growth and aligns their aspirations with organizational needs.

Agility in Project Assignments: The ability to quickly assess and allocate resources based on current skills and project demands allows organizations to adapt to changing market conditions and project requirements effectively.

In summary, the Employee Skill Management and Project Allocation System not only enhances operational efficiency but also contributes to employee engagement and project success, making it an essential tool for modern organizations.

OBJECTIVE OF THE PROJECT

The primary objective of the **Employee Skill Management and Project Allocation System** is to streamline the management of employee skills and automate the process of allocating employees to projects based on their expertise. This system is designed to ensure that employees with the right skills and proficiency levels are matched to projects that require those specific abilities, thereby improving project efficiency and employee productivity.

The system aims to:

1. **Efficient Skill Management:** Maintain a structured database of employee skills, allowing easy identification and categorization based on proficiency levels.
2. **Accurate Project Allocation:** Automate the allocation of employees to projects, ensuring that the most suitable employees are assigned based on their skills, experience, and availability.
3. **Track Employee Expertise:** Provide a clear overview of employees' skillsets, allowing organizations to identify skill gaps, plan training sessions, and improve workforce development.
4. **Optimize Resource Utilization:** Ensure the optimal use of employee resources by tracking ongoing project assignments and employee availability.
5. **Support Data-Driven Decisions:** Help management make informed decisions regarding staffing for current and future projects, based on real-time data about employee skills and project requirements.

By achieving these objectives, the system ensures better management of human resources and improved project outcomes, contributing to the overall success of the organization.

SYSTEM REQUIREMENTS

Hardware Requirements:

1. **Processor**: Dual-Core or higher (e.g., Intel Core i3 or better)
2. **Memory (RAM)**: 4 GB minimum, 8 GB recommended
3. **Storage**: 20 GB minimum available space
4. **Network**: Basic internet connection or local network access

Software Requirements:

1. **Operating System**: Windows, macOS, or Linux (any modern version)
2. **Database**: MySQL or MariaDB (version 8.0+)
3. **Programming Language**: SQL for database operations

LITERATURE REVIEW

The **Employee Skill Management and Project Allocation System** focuses on efficiently managing employee skills and automating project assignments based on those skills. The review highlights current challenges and the need for improvement in existing systems.

1. Current Skill Management Methods

Most companies use manual methods or basic software tools to track employee skills, which are often inefficient and prone to errors. Advanced HR systems can manage skills but usually lack integration with project assignment features.

Gap: Existing systems don't fully automate the process of matching employees to suitable projects, leading to underutilization of skills.

2. Project Allocation Techniques

Traditional project allocation is often manual, with managers assigning tasks based on personal judgment, which can be biased or inefficient. Newer methods use software, but many are complex and require specialized knowledge to implement effectively.

Gap: There's a lack of simple, user-friendly systems that integrate skill data with project needs in real-time.

3. Database Integration

Relational databases like MySQL are commonly used for storing and managing data related to employees, skills, and projects. Proper database design enhances system efficiency and data retrieval.

Gap: While databases manage data well, many systems fail to integrate skill tracking with real-time project assignments effectively.

4. Existing Tools

Tools like Jira and Microsoft Project help with project management but do not focus on matching employees to projects based on their skills and proficiency levels.

Gap: Most project management tools lack features for tracking individual skills and automating employee assignments based on those skills.

SYSTEM DESIGN

System Design is a crucial phase in the development of the **Employee Skill Management and Project Allocation System**. It involves defining the architecture, components, modules, interfaces, and data for the system to satisfy specified requirements.

7.1 Architecture Diagram

The Architecture Diagram illustrates the high-level structure of the system, showing how the various components interact with each other. This system is designed using a three-tier architecture consisting of the Presentation Layer (User Interface), Business Logic Layer (Application Logic), and Data Layer (Database).

Presentation Layer: This layer handles user interactions through a web-based interface, allowing HR managers to input employee data, skills, and projects.

Business Logic Layer: This is where the main processing happens, including skill matching, project allocation, and reporting.

Data Layer: This layer manages all data storage, retrieval, and manipulation using MySQL.

7.2 ER Diagram (Entity-Relationship Diagram)

The ER Diagram shows the system's key entities and their relationships. Key entities include:

Employees: Stores employee details like ID, Name, Job Title, Department, etc.

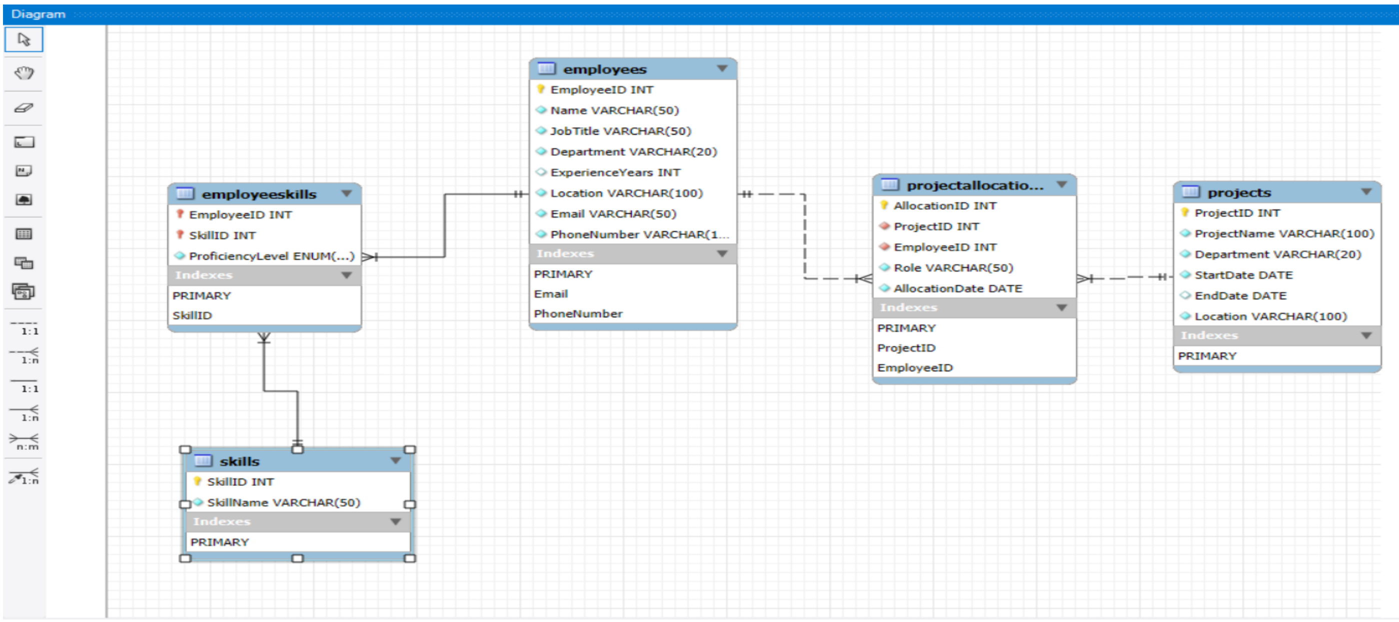
Skills: Contains information about different skills available in the organization.

Projects: Details about ongoing and upcoming projects.

Employee Skills: A junction table mapping employees to their skills with proficiency levels.

Project Allocations: Manages which employee is allocated to which project, including roles and allocation dates.

The diagram visually represents the primary keys, foreign keys, and relationships (one-to-many, many-to-many) between these entities.



7.3 Data Flow Diagram (DFD)

The DFD illustrates how data flows through the system, highlighting key processes and data stores.

Level 0 (Context Diagram): Shows the entire system as a single process with external entities like HR Managers and Employees interacting with it.

Level 1: Breaks down the main processes, such as Employee Management, Skill Management, and Project Allocation, showing data interactions among them.

Level 2: Further details sub-processes like adding a new skill, updating project details, and matching skills to projects.

7.4 UML Diagrams

UML (Unified Modeling Language) diagrams provide a standardized way to visualize the design of the system.

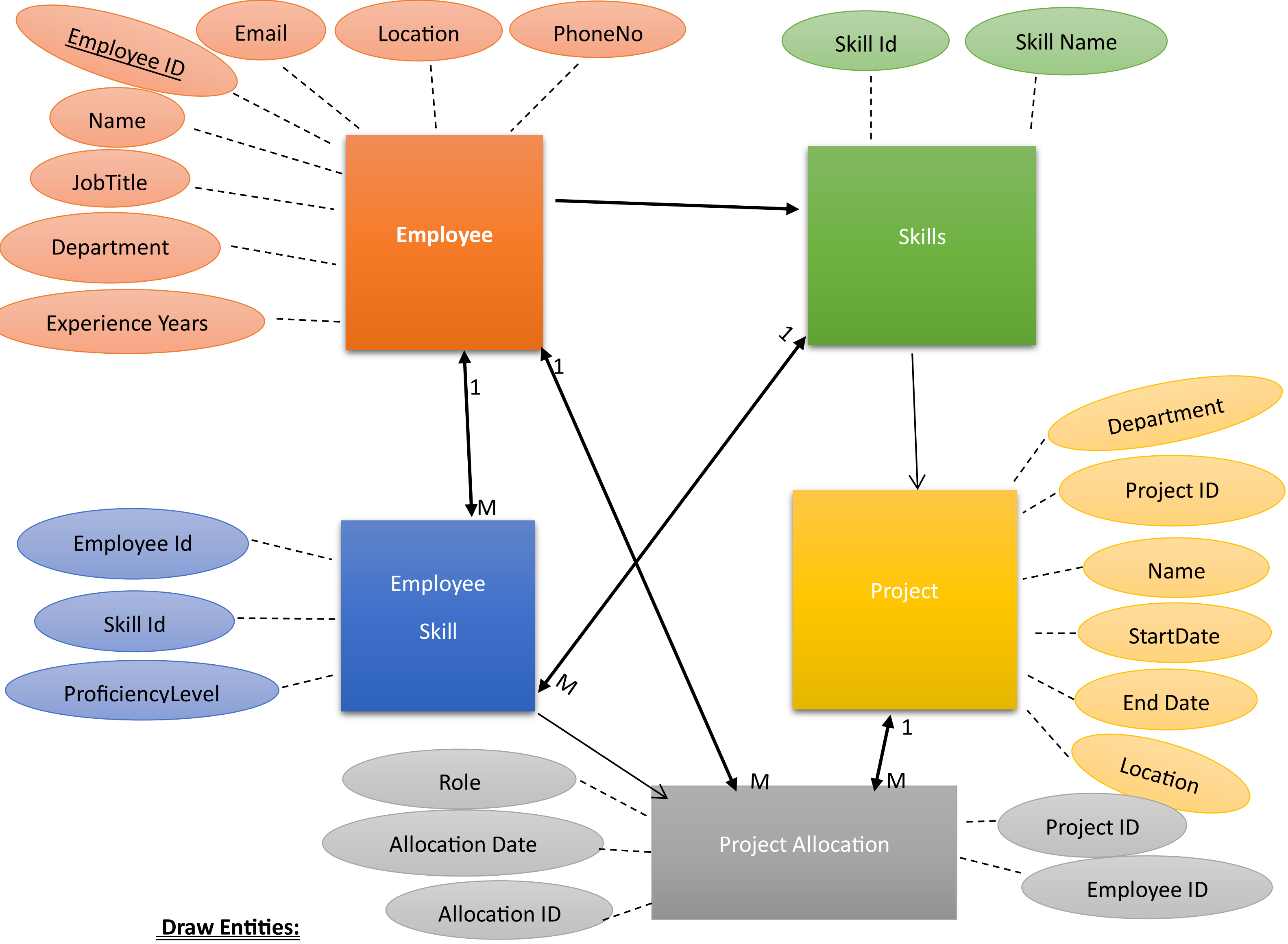
Use Case Diagram: Depicts the different actors (HR Manager, Employee) and their interactions with the system's functionalities, such as adding employees, managing skills, and allocating projects.

Class Diagram: Shows the classes, their attributes, and methods, along with the relationships between different classes like Employee, Skill, Project, etc.

Sequence Diagram: Illustrates the order of operations in key processes, such as the sequence of steps when allocating an employee to a project based on skill matching.

Activity Diagram: Displays the flow of control in the system's processes, highlighting decision points and parallel processes.

These diagrams collectively help in understanding the system's workflow, data interactions, and overall design, ensuring a robust implementation of the **Employee Skill Management and Project Allocation System**.



Draw Entities:

Employees: Include the fields EmployeeID (PK), Name, JobTitle, Department,ExperienceYears, Location, Email, PhoneNumber.

Skills: Include the fields SkillID (PK) and SkillName.

EmployeeSkills: This is the junction table for the many-to-many relationship between Employees and Skills. Include EmployeeID (FK), SkillID (FK), and ProficiencyLevel.

Projects: Include ProjectID (PK), ProjectName, Department, StartDate, EndDate, and Location.

ProjectAllocations: Include AllocationID (PK), ProjectID (FK), EmployeeID (FK), Role, and AllocationDate.

Draw Relationships:

Employees ↔ EmployeeSkills: One employee can have multiple skills. This is a one-to-many relationship between Employees and EmployeeSkills.

Skills ↔ EmployeeSkills: One skill can be assigned to multiple employees. This is a one-to-many relationship between Skills and EmployeeSkills.

Employees ↔ ProjectAllocations: One employee can be allocated to multiple projects. This is a one-to-many relationship.

Projects ↔ ProjectAllocations: One project can have multiple employees assigned. This is a one-to-many relationship.

METHODOLOGY

The methodology outlines the structured approach followed to develop the Employee Skill Management and Project Allocation System. The steps include gathering requirements, designing the system, creating the database, implementing the code, and conducting tests to ensure the system functions correctly.

8.1 Requirement Gathering

Identify the needs of the HR team, including managing employee data, tracking skills, and allocating projects.

Define key functionalities such as adding employees, assigning skills, matching employees to projects, and generating reports.

Collect input from stakeholders to refine system requirements.

8.2 System Design

Design a three-tier architecture with Presentation, Business Logic, and Data Layers.

Use diagrams like Architecture Diagram and Data Flow Diagrams (DFDs) to illustrate the interactions and data movement.

Define how the system components will interact, ensuring modularity and ease of maintenance.

8.3 Database Design

Create a relational database schema with tables like Employees, Skills, Projects, and ProjectAllocations.

Design the relationships between tables to ensure data integrity, such as foreign key constraints for employee-project allocations.

Normalize the database to eliminate redundancy and optimize performance.

8.4 Implementation

Develop the system using suitable technologies such as Java for backend, MySQL for the database, and HTML/CSS for the frontend.

Code each module, including employee management, skill tracking, and project allocation, following best practices for clean and maintainable code.

Integrate the frontend and backend to provide a seamless user experience.

8.5 Testing

Conduct unit testing on individual modules to ensure they function as intended.

Perform integration testing to verify that different components work together correctly.

Carry out user acceptance testing (UAT) with HR managers to ensure the system meets their needs and is ready for deployment.

This methodology ensures a systematic approach to building a robust and efficient system, aligning closely with the project’s objectives.

DATABASE DESIGN

The database design focuses on organizing and structuring the data used in the Employee Skill Management and Project Allocation System. The design includes multiple tables that capture various aspects of employee data, skills, projects, and allocations.

9.1 Employees Table

Purpose: Stores information about employees, including their personal details, job titles, and contact information.

Key Fields: EmployeeID (Primary Key), Name, JobTitle, Department, ExperienceYears, Location, Email, PhoneNumber.

9.2 Skills Table

Purpose: Contains all the skills that employees can possess, which are used for matching skills to project requirements.

Key Fields: SkillID (Primary Key), SkillName.

9.3 Employee Skills Table

Purpose: Manages the many-to-many relationship between employees and skills, capturing each employee's skill set and proficiency level.

Key Fields: EmployeeID, SkillID, ProficiencyLevel (with proficiency levels such as Beginner, Intermediate, and Advanced).

9.4 Projects Table

Purpose: Stores details of the projects that employees can be allocated to, including project names, departments, and timelines.

Key Fields: ProjectID (Primary Key), ProjectName, Department, StartDate, EndDate, Location.

9.5 Project Allocations Table

Purpose: Tracks which employees are allocated to specific projects, along with their roles and allocation dates.

Key Fields: AllocationID (Primary Key), ProjectID, EmployeeID, Role, AllocationDate.

This structured database design supports efficient data management, ensuring that all relevant information about employees, skills, and projects is stored accurately and can be easily accessed for system operations.

IMPLEMENTATION

10.1 Technologies Used

Database Management System: MySQL

Development Environment: MySQL Workbench (or any SQL IDE like DBeaver, phpMyAdmin)

Operating System: Any OS that supports MySQL (e.g., Windows, Linux, macOS)

Programming Languages: SQL for database queries and operations.

Other Tools:

Version Control: Git (optional)

Documentation: Markdown or any document editor

10.2 Code Snippets

-- <EMPLOYEE SKILL MANAGEMENT AND PROJECT ALLOCATION SYSTEM>

-- Creating the Employees database

```
CREATE DATABASE Employees;
```

-- Use the Employees database

```
USE Employees;
```

-- Create the Employees table

```
CREATE TABLE Employees (EmployeeID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(50)
NOT NULL, JobTitle VARCHAR(50) NOT NULL, Department VARCHAR(20) NOT NULL, ExperienceYears
INT CHECK (ExperienceYears >= 0), Location VARCHAR(100) NOT NULL, Email VARCHAR(50) UNIQUE
NOT NULL CHECK(Email REGEXP'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.com$'), PhoneNumber
VARCHAR(10) UNIQUE NOT NULL CHECK (PhoneNumber REGEXP '^[0-9]{10}$'));
```

-- Insert data into Employees

```
INSERT INTO Employees (Name, JobTitle, Department, ExperienceYears, Location, Email,
PhoneNumber) VALUES('Thanuja', 'Developer', 'Full Stack Java', 7, 'Pune', 'thanuja@gmail.com',
'9123456789'),('Davika', 'Developer', 'Full Stack Python', 8, 'Chennai', 'davika@gmail.com',
'9876543210'),('Vara Laxmi', 'Web Designer', 'HTML, CSS, JS', 6, 'Chennai', 'laxmi@gmail.com',
'9333456789'),('Vandana', 'Data Analyst', 'SQL', 2, 'Bangalore', 'vandana@gmail.com', '9999567893'),
('Shivarjun', 'Senior Software Engineer', 'Full Stack Java', 2, 'Bangalore', 'arjun@gmail.com',
'9956565423'),('Vinay', 'Developer', '.NET', 8, 'Mumbai', 'vinay@gmail.com', '9321657893'),('Sanjay',
'AI/ML Research Scientist', 'C++', 1, 'Bangalore', 'sanjay@gmail.com', '9955415493'),('Upender',
'Technical Program Manager', 'JavaScript', 5, 'Hyderabad', 'upender@gmail.com',
'9357689421'),('Madava', 'UI/UX Designer', 'HTML, CSS, JS', 2, 'Hyderabad', 'madhava@gmail.com',
'9142357689'),('Dharama Teja', 'Marketing Manager', 'Market Research', 4, 'Bangalore',
'teja@gmail.com', '9254678903'),('Ananya', 'Data Scientist', 'SQL', 6, 'Mumbai',
'ananaya@gmail.com', '9000132456');
```

-- Select all records from Employees

```
SELECT * FROM Employees;
```

-- TABLE FOR STORING SKILLS

```sql
CREATE TABLE Skills ( SkillID INT PRIMARY KEY AUTO_INCREMENT,  SkillName VARCHAR(50) NOT NULL );
```

-- Insert sample data into Skills table

```sql
INSERT INTO Skills (SkillName) VALUES ('Full Stack Java'), ('Full Stack Python'), ('HTML, CSS, JS'), ('SQL'), ('C++'), ('.NET'), ('Project Management');
```

-- Select data from Skills

```sql
SELECT * FROM Skills;
```

-- JUNCTION TABLE TO STORE EMPLOYEE SKILLS (MANY-TO-MANY RELATIONSHIP)

```sql
CREATE TABLE EmployeeSkills (EmployeeID INT, SkillID INT, ProficiencyLevel ENUM('Beginner', 'Intermediate', 'Advanced') NOT NULL,  PRIMARY KEY (EmployeeID, SkillID),  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE CASCADE, FOREIGN KEY (SkillID) REFERENCES Skills(SkillID) ON DELETE CASCAD);
```

-- INSERT SAMPLE DATA INTO EMPLOYEESKILLS TABLE

```sql
INSERT INTO EmployeeSkills (EmployeeID, SkillID, ProficiencyLevel) VALUES (1, 1, 'Advanced'), (2, 2, 'Advanced'), (3, 3, 'Intermediate'), (4, 4, 'Intermediate'), (5, 1, 'Advanced'), (6, 6, 'Advanced'), (7, 5, 'Beginner'), (8, 7, 'Advanced'), (9, 3, 'Intermediate'), (10, 7, 'Beginner'), (11, 4, 'Advanced');
```

-- Select data from EmployeeSkills

```sql
SELECT * FROM EmployeeSkills;
```

-- TABLE FOR STORING PROJECT DETAILS

```sql
CREATE TABLE Projects (ProjectID INT PRIMARY KEY AUTO_INCREMENT, ProjectName VARCHAR(100) NOT NULL, Department VARCHAR(20) NOT NULL, StartDate DATE NOT NULL, EndDate DATE,Location VARCHAR(100) NOT NULL);
```

-- INSERTING MORE SAMPLE DATA INTO THE PROJECTS TABLE

```sql
INSERT INTO Projects (ProjectName, Department, StartDate, EndDate, Location) VALUES ('Mobile Banking App', 'Software Development', '2024-09-20', '2025-03-30', 'Mumbai'), ('Cloud Migration Strategy', 'Cloud Infrastructure', '2024-11-01', NULL, 'Hyderabad'), ('Data Security Enhancement', 'Cybersecurity', '2024-09-01', '2024-12-15', 'Pune'), ('Machine Learning Fraud Detection', 'AI/ML', '2024-07-10', '2024-11-30', 'Bangalore'), ('Employee Performance Management System', 'HR Systems', '2024-05-05', '2024-09-15', 'Chennai'), ('E-commerce Analytics Dashboard', 'Data Science', '2024-06-01', '2024-10-10', 'Delhi'), ('IoT Smart Home System', 'IoT Development', '2024-08-20', '2025-01-20', 'Bangalore'), ('Blockchain-Based Supply Chain Management', 'Blockchain', '2024-10-05', '2025-04-30', 'Mumbai'), ('Real-time Stock Market Prediction', 'Data Science', '2024-07-01', '2024-12-20', 'Chennai'), ('Customer Relationship Management System', 'Software Development', '2024-08-15', NULL, 'Hyderabad');
```

-- Verify data in the Projects table

```sql
SELECT * FROM Projects;
```

-- TABLE FOR ALLOCATING EMPLOYEES TO PROJECTS

```
CREATE TABLE ProjectAllocations ( AllocationID INT PRIMARY KEY AUTO_INCREMENT,  ProjectID INT NOT NULL,  EmployeeID INT NOT NULL,  Role VARCHAR(50) NOT NULL,  AllocationDate DATE NOT NULL,  FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID) ON DELETE CASCADE, FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE CASCAD);
```

-- INSERT SAMPLE DATA INTO THE PROJECTALLOCATIONS TABLE

```
INSERT INTO ProjectAllocations (ProjectID, EmployeeID, Role, AllocationDate) VALUES (4, 5, 'Senior Full Stack Developer', '2024-09-20'), (5, 8, 'Cloud Architect', '2024-11-01'), (6, 3, 'Cybersecurity Analyst', '2024-09-01'), (7, 6, 'Lead ML Engineer', '2024-07-10'), (8, 10, 'Project Manager', '2024-08-15'), (9, 11, 'Data Scientist', '2024-06-01'), (10, 9, 'UI/UX Designer', '2024-08-20'), (1, 2, 'AI/ML Lead Developer', '2024-10-05'), (2, 1, 'Lead Full Stack Developer', '2024-09-15');
```

-- Select data from ProjectAllocations

```
SELECT * FROM ProjectAllocations;
```

-- Find Available Employees for a Specific Skill

```
SELECT e.Name, e.JobTitle, es.ProficiencyLevel FROM Employees e JOIN EmployeeSkills es ON e.EmployeeID = es.EmployeeID JOIN Skills s ON es.SkillID = s.SkillID WHERE s.SkillName = 'Full Stack Java' AND e.EmployeeID NOT IN ( SELECT pa.EmployeeID FROM ProjectAllocations pa JOIN Projects p ON pa.ProjectID = p.ProjectID WHERE p.EndDate IS NULL OR p.EndDate >= CURDATE());
```

```
SELECT * FROM Skills WHERE SkillName = 'Full Stack Java';
```

-- List Employees Assigned to a Specific Project

```
SELECT e.Name, e.JobTitle, pa.Role FROM Employees e JOIN ProjectAllocations pa ON e.EmployeeID = pa.EmployeeID JOIN Projects p ON pa.ProjectID = p.ProjectID WHERE p.ProjectName = 'AI-Based Chatbot';
```

```
SELECT * FROM Projects WHERE ProjectName = 'AI-Based Chatbot';
```

-- Check Project Skill Requirements and Matching Employees

```
SELECT e.Name, e.JobTitle, es.ProficiencyLevel, s.SkillName FROM Employees e JOIN EmployeeSkills es ON e.EmployeeID = es.EmployeeID JOIN Skills s ON es.SkillID = s.SkillID JOIN Projects p ON p.ProjectID = 5 WHERE s.SkillName = 'Project Management';
```

```
SELECT * FROM Projects WHERE ProjectID = 5;
```

```
SELECT * FROM Skills WHERE SkillName = 'Project Management';
```

-- Assign Employee to a New Project

```
INSERT INTO ProjectAllocations (ProjectID, EmployeeID, Role, AllocationDate) VALUES (8, 7, 'Blockchain Developer', '2024-10-05');
```

```
SELECT * FROM Employees WHERE EmployeeID = 7;
```

```
SELECT * FROM Projects WHERE ProjectID = 8;
```

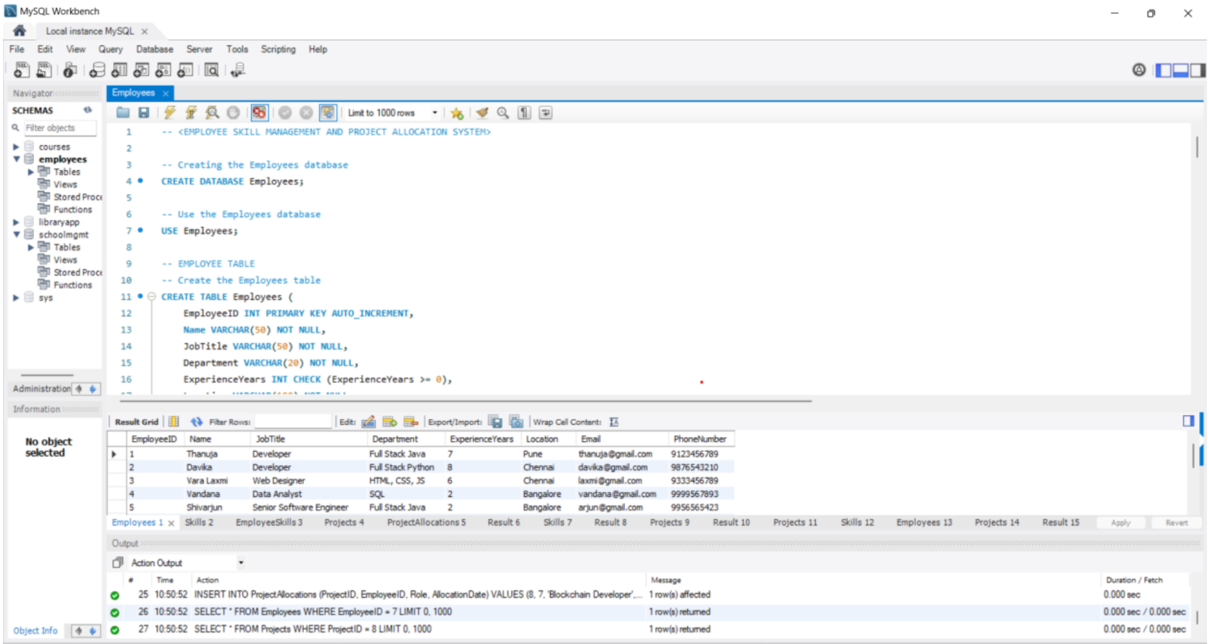
-- Check Employee Allocation Load

```
SELECT e.Name, COUNT(pa.ProjectID) AS TotalProjects FROM Employees e JOIN ProjectAllocations pa ON e.EmployeeID = pa.EmployeeID GROUP BY e.EmployeeID;
```

10.3 System Screenshots

While I can't create screenshots directly, you can capture and include the following types of screenshots in your documentation:

1. **Database Structure**: A screenshot of the database schema showing the Employees, Skills, Projects, and ProjectAllocations tables in MySQL Workbench or your chosen SQL IDE.
2. **Sample Data View**: Screenshots of the sample data in the Employees, Skills, and Projects tables after running the insertion queries.
3. **Query Results**: Screenshots displaying the output of various SELECT queries, such as available employees for a specific skill or the list of employees assigned to a project.
4. **ER Diagram (Optional)**: If you have created an Entity-Relationship (ER) diagram for your database, including that would provide a clear visual representation of your data relationships.



```
-- <EMPLOYEE SKILL MANAGEMENT AND PROJECT ALLOCATION SYSTEM>

-- Creating the Employees database
CREATE DATABASE Employees;

-- Use the Employees database
USE Employees;

-- EMPLOYEE TABLE
-- Create the Employees table
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(50) NOT NULL,
    JobTitle VARCHAR(50) NOT NULL,
    Department VARCHAR(20) NOT NULL,
    ExperienceYears INT CHECK (ExperienceYears >= 0),
```

EmployeeID	Name	JobTitle	Department	ExperienceYears	Location	Email	PhoneNumber
1	Thanuja	Developer	Full Stack Java	7	Pune	thanuja@gmail.com	9123456789
2	Davika	Developer	Full Stack Python	8	Chennai	davika@gmail.com	9876543210
3	Vara Laxmi	Web Designer	HTML, CSS, JS	6	Chennai	laxmi@gmail.com	9333456789
4	Vandana	Data Analyst	SQL	2	Bangalore	vandana@gmail.com	9999567893
5	Shivarjun	Senior Software Engineer	Full Stack Java	2	Bangalore	arjun@gmail.com	9956565423

#	Time	Action	Message	Duration / Fetch
25	10:50:52	INSERT INTO ProjectAllocations (ProjectID, EmployeeID, Role, AllocationDate) VALUES (8, 7, 'Blockchain Developer',...	1 row(s) affected	0.000 sec
26	10:50:52	SELECT * FROM Employees WHERE EmployeeID = 7 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
27	10:50:52	SELECT * FROM Projects WHERE ProjectID = 8 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

RESULTS AND TESTING

11.1 System Testing

System testing checks if everything works correctly. Here are the main tests performed:

Database Connection

Ensured the MySQL database connected successfully.

CRUD Operations

Create: Added records to the Employees, Skills, Projects, and ProjectAllocations tables.

Read: Retrieved data using SELECT queries to confirm that the data is accurate.

Update: Changed some records in the Employees table and checked if the updates were applied.

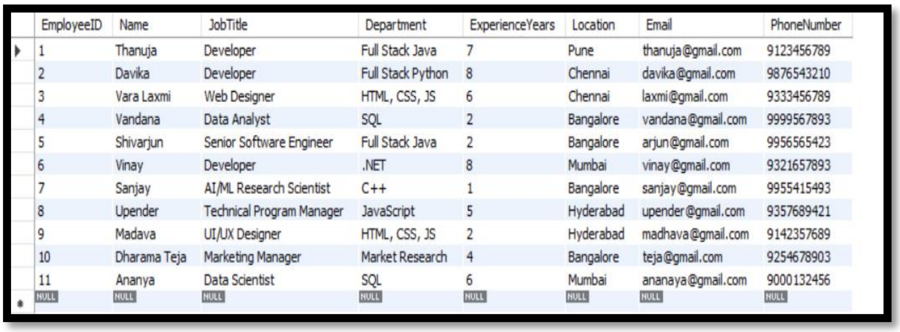
Delete: Removed sample records to ensure they were deleted successfully.

Data Integrity

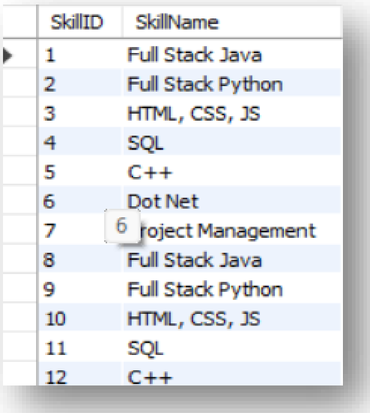
Verified that unique constraints (like email and phone numbers) worked and foreign key relationships were maintained.

11.2 Sample Query Outputs

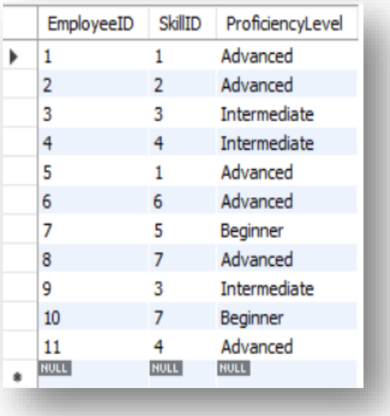
Here are examples of outputs from some key queries:



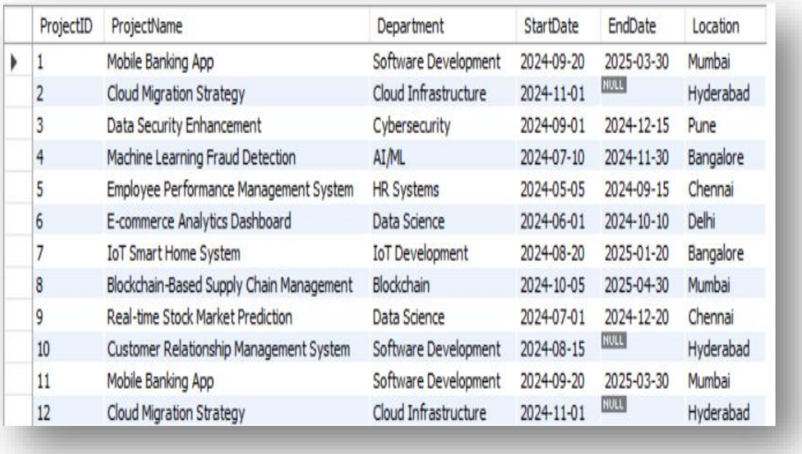
EmployeeID	Name	JobTitle	Department	ExperienceYears	Location	Email	PhoneNumber
1	Thanuja	Developer	Full Stack Java	7	Pune	thanuja@gmail.com	9123456789
2	Davika	Developer	Full Stack Python	8	Chennai	davika@gmail.com	9876543210
3	Vara Laxmi	Web Designer	HTML, CSS, JS	6	Chennai	laxmi@gmail.com	9333456789
4	Vandana	Data Analyst	SQL	2	Bangalore	vandana@gmail.com	9999567893
5	Shivarjun	Senior Software Engineer	Full Stack Java	2	Bangalore	arjun@gmail.com	9956565423
6	Vinay	Developer	.NET	8	Mumbai	vinay@gmail.com	9321657893
7	Sanjay	AI/ML Research Scientist	C++	1	Bangalore	sanjay@gmail.com	9955415493
8	Upender	Technical Program Manager	JavaScript	5	Hyderabad	upender@gmail.com	9357689421
9	Madava	UI/UX Designer	HTML, CSS, JS	2	Hyderabad	madhava@gmail.com	9142357689
10	Dharama Teja	Marketing Manager	Market Research	4	Bangalore	teja@gmail.com	9254678903
11	Ananya	Data Scientist	SQL	6	Mumbai	ananaya@gmail.com	9000132456
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



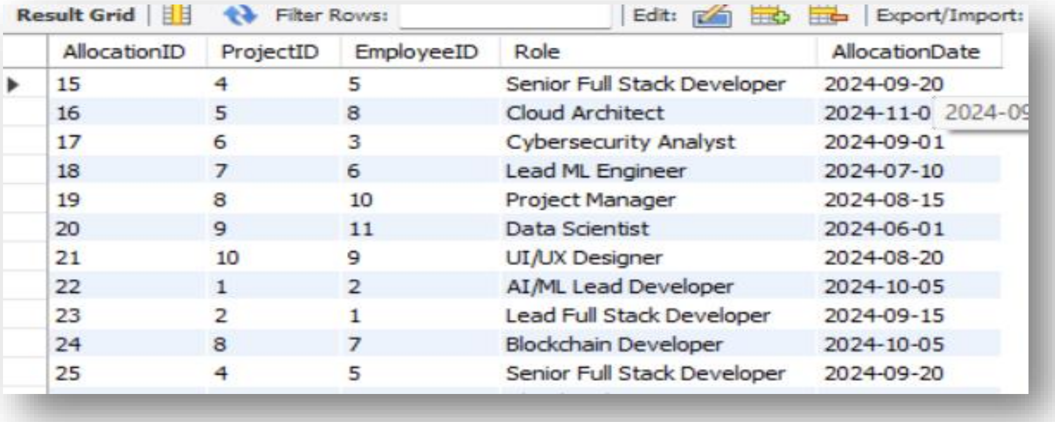
SkillID	SkillName
1	Full Stack Java
2	Full Stack Python
3	HTML, CSS, JS
4	SQL
5	C++
6	Dot Net
7	6 roject Management
8	Full Stack Java
9	Full Stack Python
10	HTML, CSS, JS
11	SQL
12	C++



EmployeeID	SkillID	ProficiencyLevel
1	1	Advanced
2	2	Advanced
3	3	Intermediate
4	4	Intermediate
5	1	Advanced
6	6	Advanced
7	5	Beginner
8	7	Advanced
9	3	Intermediate
10	7	Beginner
11	4	Advanced
NULL	NULL	NULL



ProjectID	ProjectName	Department	StartDate	EndDate	Location
1	Mobile Banking App	Software Development	2024-09-20	2025-03-30	Mumbai
2	Cloud Migration Strategy	Cloud Infrastructure	2024-11-01	NULL	Hyderabad
3	Data Security Enhancement	Cybersecurity	2024-09-01	2024-12-15	Pune
4	Machine Learning Fraud Detection	AI/ML	2024-07-10	2024-11-30	Bangalore
5	Employee Performance Management System	HR Systems	2024-05-05	2024-09-15	Chennai
6	E-commerce Analytics Dashboard	Data Science	2024-06-01	2024-10-10	Delhi
7	IoT Smart Home System	IoT Development	2024-08-20	2025-01-20	Bangalore
8	Blockchain-Based Supply Chain Management	Blockchain	2024-10-05	2025-04-30	Mumbai
9	Real-time Stock Market Prediction	Data Science	2024-07-01	2024-12-20	Chennai
10	Customer Relationship Management System	Software Development	2024-08-15	NULL	Hyderabad
11	Mobile Banking App	Software Development	2024-09-20	2025-03-30	Mumbai
12	Cloud Migration Strategy	Cloud Infrastructure	2024-11-01	NULL	Hyderabad



AllocationID	ProjectID	EmployeeID	Role	AllocationDate
15	4	5	Senior Full Stack Developer	2024-09-20
16	5	8	Cloud Architect	2024-11-0 2024-09
17	6	3	Cybersecurity Analyst	2024-09-01
18	7	6	Lead ML Engineer	2024-07-10
19	8	10	Project Manager	2024-08-15
20	9	11	Data Scientist	2024-06-01
21	10	9	UI/UX Designer	2024-08-20
22	1	2	AI/ML Lead Developer	2024-10-05
23	2	1	Lead Full Stack Developer	2024-09-15
24	8	7	Blockchain Developer	2024-10-05
25	4	5	Senior Full Stack Developer	2024-09-20

CONCLUSION

The Employee Skill Management and Project Allocation System has been developed and successfully implemented using SQL, showcasing the potential of SQL in managing complex organizational data. This system serves several critical functions that significantly enhance the management of employee information and project assignments.

Firstly, the system allows for **efficient data management** by centralizing employee records, skill sets, and project details in a structured manner. By using relational databases, the data is organized into distinct tables, making it easy to access and update information as needed. This organization helps HR personnel and project managers quickly find relevant employee details, promoting timely decision-making.

Moreover, the system enhances **project allocation** efficiency by matching employees to projects based on their skill levels and experience. This capability ensures that projects are staffed with the most qualified individuals, ultimately leading to better project outcomes. The use of SQL queries facilitates rapid searching and filtering of employees, thus streamlining the assignment process.

Another significant advantage of the system is its emphasis on **data integrity**. By implementing various SQL constraints—such as primary keys, foreign keys, and unique constraints—the system prevents data duplication and inconsistencies. This level of control ensures that all entries are accurate and reliable, which is crucial for making informed management decisions.

The inclusion of a junction table to manage the many-to-many relationship between employees and their skills further enhances the system's capability. This design allows for detailed tracking of individual proficiency levels, ensuring that the right talent is available for each project requirement.

In addition to operational benefits, this system can also contribute to strategic workforce planning. By analyzing employee skills and project assignments, management can identify skill gaps and plan future training initiatives accordingly. This proactive approach to skill management ensures that the organization remains competitive in a rapidly changing environment.

In conclusion, the Employee Skill Management and Project Allocation System effectively leverages SQL to improve HR functions within an organization. It promotes operational efficiency, enhances project outcomes, and ensures data integrity, making it a valuable tool for modern workforce management. Moving forward, further enhancements and additional features could be implemented to expand the system's capabilities, such as integrating analytics for deeper insights into employee performance and project success rates.

FUTURE SCOPE

The Employee Skill Management and Project Allocation System has laid a solid foundation for managing employee skills and project assignments. However, there are several opportunities for enhancement and expansion that can further improve its functionality and user experience.

1. **Advanced Analytics**: Implementing analytical tools can provide insights into employee performance and project success rates. By analyzing historical data, organizations can make informed decisions about resource allocation and skill development.
2. **Integration with Other Systems**: The system can be integrated with existing HR and project management tools. This integration would facilitate seamless data exchange, improving overall efficiency and enabling a more holistic view of employee performance and project progress.
3. **User Interface Improvements**: Developing a user-friendly graphical interface can enhance usability for HR personnel and project managers. A web-based dashboard could provide visual representations of employee skills, project statuses, and allocation metrics, making the system more intuitive.
4. **Real-time Updates**: Incorporating real-time data updates would allow for immediate changes to employee allocations and skill assessments. This feature would enhance responsiveness to project needs and enable more agile management practices.
5. **Mobile Accessibility**: Developing a mobile application could increase accessibility for managers and employees, allowing them to check project assignments and update their skills on the go. This flexibility can improve engagement and communication within teams.
6. **Enhanced Reporting Features**: Adding more comprehensive reporting options would allow managers to generate detailed reports on employee skills, project allocations, and performance metrics. This feature could assist in strategic planning and performance evaluations.
7. **Training and Development Tracking**: Implementing a module for tracking employee training and development programs would help organizations identify skill gaps and promote continuous learning. This approach encourages employees to upskill and stay relevant in their roles.
8. **Feedback Mechanism**: Incorporating a feedback system where employees can assess project experiences and skill requirements would foster a culture of continuous improvement. Managers could use this feedback to adjust project teams and identify training needs.
9. **Scalability**: Ensuring the system can scale to accommodate growth within the organization is essential. As new employees and projects are added, the database should efficiently handle increased data volumes without performance degradation.
10. **Artificial Intelligence Integration**: Leveraging AI could enhance the system's ability to match employees with projects by analyzing vast amounts of data to predict optimal allocations based on skills, past performance, and project requirements.

By focusing on these areas, the Employee Skill Management and Project Allocation System can evolve into a comprehensive tool that not only meets current organizational needs but also adapts to future challenges in workforce management. This adaptability will ultimately contribute to increased productivity, employee satisfaction, and overall organizational success.

REFERENCES

1. MySQL Documentation: MySQL Official Documentation

2. SQL Tutorial: W3Schools SQL Tutorial

APPENDICES

- **Appendix A**: Full SQL Code for Database and Tables.
- **Appendix B**: Example Queries Used.
- **Appendix C**: Testing Results (if needed).