

Visveswaraya Technological University Belagavi – 590018, Karnataka



A Mini Project Report on “APARTMENT MANAGEMENT DATABASE”

Mini Project Report submitted in partial fulfilment of the requirement for the
DBMS Laboratory with Mini Project [18CSL58]

**Bachelor of Engineering
In
Computer Science and Engineering**

**Submitted By
Aditya Krishnan
1JT19CS004
Abhishek Kumar
1JT19CS002**



**Department of Computer Science and Engineering
Jyothy Institute of Technology,
Tataguni, Bengaluru – 560082**

**Department of Computer Science and Engineering
Jyothy Institute of Technology,
Tataguni, Bengaluru – 560082**



CERTIFICATE

Certified that the mini project work entitled "**APARTMENT MANAGEMENT DATABASE**" carried out by **Aditya Krishnan [1JT19CS004]** and **Abhishek Kumar [1JT19CS002]** bonafide students of Jyothy Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree

Mr. Arun Kumar N
Guide, Asst. Professor
Dept. of CSE

External Viva Examiner
1.
2.

Dr. Prabhanjan S
Professor & HOD
Dept. of CSE

Signature with Date:

ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution “**Jyothy Institute of Technology**” for providing us an opportunity to complete our project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Prabhanjan S, Professor and Head of Computer Science and Engineering Department** for providing for his valuable support.

We would like to thank our guides **Mr. Arun Kumar N , Assistant Professor** for their keen interest and guidance in preparing this work.

Finally, we would thank all our friends who have helped us directly or indirectly in this project.

**Aditya Krishnan [1JT19CS004]
Abhishek Kumar [1JT19CS002]**

ABSTRACT

The aim of this project is to create a functional application to manage the daily operations of apartments and simplify various aspects of apartment management such as rent collection, booking an apartment, availability of apartments etc that are normally very tedious to maintain using traditional methods.

This software helps them to digitize their records, which in turn saves a lot of time and money. For a manager or a company that owns multiple apartments, keeping track of each one is very difficult. They need to note down and maintain every detail for every apartment, keep track of apartments that are already booked, yet to be booked, payment status, etc. Expansion also bring in many challenges as their existing data need to be modified to reflect any additions.

In the proposed system, each tenant has his/her details stored in the database. The monthly rent can be entered for each resident which can save time. With this application, enable property managers to keep track and maintain records of their entire property easily. Any important information such as available blocks, flats, details of residents, rent paid can be found with just a click of a button.

TABLE OF CONTENTS

Sl No	Description	Page No
1	INTRODUCTION	6
2	DESIGN	9
3	IMPLEMENTATION & SNAPSHOT	13
4	CONNECTION WITH PHP MYADMIN	28
5	QUERIES RELATED SNAPSHOTS	36
6	CONCLUSION	43

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 Introduction to DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by many concurrent users, it is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

Databases are managed by a Database Management System(DBMS) which is a collection of programs that enables user to create and maintain a database.

Advantages of DBMS:

- Redundancy is controlled.
- Unauthorized access is restricted.
- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.

1.2 Introduction to SQL

Structured Query Language (SQL), is a language used to request data from a database which includes database creation, deletion, retrieval of required tables and even manipulation of data held in a relational database management system.

SQL is considered as a Non-Procedural or a High level language in which the expected result or operation is given without the specific details about how to accomplish the task. So, SQL is a declarative language.

Therefore, SQL is designed at a higher conceptual level of operation than procedural languages as procedural languages includes only the information about opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems, but the lower level logical and physical operations are not specified in SQL.

1.3 Introduction to Apartment Management Database

An apartment is a residential building, consisting of many units. Each unit is owned by the apartment owner, who leases it out to tenants.

The “Apartment Database” is a database that has information about each tenant and his respective apartment. It manages booking, expenses calculation and rent collection through a simple interface.

1.4 Scope and importance of work

The scope of the project is to create a webpage than enables users to overcome the drawbacks of traditional record keeping systems.

The database stores details about the tenant – name, phone number and dependents.

The apartment leased to the tenant is also recorded, along with the type of apartment and the respective block. Apartments have expenses such maintenance charges, repair charges and security charges, which are stored in the database for easy access. Rent due for each month is set by the owner.

CHAPTER 2

DESIGN

Theory of ER Diagram

The Entity–Relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An Entity Relationship Diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of database. ER diagrams are used to sketch out the design of a database.

ENTITIES

An entity is an ‘object’ in the real world with an independent existence and an entity type defines a collection (or set) of entities that have the same attributes. Each entity type in the database is described by its name and attributes. An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

RELATIONSHIPS

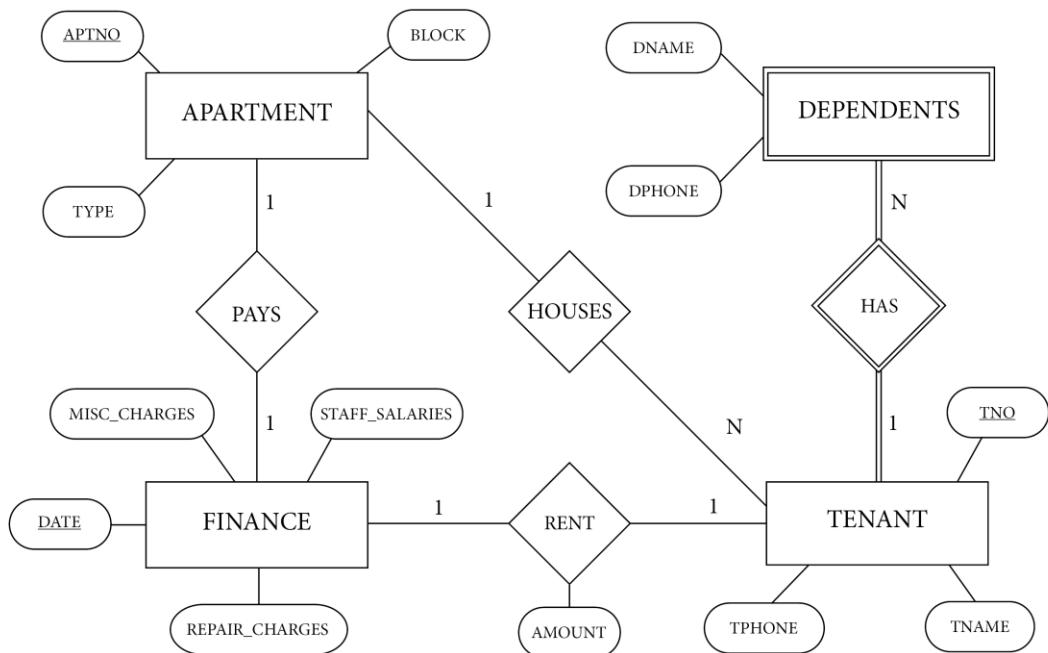
A relationship among two or more entities represents an association among the entities and whenever an attribute of one entity refers to another entity, there exists a relationship between the two entities.

In a relationship, a foreign key of one table refers the primary key of the other table and it is represented by diamond shape in ER diagram.

ATTRIBUTES

An attribute represents some property of interest that further describes an entity and the column header of the table shows the attributes. Each attribute in a table has a certain domain which allows it to accept a certain ‘set of values’ only. The attribute values, of each entity, will define its characteristics in the table and is represented by oval in the ER diagram.

ENTITY RELATIONSHIP DIAGRAM



SCHEMA DIAGRAM

LIST OF TABLES

APARTMENTS:

- APTN
- TYPE
- BLOCK

TENANT :

- TNO
- TNAME
- TPHONE
- APTN

DEPENDENTS :

- TNO
- DNAME
- DPHONE

RENT :

- DATE
- TNO
- AMOUNT

FINANCE :

- DATE
- REPAIR_CHARGES
- STAFF_SALARIES
- MISC_CHARGES
- APTNO

CHAPTER 3

IMPLEMENTATION & SNAPSHOT

CREATION:

- 1 CREATE TABLE tenant (TNO INTEGER(5) PRIMARY KEY, TNAME VARCHAR(40), TPHONE INTEGER(10));
- 2 CREATE TABLE apartment (APTNO INTEGER(4) PRIMARY KEY, TNO INTEGER(5), FOREIGN KEY(TNO) REFERENCES TENANT(TNO) ON DELETE CASCADE, TYPE VARCHAR(20), BLOCK INTEGER(2));
- 3 CREATE TABLE dependent (TNO INTEGER(5), FOREIGN KEY(TNO) REFERENCES TENANT(TNO) ON DELETE CASCADE, DNAME VARCHAR(20), DPHONE INTEGER(10));
- 4 CREATE TABLE finance (DATE DATE PRIMARY KEY, REPAIR_CHARGES INTEGER(11), STAFF_SALARIES INTEGER(11), MISC_CHARGES INTEGER(11), APTNO INTEGER(5), FOREIGN KEY(TNO) REFERENCES APRTMENT(APTNO) ON DELETE CASCADE);
- 5 CREATE TABLE rent (DATE DATE, FOREIGN KEY(DATE) REFERENCES FINANCE(TNO) ON DELETE CASCADE, TNO INTEGER(5), FOREIGN KEY(TNO) REFERENCES TENANT(TNO) ON DELETE CASCADE);
- 6 CREATE TABLE contact_us (
 `Name` text NOT NULL,
 `Email` varchar(30) NOT NULL,
 `Subject` varchar(30) NOT NULL,
 `Message` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERTION:

```
INSERT INTO apartment(`APTNO`, `TNO`, `TYPE`, `BLOCK`) VALUES  
(1001, 1, '9 bhk', 9),  
(1002, 2, '2BHK', 1),  
(1003, 3, '3BHK', 3),  
(1051, 5, '2 BHK', 2),  
(1078, 4, '4BHK', 1);
```

```
INSERT INTO dependent (`TNO`, `DNAME`, `DPHONE`) VALUES  
(1, 'harsh soni', 2147483647),  
(2, 'M SUMESH', 2147483647),  
(3, 'R NAMBIAR', 70787889),  
(4, 'DHIRU M', 860394489),  
(5, 'M SINGH', 60987867);
```

```
INSERT INTO finance (`DATE`, `REPAIR_CHARGES`,  
`STAFF_SALARIES`, `MISC_CHARGES`, `APTNO`) VALUES  
('0000-00-00', 340, 120, 250, 1002),  
('2021-01-12', 1200, 500, 270, 1001),  
('2022-01-27', 1208, 1201, 1110, 1003),  
('2022-01-29', 1500, 1234, 1000, 1078),  
('2022-01-31', 1000, 1110, 1200, 1051);
```

```
INSERT INTO rent (`DATE`, `TNO`, `AMOUNT`) VALUES  
('2021-01-12', 1, 20000),  
('0000-00-00', 2, 23000),  
('2022-01-27', 3, 25000),  
('2022-01-29', 4, 35000),  
('2022-01-31', 5, 20000);
```

```
INSERT INTO tenant (`TNO`, `TNAME`, `TPHONE`) VALUES  
(1, 'poorvi soni', 898765544),  
(2, 'Ajin Sumesh', 2147483647),  
(3, 'ADITYA KRISHNAN', 787971756),  
(4, 'VINAY M', 89876574),  
(5, 'BHUPINDER SINGH', 98887665);
```

```
INSERT INTO contact_us (`Name`, `Email`, `Subject`, `Message`) VALUES  
('ADITYA KRISHNAN', 'ADITYA123@GMAIL.COM', 'REGARDING  
2BHK HOUSE', 'HEY, THIS IS ADITYA I AM LOOKING FOR 2 BHK  
HOUSE NEAR J BLOCK ,AMAUSI ROAD LUCKNOW'),  
('bhupinder singh', 'BHUPI@GMAIL.COM', 'I WANT 3 BHK LAND NEAR  
KONKUNT', 'PLEASE REACH OUT ME \r\nMY WHATSAPP IS  
WA.ME/918603944898');
```

Screenshot of the phpMyAdmin interface showing the structure of the 'tenant' table in the 'apt_mgmt' database.

Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	TNO	int(5)	utf8mb4_general_ci		No	None			Change Drop More
2	TNAME	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	TPHONE	int(10)			Yes	NULL			Change Drop Primary Unique Index Spatial Fulltext

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	TNO	5	A	No	

Information:

Data	16.0 KB	Format	dynamic
Index	0 B	Collation	utf8mb4_general_ci
Overhead	8.8 MB	Next autoindex	8
Effective	-8,372,224 B	Creation	Jan 07, 2022 at 03:34 PM
Total	16.0 KB	Last update	Jan 18, 2022 at 10:15 PM
Optimize table		Last check	Jan 28, 2022 at 05:54 PM
Space usage		Row statistics	

Screenshot of the phpMyAdmin interface showing the structure of the 'apartment' table in the 'apt_mgmt' database.

Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	APTNO	int(4)	utf8mb4_general_ci		No	None			Change Drop More
2	TNO	int(5)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	TYPE	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	BLOCK	int(2)			Yes	NULL			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	APTNO	5	A	No	
Edit Rename Drop	TNO	BTREE	No	No	TNO	5	A	Yes	

Information:

Data	16.0 KB	Format	dynamic
Index	16.0 KB	Collation	utf8mb4_general_ci
Overhead	0 B	Next autoindex	0
Effective	32.0 KB	Creation	Jan 07, 2022 at 03:34 PM
Total	32.0 KB	Last update	Jan 18, 2022 at 10:15 PM
Optimize table		Last check	Jan 28, 2022 at 05:54 PM
Space usage		Row statistics	

Screenshot of the phpMyAdmin interface showing the structure of the 'dependent' table in the 'apt_mgmt' database.

Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	TIN	int(5)	utf8mb4_general_ci		Yes	NULL			Change Drop More
2	DNAME	varchar(40)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	DPHONE	int(10)			Yes	NULL			Change Drop Primary Unique Index Spatial Fulltext

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	TNO	BTREE	No	No	TNO	5	A	Yes	

Information:

Data	16.0 Kib	Format	dynamic
Index	16.0 Kib	Collation	utf8mb4_general_ci
Overhead	0 B	Next autoidx	8
Effective	32.0 Kib	Creation	Jan 07, 2022 at 03:34 PM
Total	32.0 Kib	Last update	Jan 18, 2022 at 10:15 PM
Optimize table			
Space usage Row statistics			

Screenshot of the phpMyAdmin interface showing the structure of the 'finance' table in the 'apt_mgmt' database.

Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	DATE	date	utf8mb4_general_ci		No	None			Change Drop More
2	REPAIR_CHARGES	int(11)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	STAFF_SALARIES	int(11)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	MISC_CHARGES	int(11)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	APTNNO	int(4)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Indexes:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	DATE	5	A	No	
Edit Rename Drop	APTNNO	BTREE	No	No	APTNNO	5	A	Yes	

Information:

Data	16.0 Kib	Format	dynamic
Index	16.0 Kib	Collation	utf8mb4_general_ci
Overhead	0 B	Next autoidx	8
Effective	32.0 Kib	Creation	Jan 07, 2022 at 03:34 PM
Total	32.0 Kib	Last update	Jan 18, 2022 at 10:16 PM
Optimize table			
Space usage Row statistics			

localhost / 127.0.0.1 / apt_mgmt | Apartment Management System | Apartment Management System | +

localhost/phpmyadmin/index.php?route=table/structure&server=1&db=apt_mgmt&table=rent

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	DATE	date	utf8mb4_general_ci		Yes	NULL			Change Drop More
2	TNO	int(5)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	AMOUNT	int(11)	utf8mb4_general_ci		Yes	NULL			Change Drop Primary Unique Index Spatial Fulltext

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	DATE	BTREE	No	No	DATE	5	A	Yes	
Edit Rename Drop	TNO	BTREE	No	No	TNO	5	A	Yes	

Create an index on columns [Go](#)

Partitions

No partitioning defined!

Partition table

Information

Data	16.0	KB	Format	dynamic
Index	32.0	KB	Collation	utf8mb4_general_ci
Overhead	0	B	Next autoindex	0
Effective	48.0	KB	Created	Jan 07, 2022 at 01:34 PM
Total	48.0	KB	Last update	Jan 28, 2022 at 10:16 PM
Optimize table				
Last check: Jan 28, 2022 at 01:56 PM				
Row statistics				

[Space usage](#)

[Console](#)

localhost / 127.0.0.1 / apt_mgmt | Apartment Management System | Apartment Management System | +

localhost/phpmyadmin/index.php?route=table/structure&server=1&db=apt_mgmt&table=contact_us

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Name	text	utf8mb4_general_ci		No	None			Change Drop More
2	Email	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	Subject	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	Message	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	Email	0	A	No	

Create an index on columns [Go](#)

Partitions

No partitioning defined!

Partition table

Information

Data	16.0	KB	Format	dynamic
Index	0	B	Collation	utf8mb4_general_ci
Overhead	0	B	Next autoindex	0
Effective	16.0	KB	Created	Jan 08, 2022 at 01:35 PM
Total	16.0	KB	Last update	Jan 28, 2022 at 01:56 PM
Optimize table				
Last check: Jan 28, 2022 at 01:56 PM				
Row statistics				

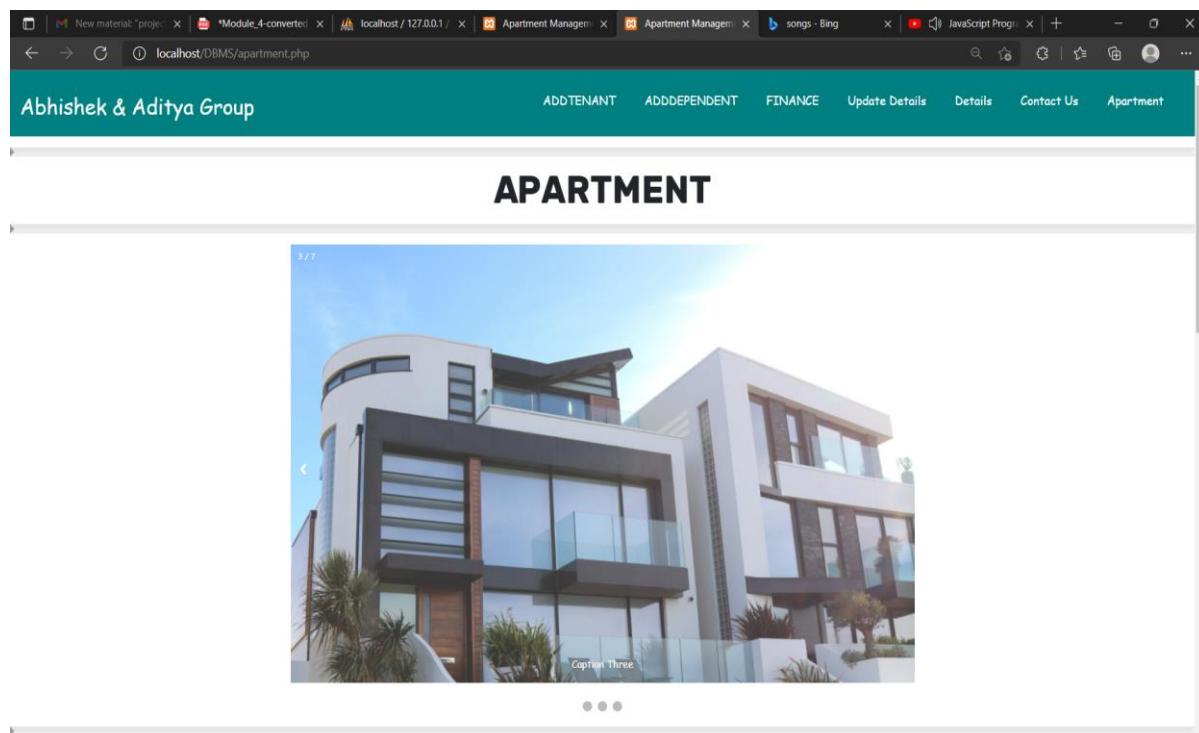
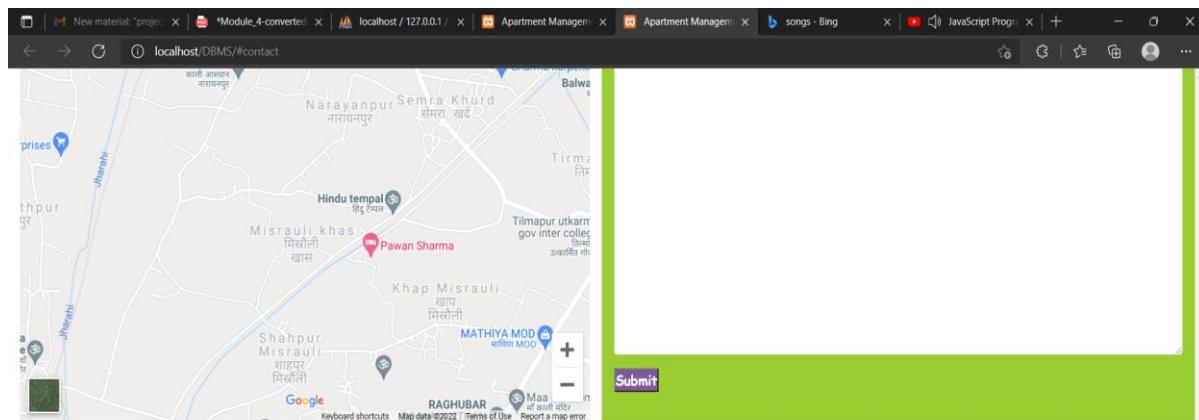
[Space usage](#)

[Console](#)

HOME PAGE

The screenshot shows a web browser window with multiple tabs open. The active tab displays the 'Abhishek & Aditya Group' website. The page features a navigation bar with links for ADDTENANT, ADDDEPENDENT, FINANCE, Update Details, Details, Contact Us, and Apartment. Below the navigation bar, a large section titled 'PHOTOS' contains a grid of several apartment building images. A small navigation arrow is visible at the bottom right of the photo grid.

The screenshot shows a web browser window with multiple tabs open. The active tab displays the 'Contact Us' page of the 'Abhishek & Aditya Group' website. The page features a map on the left showing the location of 'Creativeabhi13' in Khalwa, Bihar. The map includes labels for various landmarks such as Madarsa, Khap Banakat, Durga Temple, M/S Gangotri Group, Khalwa Aashram, Ankit Digital Service Center, Khalwa Village, Shrikant Singh, Smart Coaching Centre, Famous Manure Seed Store and Wholesaler, and Hanuman Mandir. To the right of the map is a green contact form area with the heading 'Let us know'. It includes fields for Name, Email, Subject, and Message, each with an associated input box.



ADD TENANT

The screenshot shows a web browser window with a teal header bar. The header contains the text "Abhishek & Aditya Group" on the left and navigation links "ADD TENANT", "ADDDPENDENT", "FINANCE", "Update Details", "Details", "Contact Us", and "Apartment" on the right. The main content area has a light blue background with a dark blue hexagonal pattern. It features a title "Add New Tenant" and a sub-section "Enter New Tenant Details". Below this are five input fields: "Name" (RAMKUWAR SINGH), "Apartment Number" (1006), "Phone" (8603944898), "Type" (3BHK), and "Block" (3). A "Submit" button is at the bottom. Below the form is a "Contact Us" section.

DATA INSERTED SUCCESSFULLY

The screenshot shows the same web browser window after a successful insertion. The header and main structure are identical to the previous screenshot. The "Enter New Tenant Details" section now displays placeholder text in all input fields: "Enter the Tenant Name", "Enter The Apartment Number", "Enter The Phone Number", "Enter The Type", and "Enter The Block". The "Submit" button remains at the bottom. Below the form is a "Contact Us" section.

The screenshot shows the phpMyAdmin interface for the 'apt_mgmt' database. The left sidebar lists databases: New, apt_mgmt, New, admin, apartment, contact_us, dependent, finance, rent, tenant, information_schema, mysql, performance_schema, phpmyadmin, sms, and test. The 'tenant' table is selected in the 'apt_mgmt' database. The table structure has columns TNO, TNAME, and TPHONE. The data shows four rows:

	TNO	TNAME	TPHONE
1	1	poochi soni	898765544
2	2	Ajin Sunresh	2147483647
3	3	arun kumar n	87676543
4	4	RAMKUWAR SINGH	2147483647

Below the table, there are buttons for Edit, Copy, Delete, and Export.

The screenshot shows the phpMyAdmin interface for the 'apt_mgmt' database. The left sidebar lists databases: New, apt_mgmt, New, admin, apartment, contact_us, dependent, finance, rent, tenant, information_schema, mysql, performance_schema, phpmyadmin, sms, and test. The 'apartment' table is selected in the 'apt_mgmt' database. The table structure has columns APTNO, TNO, TYPE, and BLOCK. The data shows four rows:

	APTNNO	TNO	TYPE	BLOCK
1	4	3	3bhk	2
2	1001	1	9 bhk	9
3	1002	2	2BHK	1
4	1006	4	3BHK	3

Below the table, there are buttons for Edit, Copy, Delete, and Export.

ADD DEPENDENT

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Add Dependent". The page has a teal header with the text "Abhishek & Aditya Group" and navigation links for ADDTENANT, ADDDEPENDENT, FINANCE, Update Details, Details, Contact Us, and Apartment. Below the header is a section titled "Add Dependent" with a sub-section titled "Enter Dependent Details". This section contains three input fields: "Tenant Number" (value: 4), "Dependent Name" (value: SAHDEO SINGH), and "Dependent Phone Number" (value: 78767574732). A "Submit" button is at the bottom. Below this form is a "Contact Us" section featuring a map and a "Let us know" input field.

DATA INSERTED SUCCESSFULLY

The screenshot shows the same web browser window after a successful data insertion. The message "Data Inserted Successfully" is displayed above the header. The rest of the page structure is identical to the previous screenshot, including the teal header, "Add Dependent" section, and "Contact Us" section.

The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'apt_mgmt'. The left sidebar lists various databases and tables. The main area displays the contents of the 'dependent' table. The table has two columns: 'TNO' and 'DNAME'. The data shows three rows:

TNO	DNAME
1	Hansi soni
2	M SUMESH
4	SANDEO SINGH

ADD FINANCE

The screenshot shows a web browser window with a teal header bar containing navigation links: ADDTENANT, ADDDEPENDENT, FINANCE, Update Details, Contact Us, and Apartment. The main content area is titled 'Update Finance'. It features a form with the following fields and values:

Field	Value
Tenant Number	4
Apartment Number	1006
Staff Salary	1201
Miscellaneous Charges	1501
Repair Charges	1250
Rent	30000
Date (YYYY-MM-DD)	2022-01-24

DATA INSERTED SUCCESSFULLY

The screenshot shows a web browser window with multiple tabs open. The active tab displays a success message: "Data Inserted Successfully". Below this, the page title is "Abhishek & Aditya Group". A navigation bar includes links for ADDTENANT, ADDDEPENDENT, FINANCE, Update Details, Details, Contact Us, and Apartment. The main content area features a large heading "Update Finance" and a form titled "Update Finance". The form fields include:

- Tenant Number (Text input)
- Apartment Number (Text input)
- Staff Salary (Text input)
- Miscellaneous Charges (Text input)
- Repair Charges (Text input)
- Rent (Text input)
- Date (Text input, format YYYY-MM-DD)

The screenshot shows the phpMyAdmin interface connected to a MySQL database named "apt_mgmt". The left sidebar lists databases like "New", "apt_mgmt", "information_schema", "mysql", "performance_schema", "phpmyadmin", "orms", and "test". The "rent" table is selected in the main query results pane. The table structure is shown with columns: DATE, TNO, and AMOUNT. The data in the table is:

DATE	TNO	AMOUNT
2021-01-12	1	20000
0900-00-00	2	23000
2022-01-24	4	30000

The screenshot shows the phpMyAdmin interface connected to a MySQL server (localhost: 127.0.0.1). The database selected is 'apt_mgmt'. The 'Table' tab is active, displaying the 'finance' table. The table has columns: DATE, REPAIR_CHARGES, STAFF_SALARIES, MISC_CHARGES, and APTNO. The data is as follows:

	DATE	REPAIR_CHARGES	STAFF_SALARIES	MISC_CHARGES	APTN0
1	2022-01-01	340	120	250	1002
2	2022-01-12	1200	500	270	1001
3	2022-01-24	1250	1201	1501	1006

TENANT DETAILS

The screenshot shows a web page titled "Tenant Details". The header includes links for ADDTENANT, ADDDEPENDENT, FINANCE, Update Details, Details, Contact Us, and Apartment. The page contains three main sections:

- Enter the Tenant Number:** A form with a text input field containing the value "4" and a "Submit" button.
- Delete tenant:** A section with the text "Enter the Tenant Number" and a text input field containing "Enter The Tenant Number".
- Contact Us:** A section with the text "Enter the Tenant Number" and a text input field containing "Enter The Tenant Number".

The screenshot shows a web browser window with the URL `localhost/DBMS/connect4.php`. The page contains four data entry forms:

- Tenant Details:**

TENANT NUMBER	TENANT NAME	TENANT PHONE
4	RAMKUWAR SINGH	2147483647
- Apartment Details:**

APARTMENT NUMBER	APARTMENT TYPE	APARTMENT BLOCK
1006	3BHK	3
- Dependent Details:**

DEPENDENT NAME	DEPENDENT PHONE
SAHDEO SINGH	2147483647
- Payment Details:**

PAYMENT DATE	TOTAL AMOUNT
2022-01-24	\$9000

A red button labeled "ADD MORE" is located at the bottom right of the page.

CHAPTER 4

CONNECTION WITH PHP

MYADMIN

```
<!DOCTYPE html>
<html>
<head>
    <title>CONNECT PHP</title>
</head>
<body>
    <center>
        <?php
            session_start();
            $conn = mysqli_connect("localhost", "root", "", "apt_mgmt");
            if($conn === false){
                die("ERROR: Could not connect. " . mysqli_connect_error());
            }
            $tname = $_REQUEST['tname'];
            $aptno = $_REQUEST['aptno'];
            $tphone = $_REQUEST['tphone'];
            $type = $_REQUEST['type'];
            $block = $_REQUEST['block'];
            for ($i = 1; $i <= 99999; $i++){
                $tno = mysqli_query($conn,"SELECT TNO FROM tenant WHERE TNO = '$i'");
                if (mysqli_num_rows($tno) == 0)
                {
                    $table1 = "INSERT INTO tenant VALUES ($i,$tname, $tphone)";
                    $table2 = "INSERT INTO apartment VALUES ($aptno,$i, '$type',$block)";
                    if(mysqli_query($conn, $table1)){
                        $_SESSION['status'] = "Data Inserted Successfully";
                        header('location: addtenant.php');
                    }
                    if(mysqli_query($conn, $table2)){
                        $_SESSION['status'] = "Data Inserted Successfully";
                        header('location: addtenant.php');
                    }
                    break;
                }
            }
            mysqli_close($conn);
        <a href="index.php">
            <button style="color:green">ADMIN</button>
        </a>
    </center>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>CONNECT PHP</title>
</head>
<body>
    <center>
        <?php
            session_start();
            for ($i = 1; $i <= 99999; $i++){
                $tno = mysqli_query($conn,"SELECT TNO FROM tenant WHERE TNO = '$i'");
                if (mysqli_num_rows($tno) == 0)
                {
                    $table1 = "INSERT INTO tenant VALUES ($i,$tname, $tphone)";
                    $table2 = "INSERT INTO apartment VALUES ($aptno,$i, '$type',$block)";
                    if(mysqli_query($conn, $table1)){
                        $_SESSION['status'] = "Data Inserted Successfully";
                        header('location: addtenant.php');
                    }
                    if(mysqli_query($conn, $table2)){
                        $_SESSION['status'] = "Data Inserted Successfully";
                        header('location: addtenant.php');
                    }
                    break;
                }
            }
            mysqli_close($conn);
        <a href="index.php">
            <button style="color:green">ADMIN</button>
        </a>
    </center>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>CONNECT PHP</title>
</head>
<body>
<center>
<?php
session_start();
$conn = mysqli_connect("localhost", "root", "", "apt_mgmt");
if($conn === false){
die("ERROR: Could not connect. " .
mysqli_connect_error());
}
$tno = $_REQUEST['tno'];
$dname = $_REQUEST['dname'];
$dphone = $_REQUEST['dphone'];

$table1 = "INSERT INTO dependent VALUES ($tno, '$dname', '$dphone')";

if(mysqli_query($conn, $table1)){
$_SESSION['status']="Data Inserted Successfully";
header('location: adddependent.php');
}

} else{
echo "ERROR: Hush! Sorry $sql. " .
mysqli_error($conn);
}

;mysqli_close($conn);
<a href="index.php">
<button style="color:black; background-color:red;">ADMIN</button>
</a>
</center>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>CONNECT PHP</title>
</head>
<body>
<center>
<?php
session_start();
$conn = mysqli_connect("localhost", "root", "", "apt_mgmt");

if($conn === false){
die("ERROR: Could not connect. " .
mysqli_connect_error());
}

$tno = $_REQUEST['tno'];
$aptno = $_REQUEST['aptno'];
$staff_sal = $_REQUEST['staff_sal'];

$misc_charges = $_REQUEST['misc_charges'];
$repaircharges = $_REQUEST['repaircharges'];
$rent = $_REQUEST['rent'];
$date = $_REQUEST['date'];

$table1 = "INSERT INTO finance VALUES ('$date', $repaircharges, $staff_sal, $misc_charges, $aptno)";

$table2 = "INSERT INTO rent VALUES ('$date', $tno, $rent)";

if(mysqli_query($conn, $table1)){
$_SESSION['status']="Data Inserted Successfully";
header('location: pricing.php');
}

if(mysqli_query($conn, $table2)){
$_SESSION['status']="Data Inserted Successfully";
}

</center>
</body>
</html>
```

```
$tno = $_REQUEST['tno'];
$aptno = $_REQUEST['aptno'];
$staff_sal = $_REQUEST['staff_sal'];

$misc_charges = $_REQUEST['misc_charges'];
$repaircharges = $_REQUEST['repaircharges'];
$rent = $_REQUEST['rent'];
$date = $_REQUEST['date'];

$table1 = "INSERT INTO finance VALUES ('$date','$repaircharges', $staff_sal,$misc_charges,$tno)";


```

```
<!DOCTYPE html>
<head>
    <title>CONNECT PHP</title>
</head>
<body>
    <center>
        <?php
        session_start();

        $conn = mysqli_connect("localhost", "root", "", "apt_mgmt");

        if($conn === false){
            die("ERROR: Could not connect. " .
                . mysqli_connect_error());
        }

        $name = $_REQUEST['name'];
        $email = $_REQUEST['email'];
        $subject = $_REQUEST['subject'];
        $message = $_REQUEST['message'];

        $sql = "INSERT INTO contact_us VALUES ('$name',
            '$email', '$subject', '$message')";

        if(mysqli_query($conn)){
            $_SESSION['status'] = "Data Inserted Successfully";
            header('location: index.php');
            /* echo "child:DATA STORED SUCCESSFULLY !"
            . " /n/n";
            echo nl2br("\n$name\n$email\n".
            . "$subject\n$message\n"); */

        } else{
            echo "ERROR: Sorry $sql. " .
                . mysqli_error($conn);
        }
    </?php>
</center>
</body>
</html>
```

A screenshot of Visual Studio Code showing a PHP file named `connect3.php`. The code handles a POST request to insert data into a MySQL database table. It checks if the connection is successful, then inserts the name, email, subject, and message into the `contact_us` table. If successful, it sets a session variable and displays a success message. If failed, it displays an error message. Finally, it closes the database connection and provides a link to the index page.

```
<?php
if($conn === false){
    die("ERROR: Could not connect. "
        . mysqli_connect_error());
}

$name = $_REQUEST['name'];
$email = $_REQUEST['email'];
$subject = $_REQUEST['subject'];
$message = $_REQUEST['message'];

$sql = "INSERT INTO contact_us VALUES ('$name',
    '$email', '$subject', '$message')";

if(mysqli_query($conn, $sql)){
    $_SESSION['status'] = "Data Inserted Successfully";
    header('location: index.php');
    /* echo "<h1>DATA STORED SUCCESSFULLY !</h1>";
    echo nl2br("<n>$name<n>$email<n>
    $subject<n>$message<n>"); */

} else{
    echo "ERROR: Hush! Sorry $sql. "
        . mysqli_error($conn);
}

mysqli_close($conn);
?>
<a href="index.php">
<button style="color:black; background-color:red;">ADMIN</button>
</a>
</center>
</body>
</html>
```

A screenshot of Visual Studio Code showing a PHP file named `connect.php`. The code connects to a MySQL database and retrieves tenant information. It starts by connecting to the database using `mysqli_connect`. If the connection fails, it dies with an error message. Otherwise, it fetches all rows from the `tenant` table and iterates through them, printing each row's values in a table format.

```
<?php
session_start();

$conn = mysqli_connect("localhost", "root", "", "apt_mgnt");

if($conn === false){
    die("ERROR: Could not connect. "
        . mysqli_connect_error());
}

$tno = $_REQUEST['tno'];
$result = mysqli_query($conn, "SELECT * FROM tenant where tno=$tno");
while($row = mysqli_fetch_array($result))
{
    (
        echo "<table border='1'>
        <tr>
        <th>TENANT NUMBER</th>
        <th>TENANT NAME</th>
        <th>TENANT PHONE</th>
        </tr>";
        echo "<tr>";
        echo "<td>" . $row['TNO'] . "</td>";
        echo "<td>" . $row['TNAME'] . "</td>";
        echo "<td>" . $row['TPHONE'] . "</td>";
        echo "</tr>";
        echo "<br/>";echo "<br/>";
    );
    echo "</table>";
}
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view of files in the DBMS folder. The `connect4.php` file is currently selected.
- Code Editor:** Displays the PHP code for `connect4.php`. The code connects to a MySQL database named `DBMS` and retrieves data from the `apartment`, `tenant`, and `dependent` tables to display it in tables.
- Output:** Shows the results of the database queries being executed.
- Terminal:** Shows the command `php connect4.php` being run.
- Status Bar:** Shows the current file is `connect4.php`, the line number is 1, and the character position is 1.

```
File Edit Selection View Go Run Terminal Help connect4.php - DBMS - Visual Studio Code

explorer DBMS
  - connect4.php
    42 echo "</table>";
    43 echo "<br>";
    44
    45
    46
    47 }
    48 $result = mysqli_query($conn,"SELECT APTNO,TYPE,BLOCK FROM apartment,tenant where tenant.tno=$tno and apartment.tno=tenant.tno");
    49
    50 while($row = mysqli_fetch_array($result))
    51 {
    52
    53   echo "<table border='2'>
    54     <tr>
    55       <th>APARTMENT NUMBER</th>
    56       <th>APARTMENT TYPE</th>
    57       <th>APARTMENT BLOCK</th>
    58     </tr></table>";
    59
    60   echo "<tr>";
    61   echo "<td>" . $row['APTNODE'] . "</td>";
    62   echo "<td>" . $row['TYPE'] . "</td>";
    63   echo "<td>" . $row['BLOCK'] . "</td>";
    64   echo "</tr>";
    65   echo "<br/>";echo "<br/>";
    66
    67   echo "</table>";
    68   echo "<br/>";
    69
    70 }
    71 $result = mysqli_query($conn,"SELECT DNAME,DPHONE FROM dependent,tenant where tenant.tno=$tno and dependent.tno=tenant.tno");
    72
    73 while($row = mysqli_fetch_array($result))
    74 {
    75
    76   echo "<table border='3'>
    77     <tr>
    78       <th>DEPENDENT NAME</th>
    79       <th>DEPENDENT PHONE</th>
    80     </tr></table>";
    81
    82   echo "<tr>";
    83   echo "<td>" . $row['DNAME'] . "</td>";
    84   echo "<td>" . $row['DPHONE'] . "</td>";
    85   echo "</tr>";
    86
    87   echo "<br/>";echo "<br/>";
    88
    89   echo "</table>";
    90   echo "<br/>";
    91
    92 }
```

```
<!DOCTYPE html>
<html>
<head>
<title>CONNECT PHP</title>
</head>
<body>
<center>
<?php
session_start();

$conn = mysqli_connect("localhost", "root", "", "apt_mgmt");

if($conn === false){
die("ERROR: Could not connect. "
. mysqli_connect_error());
}

$tname = $_REQUEST['tname'];
$aptno=$_REQUEST['aptno'];
$tno = $_REQUEST['tno'];
$tphone = $_REQUEST['phone'];
$type = $_REQUEST['type'];
$block = $_REQUEST['block'];
$dname = $_REQUEST['dname'];
$phone = $_REQUEST['dphone'];

$table1="update tenant set TNAME = '$tname' where TNO = '$tno'";
$table2="update tenant set TPHONE = '$tphone' where TNO = '$tno'";
$table3="update apartment set TYPE = '$type' where TNO = '$tno'";
$table4="update apartment set BLOCK = '$block' where TNO = '$tno';
```

```
$table1="update apartment set TYPE = '$type' where TNO = '$tno'";
$table2="update apartment set BLOCK = '$block' where TNO = '$tno'";
$table3="update dependent set DNAME = '$dname' where TNO = '$tno'";
$table4="update dependent set DPHONE = '$phone' where TNO = '$tno';

if(mysqli_query($conn, $table1)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

if(mysqli_query($conn, $table2)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

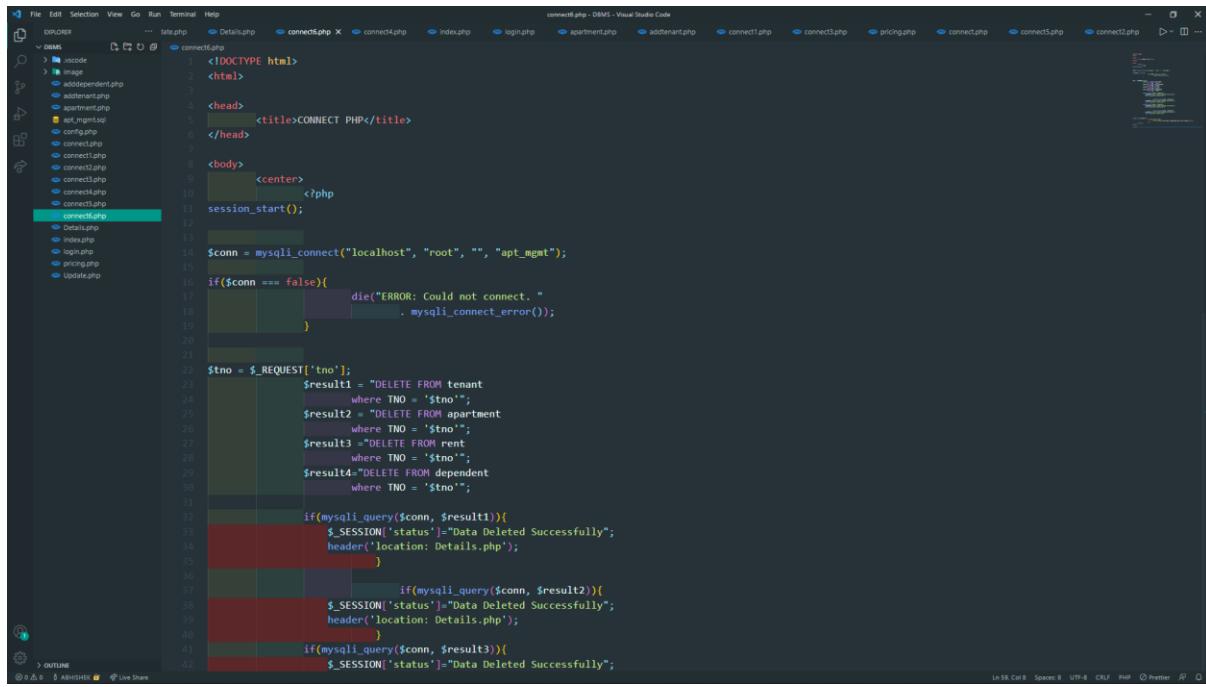
if(mysqli_query($conn, $table3)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

if(mysqli_query($conn, $table4)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

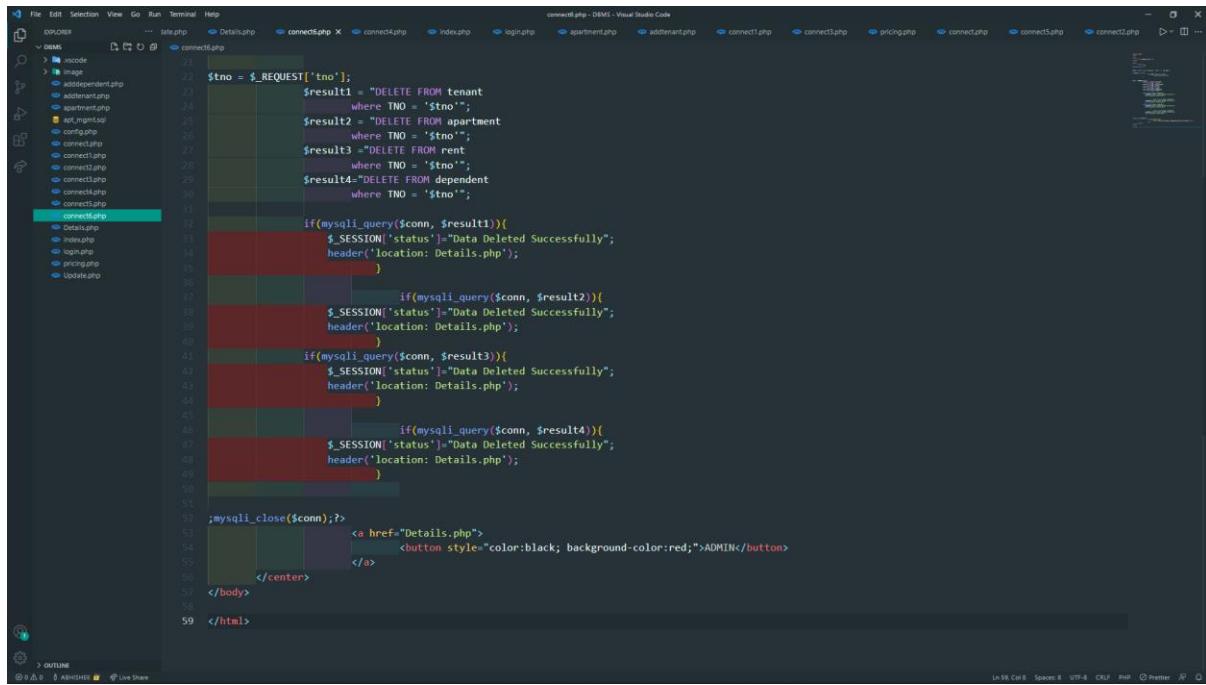
if(mysqli_query($conn, $table5)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

if(mysqli_query($conn, $table6)){
$_SESSION['status']="Data Updated Successfully";
header('location: Update.php');
}

:mysqli_close($conn);
?>
```



```
<!DOCTYPE html>
<html>
<head>
    <title>CONNECT PHP</title>
</head>
<body>
    <center>
        <?php
            session_start();
            $tno = $_REQUEST['tno'];
            $result1 = "DELETE FROM tenant
                        where TNO = '$tno'";
            $result2 = "DELETE FROM apartment
                        where TNO = '$tno'";
            $result3 = "DELETE FROM rent
                        where TNO = '$tno'";
            $result4 = "DELETE FROM dependent
                        where TNO = '$tno'";
            if(mysqli_query($conn, $result1)){
                $_SESSION['status'] = "Data Deleted Successfully";
                header('location: Details.php');
            }
            if(mysqli_query($conn, $result2)){
                $_SESSION['status'] = "Data Deleted Successfully";
                header('location: Details.php');
            }
            if(mysqli_query($conn, $result3)){
                $_SESSION['status'] = "Data Deleted Successfully";
                header('location: Details.php');
            }
            if(mysqli_query($conn, $result4)){
                $_SESSION['status'] = "Data Deleted Successfully";
                header('location: Details.php');
            }
        </?php>
        <a href="Details.php">
            <button style="color:black; background-color:red;">ADMIN</button>
        </a>
    </center>
</body>
</html>
```



```
<?php
    session_start();
    $tno = $_REQUEST['tno'];
    $result1 = "DELETE FROM tenant
                where TNO = '$tno'";
    $result2 = "DELETE FROM apartment
                where TNO = '$tno'";
    $result3 = "DELETE FROM rent
                where TNO = '$tno'";
    $result4 = "DELETE FROM dependent
                where TNO = '$tno'";
    if(mysqli_query($conn, $result1)){
        $_SESSION['status'] = "Data Deleted Successfully";
        header('location: Details.php');
    }
    if(mysqli_query($conn, $result2)){
        $_SESSION['status'] = "Data Deleted Successfully";
        header('location: Details.php');
    }
    if(mysqli_query($conn, $result3)){
        $_SESSION['status'] = "Data Deleted Successfully";
        header('location: Details.php');
    }
    if(mysqli_query($conn, $result4)){
        $_SESSION['status'] = "Data Deleted Successfully";
        header('location: Details.php');
    }
    ;mysql_close($conn);?
    <a href="Details.php">
        <button style="color:black; background-color:red;">ADMIN</button>
    </a>
</?php>
```

CHAPTER 5

QUERIES RELATED SNAPSHOTS

UPDATE FINANCE

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Update Tenant" and displays a form titled "Update Tenant Details". The form contains the following data:

Field	Value
Name	ABHISHEK KUMAR
Tenant Number	4
Apartment Number	1006
Phone	89867665
Type	6BHK
Block	6
Dependent Name	RAMKUWAR SINGH
Dependent Phone Number	91999038

At the bottom of the form is a "Submit" button.

[**Contact Us**](#)

DATA UPDATED SUCCESSFULLY

Data Updated Successfully

Abhishek & Aditya Group

ADD TENTANT ADD DEPENDENT FINANCE Update Details Details Contact Us Apartment

Update Tenant

Update Tenant Details

Name	Enter the Tenant Name
Tenant Number	Enter The Tenant Number
Apartment Number	Enter The Apartment Number
Phone	Enter The Phone Number
Type	Enter The Type
Block	Enter The Block
Dependent Name	Enter The Dependent Name

localhost/phpmyadmin/index.php?route=/sql&server=1&db=apt_mgmt&table=tenant&pos=0

phpMyAdmin

Recent Favorites

- apt_mgmt
- New
- admin
- apartment
- contact_us
- dependent
- finance
- rent
- tenant
- information_schema
- mysql
- performance_schema
- phpmyadmin
- arms
- test

Showing rows 0 - 3 (4 total). Query took 0.0004 seconds.

SELECT * FROM `tenant`

	TNO	TNAME	TPHONE			
<input type="checkbox"/>	Edit	Copy	Delete	1	poonam soni	898765544
<input type="checkbox"/>	Edit	Copy	Delete	2	Ajin Sunesh	2147483647
<input type="checkbox"/>	Edit	Copy	Delete	3	arun kumar n	87676543
<input type="checkbox"/>	Edit	Copy	Delete	4	ABHISHEK KUMAR	898767665

Showing rows 0 - 3 (4 total). Query took 0.0004 seconds.

SELECT * FROM `apartment`

	APTNNO	TNO	TYPE	BLOCK
<input type="checkbox"/>	4	3	3bhk	2
<input checked="" type="checkbox"/>	1001	1	9bhk	9
<input checked="" type="checkbox"/>	1002	2	2BHK	1
<input checked="" type="checkbox"/>	1005	4	6BHK	6

Showing rows 0 - 2 (3 total). Query took 0.0003 seconds.

SELECT * FROM `dependent`

	TNO	DNAME	DPHONE
1	1	Harsh soni	2147483647
2	2	N SUMESH	2147483647
4	4	RAMKUWAR SINGH	919990338

DELETE TENANT

localhost/DBMS/Details.php

Delete tenant

Enter the Tenant Number

Tenant Number
4
Submit

Contact Us

Creativebh13
26.300419,84.187264 ABHISHEK KUMAR, S/O-RAMJUWAR SINGH & BINOD KUMARI VILL-BHAGWAHPUR, POKHARA, NEPAL 26100
4.6 ★★★★★ 19 reviews
View larger map

Directions

Madarsa hotel, Hanuman Temple, Kali mata temple, Durga Temple, M/S Gangotri Group, Ankit Digital Service Center, Khalwa Ashram, Khalwa Village, Durga temple, Shrikant Singh, Smart Coaching Centre, Khalwa naickeri Shali.

Let us know

Name
Enter Your Name

Email
Enter Your Email

Subject
Enter The Subject

DATA DELETED SUCCESSFULLY

localhost/DBMS/Details.php

Data Deleted Successfully

Abhishek & Aditya Group ADDTENANT ADDEPENDENT FINANCE Update Details Details Contact Us Apartment

Tenant Details

Enter the Tenant Number

Tenant Number
Enter The Tenant Number
Submit

Delete tenant

Enter the Tenant Number

Tenant Number
Enter The Tenant Number
Submit

Contact Us

phpMyAdmin

localhost / 127.0.0.1 / apt_mgmt | Apartment Management System | #VIDEO_SONG | #Rakesh M | +

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds)

SELECT * FROM `tenant`

	TNO	TNAME	TPHONE			
<input type="checkbox"/>	Edit	Copy	Delete	1	pooni soni	898765544
<input type="checkbox"/>	Edit	Copy	Delete	2	Ajin Sunesh	2147483647
<input type="checkbox"/>	Edit	Copy	Delete	3	arun kumar n	87676543

Query results operations

Print Copy to clipboard Export Display chart Create view

CONTACT US

Creativebhv13
26-204-B1-A1-17724 ABHISHEK
KUMAR 5/5 RAMKUWAR SINGH &
BINOD KUMARI VILL-BHAGWANPUR,
PO-CHAKWA, Distt-BHEL 841243
4.6 ★★★★★ 19 reviews
View larger map

Directions

Shrikant Singh

Kali Aashthan Narayapur

Creativebhv13

Smart Coaching Centre

Mahal Nadi Chhatri

KAALI AASHTHAN NARAYANPUR

Ansi Enterprises

Balkunhpur

Mas Hadadwa Bhawani Temple

Hindu tempal

Mirzul khas

Pawan Sharma

Shapur Mireuti

Khap Mirzuli

MATHIYA MOD

Tilmapur ukamit

Sharma karpantar

Bala Masjid

Bala kali manda

Tilmapur

Parta

RACHURAD

Mas Kali Temple

Google

Let us know

Name
ADITYA KRISHNAN

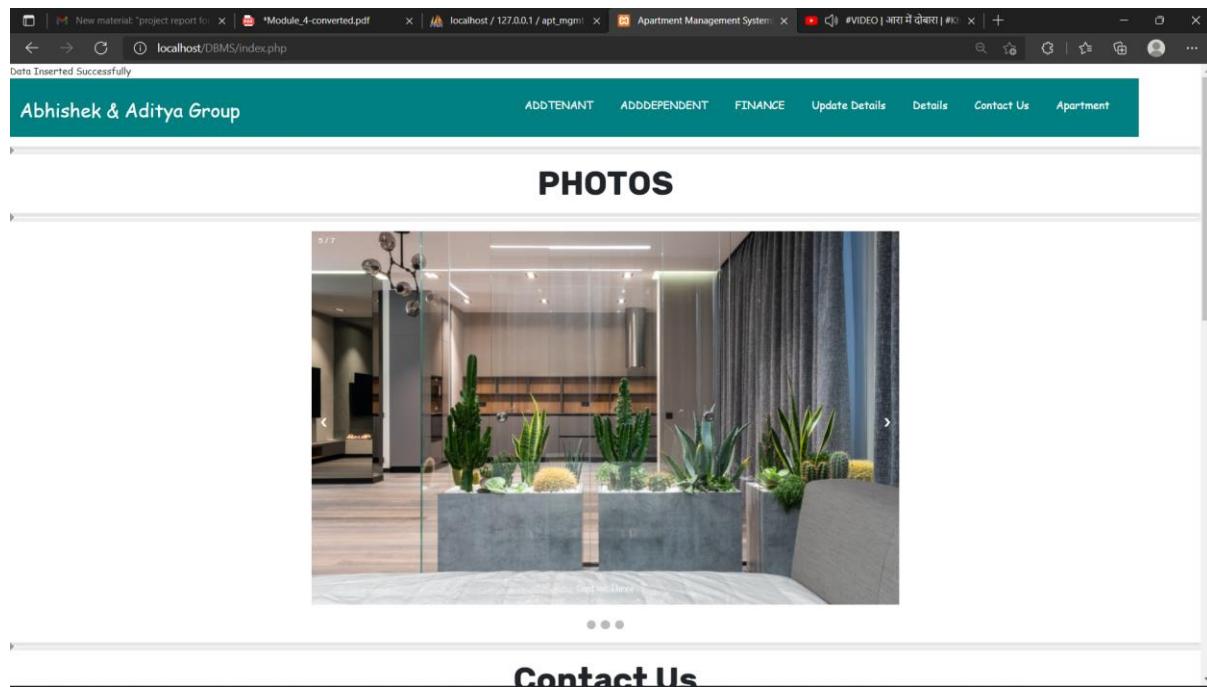
Email
ADITYA123@GMAIL.COM

Subject
REGARDING 2BHK HOUSE

Message
HEY, THIS IS ADITYA I AM LOOKING FOR 2 BHK HOUSE NEAR J BLOCK ,AMAUSI ROAD LUCKNOW

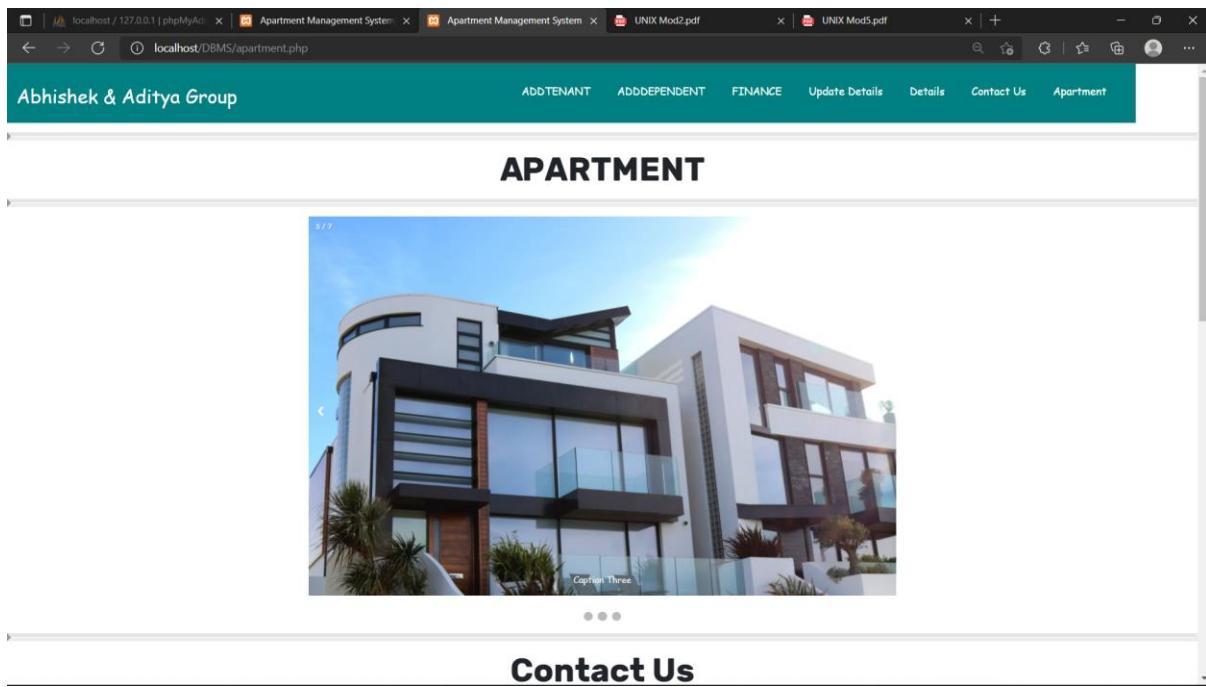
Submit

DATA INSERTED SUCCESSFULLY



The screenshot shows the phpMyAdmin interface connected to a MySQL database named "apt_mgmt". The left sidebar shows the database structure with tables like "New", "apt_mgmt", "admin", "apartment", "contact_us", "dependent", "finance", "rent", and "tenant". The right panel displays the "contact_us" table. A query result is shown: "Showing rows 0 - 2 (total: 2) Query took 0.0007 seconds." The table has columns: Name, Email, Subject, and Message. Two rows are listed:

Name	Email	Subject	Message
ADITYA KRISHNAN	ADITYA123@GMAIL.COM	REGARDING 2BHK HOUSE N...	HEY THIS IS ADITYA I AM LOOKING FOR 2 BHK HOUSE N...
bhupinder singh	BHUP1@GMAIL.COM	I WANT 3 BHK LAND NEAR KONKUN...	PLEASE REACH OUT ME MY WHATSAPP IS WA ME-9186039



CONCLUSION

The use of this software to manage all the aspects of an apartment makes it very easy for owners and property managers to maintain records of their apartments and to quickly look up important details. It saves a significant amount of time when compared to the traditional way of pen and paper, where records tend to get lost or are hard to update and search. Data entry is much faster and less tedious. There can also be multiple copies of the database ensuring that if the main record database runs into issues and is not available, the data is safe elsewhere and the records are not lost.

REFERENCES

PHP MYADMIN

<https://www.phpmyadmin.net/docs/>

MYSQL

<https://dev.mysql.com/doc/>