

# Database Mini Project Report

**Project Title: Freelancer Payment & Project Tracker**

**Name:** Kammari Vijaya Tejaswini

**Date:** 01/05/2025

BY

## Objective:

To design and implement a database system that helps **freelancers** manage clients, projects, tasks, invoices, and payments. The system enables tracking of deadlines, payment statuses, pending dues, and overdue invoices.

## Tables Used:

Table Name	Purpose
Clients	Stores client information (name, email, phone)
Projects	Stores project details linked to clients
Project_Tasks	Stores tasks/milestones for each project
Invoices	Stores invoice details and payment status
Payments	Stores payment records linked to invoices

## Relationships:

- One **Client** → Many **Projects**
- One **Project** → Many **Tasks, Invoices**
- One **Invoice** → Many **Payments**

## Database Schema (SQL):

```
CREATE TABLE Clients (  
    client_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    phone VARCHAR(15)  
);
```

```
CREATE TABLE Projects (  
    project_id INT PRIMARY KEY AUTO_INCREMENT,  
    client_id INT,  
    project_name VARCHAR(100),  
    start_date DATE,  
    end_date DATE,  
    status VARCHAR(20),  
    FOREIGN KEY (client_id) REFERENCES Clients(client_id)  
);
```

```
CREATE TABLE Project_Tasks (  
    task_id INT PRIMARY KEY AUTO_INCREMENT,  
    project_id INT,  
    task_name VARCHAR(100),  
    due_date DATE,  
    is_completed BOOLEAN,  
    FOREIGN KEY (project_id) REFERENCES Projects(project_id)  
);
```

```
CREATE TABLE Invoices (  
    invoice_id INT PRIMARY KEY AUTO_INCREMENT,  
    project_id INT,  
    amount DECIMAL(10,2),  
    due_date DATE,  
    is_paid BOOLEAN,  
    FOREIGN KEY (project_id) REFERENCES Projects(project_id)  
);
```

```
CREATE TABLE Payments (  
    payment_id INT PRIMARY KEY AUTO_INCREMENT,  
    invoice_id INT,  
    paid_amount DECIMAL(10,2),  
    payment_date DATE,  
    FOREIGN KEY (invoice_id) REFERENCES Invoices(invoice_id)  
);
```

## Sample Data Inserted:

-- Clients

```
INSERT INTO Clients (name, email, phone) VALUES
('Alice Smith', 'alice@example.com', '1234567890'),
('Bob Kumar', 'bob@example.com', '9876543210');
```

-- Projects

```
INSERT INTO Projects (client_id, project_name, start_date, end_date,
status) VALUES
(1, 'Website Redesign', '2025-07-01', '2025-08-01', 'Ongoing'),
(2, 'Mobile App Development', '2025-06-15', '2025-08-15',
'Ongoing');
```

-- Tasks

```
INSERT INTO Project_Tasks (project_id, task_name, due_date,
is_completed) VALUES
(1, 'Design UI Mockups', '2025-07-10', TRUE),
(1, 'Develop Frontend', '2025-07-20', FALSE),
(2, 'Setup Backend', '2025-07-25', FALSE);
```

-- Invoices

```
INSERT INTO Invoices (project_id, amount, due_date, is_paid) VALUES
(1, 5000.00, '2025-07-15', FALSE),
(2, 8000.00, '2025-07-25', TRUE);
```

-- Payments

```
INSERT INTO Payments (invoice_id, paid_amount, payment_date) VALUES
(2, 8000.00, '2025-07-20');
```

## Sample Queries & Outputs:

### 1. Total Pending Amount per Client:

SELECT

```
    c.name AS client_name,  
    SUM(i.amount) - IFNULL(SUM(p.paid_amount), 0) AS pending_amount  
FROM Clients c  
JOIN Projects pr ON c.client_id = pr.client_id  
JOIN Invoices i ON pr.project_id = i.project_id  
LEFT JOIN Payments p ON i.invoice_id = p.invoice_id  
GROUP BY c.client_id, c.name;
```

The screenshot shows a SQL query editor with a 'Schema SQL' tab on the left and a 'Query SQL' tab on the right. The 'Query SQL' tab contains the query for 'Total Pending Amount per Client'. Below the query editor, the 'Results' section shows the output of the query. The results are displayed in a table with two columns: 'client\_name' and 'pending\_amount'. The table contains two rows: 'Alice Smith' with a pending amount of 5000.00, and 'Bob Kumar' with a pending amount of 0.00. The execution time is 0.48ms.

client_name	pending_amount
Alice Smith	5000.00
Bob Kumar	0.00

### 2. Overdue Invoices:

SELECT

```
    i.invoice_id, pr.project_name, i.amount, i.due_date  
FROM Invoices i  
JOIN Projects pr ON i.project_id = pr.project_id  
WHERE i.is_paid = FALSE AND i.due_date < CURDATE();
```

The screenshot shows a SQL query editor with a 'Schema SQL' tab on the left and a 'Query SQL' tab on the right. The 'Query SQL' tab contains the query for 'Overdue Invoices'. Below the query editor, the 'Results' section shows the output of the query. The results are displayed in a table with four columns: 'invoice\_id', 'project\_name', 'amount', and 'due\_date'. The table contains one row: '1' for 'Website Redesign' with an amount of 5000.00 and a due date of 2025-07-15. The execution time is 1.48ms.

invoice_id	project_name	amount	due_date
1	Website Redesign	5000.00	2025-07-15

### 3. Incomplete Tasks:

SELECT

```
    pt.task_name, pr.project_name, pt.due_date
FROM Project_Tasks pt
JOIN Projects pr ON pt.project_id = pr.project_id
WHERE pt.is_completed = FALSE;
```

The screenshot shows a SQL IDE interface. On the left, the 'Schema SQL' pane contains DDL for tables: Tasks, Invoices, and Payments. On the right, the 'Query SQL' pane shows a query to select task names, project names, and due dates from Project\_Tasks, joined with Projects, where the task is not completed. Below these panes, the 'Results' section shows the execution of 'Query #1' with a table of 3 rows and 3 columns: task\_name, project\_name, and due\_date.

task_name	project_name	due_date
Develop Frontend	Website Redesign	2025-07-20
Setup Backend	Mobile App Development	2025-07-25

### 4. List all ongoing projects with their client names

SELECT

```
    pr.project_name,
    pr.status,
    c.name AS client_name
FROM Projects pr
JOIN Clients c ON pr.client_id = c.client_id
WHERE pr.status = 'Ongoing';
```

The screenshot shows a SQL IDE interface. On the left, the 'Schema SQL' pane contains DDL for Clients and Projects tables. On the right, the 'Query SQL' pane shows a query to select project names, status, and client names from Projects, joined with Clients, where the project status is 'Ongoing'. Below these panes, the 'Results' section shows the execution of 'Query #1' with a table of 2 rows and 3 columns: project\_name, status, and client\_name.

project_name	status	client_name
Website Redesign	Ongoing	Alice Smith
Mobile App Development	Ongoing	Bob Kumar

## 5.Show All Incomplete Project Tasks

SELECT

pt.task\_name,  
pr.project\_name,  
pt.due\_date

FROM Project\_Tasks pt

JOIN Projects pr ON pt.project\_id = pr.project\_id

WHERE pt.is\_completed = FALSE;

Schema SQL

```
1 CREATE TABLE Clients (  
2   client_id INT PRIMARY KEY AUTO_INCREMENT,  
3   name VARCHAR(100),  
4   email VARCHAR(100),  
5   phone VARCHAR(15)  
6 );  
7  
8 CREATE TABLE Projects (  
9   project_id INT PRIMARY KEY AUTO_INCREMENT,  
10  client_id INT,  
11  project_name VARCHAR(100),  
12  start_date DATE,  
13  end_date DATE,  
14  status VARCHAR(20),
```

Text to DDL

Query SQL

Have any feedback?

```
1 SELECT  
2   pt.task_name,  
3   pr.project_name,  
4   pt.due_date  
5 FROM Project_Tasks pt  
6 JOIN Projects pr ON pt.project_id = pr.project_id  
7 WHERE pt.is_completed = FALSE;  
8
```

Results

Copy as Markdown

Query #1

Execution time: 0.27ms

task_name	project_name	due_date
Develop Frontend	Website Redesign	2025-07-20
Setup Backend	Mobile App Development	2025-07-25

## Conclusion:

This project simulates a real-world scenario where freelancers manage multiple projects and clients. It enables financial tracking, task management, and automated pending amount calculations. The design mimics platforms like Upwork and can serve as a backend for future web applications.