

TOPIC
AUCTION BIDDING PLATFORM
On
UCS4001
DATABASE MANAGEMENT SYSTEM



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
IILM UNIVERSITY
JANUARY 2026 – MAY 2026

Submitted To:

MR. ATUL KUMAR

Submitted By:

Roll No.: 2410030009

Student Name: ASHISH PRATAP SINGH

Section: 2CSE10

Team: G10

CERTIFICATE

This is to certify that the project entitled **“AUCTION BIDDING PLATFORM”** is a bonafide record of work carried out during the academic year 2025–2026. The work has been completed under supervision and guidance. The student has demonstrated sincere effort, dedication, and a clear understanding of the concepts of Database Management Systems, including Entity-Relationship (ER) modelling, relational schema design, normalization techniques, and implementation of SQL queries.

The project report has been examined and is approved as it satisfies the academic requirements prescribed by the institution.

Project Guide:

Name: Mr. Atul Kumar

Department: Computer Science and Engineering

PROBLEM STATEMENT

In many online marketplaces and bidding platforms, auction-related data such as user details, item listings, bids, auctions, and payment information are often managed in an unstructured or inefficient manner.

Without a proper database system, the following problems arise:

- Difficulty in tracking users and their bids
- Improper management of auction start and end times
- Confusion in identifying highest bids
- Data redundancy and inconsistency
- Poor payment tracking
- Risk of data loss

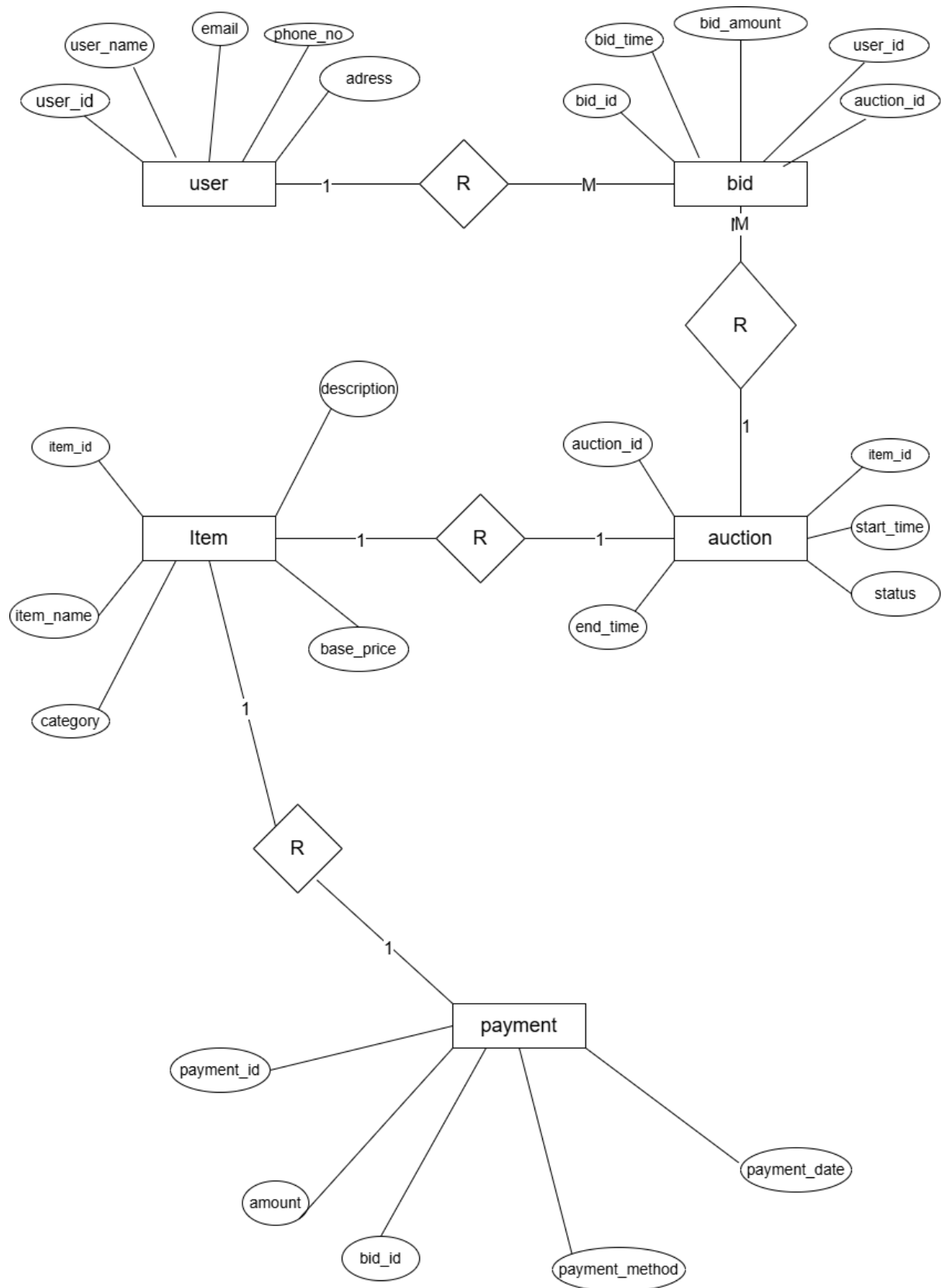
To overcome these issues, there is a need for a structured and centralized database system that efficiently manages auction-related information.

The proposed **Auction Bidding Platform** aims to design and implement a DBMS that organizes data using entities, attributes, and relationships. The system allows systematic storage, retrieval, updating, and management of:

- Users
- Items
- Auctions
- Bids
- Payments

The project focuses on designing an ER model to represent real-world auction relationships, converting it into relational schema, applying normalization techniques, and implementing SQL queries.

AUCTION BIDDING PLATFORM



ENTITY RELATIONSHIP (ER) DIAGRAM

What is an ER Diagram?

An Entity Relationship (ER) Diagram is a graphical representation of the database structure. It shows:

- Entities (objects)
 - Attributes (properties)
 - Relationships (connections between entities)
-

ENTITIES IN THE ER DIAGRAM

All entities in this diagram are **strong entities**.

1. USER

Represents users participating in auctions.

Attributes:

- user_id (Primary Key)
- user_name
- email
- phone_no
- address

Primary Key: **user_id**

2. ITEM

Represents items listed for auction.

Attributes:

- item_id (Primary Key)
- item_name
- category
- description
- base_price

Primary Key: **item_id**

3. AUCTION

Represents auction events for items.

Attributes:

- auction_id (Primary Key)
- item_id (Foreign Key)
- start_time
- end_time
- status

Primary Key: **auction_id**

Foreign Key: **item_id**

4. BID

Represents bids placed by users.

Attributes:

- bid_id (Primary Key)
- bid_amount
- bid_time
- user_id (Foreign Key)
- auction_id (Foreign Key)

Primary Key: **bid_id**

Foreign Keys:

- user_id
- auction_id

5. PAYMENT

Represents payment details after winning an auction.

Attributes:

- payment_id (Primary Key)
- amount
- payment_method
- payment_date
- bid_id (Foreign Key)

Primary Key: **payment_id**

Foreign Key: **bid_id**

What is a Relationship?

A **Relationship** shows how two or more entities are connected.

Relationships are represented by **diamonds**.

Relationships in Diagram

1. USER — BID

Cardinality: 1 : M

One user can place multiple bids.

Each bid belongs to one user.

2. AUCTION — BID

Cardinality: 1 : M

One auction can have multiple bids.

Each bid belongs to one auction.

3. ITEM — AUCTION

Cardinality: 1 : 1

One item is associated with one auction.

Each auction is for one item.

4. ITEM — PAYMENT

Cardinality: 1 : 1

Each item has one payment record after successful bidding.

What is Cardinality?

Cardinality defines the number of instances of one entity that can be associated with another.

Types:

1. One-to-One (1:1)
2. One-to-Many (1:N)
3. Many-to-Many (M:N)

In your diagram:

Relationship	Type
USER-BID	1:N
AUCTION-BID	1:N
ITEM- AUCTION	1:1
ITEM-PAYMENT	1:1

What is Primary Key?

A **Primary Key**:

- Uniquely identifies each record.
- Cannot be NULL.
- Cannot be duplicate.

Example:

- user_id
- item_id
- auction_id
- bid_id
- payment_id

What is Foreign Key?

A **Foreign Key**:

- Links two tables
- References primary key of another table

Example:

- user_id in BID table
- auction_id in BID table
- item_id in AUCTION table
- bid_id in PAYMENT table


```

C:\Users\Ashish Pratap Singh>cd../..

C:\>cd xampp

C:\xampp>cd mysql/bin

C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE auction_db;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| auction_db |
| college |
| company |
| information_schema |
| mysql |
| onlineauctionsystem |
| performance_schema |
| phpmyadmin |
| test |
+-----+
9 rows in set (0.001 sec)

MariaDB [(none)]> USE auction_db;
Database changed
MariaDB [auction_db]> CREATE TABLE user (
->     user_id INT PRIMARY KEY AUTO_INCREMENT,
->     user_name VARCHAR(100) NOT NULL,
->     email VARCHAR(100) UNIQUE,
->     phone_no VARCHAR(15),
->     address VARCHAR(255)
-> );
Query OK, 0 rows affected (0.034 sec)

```

```
MariaDB [auction_db]> CREATE TABLE item (
->     item_id INT PRIMARY KEY AUTO_INCREMENT,
->     item_name VARCHAR(100) NOT NULL,
->     description TEXT,
->     base_price DECIMAL(10,2) NOT NULL,
->     category VARCHAR(50)
-> );
Query OK, 0 rows affected (0.017 sec)

MariaDB [auction_db]> CREATE TABLE auction (
->     auction_id INT PRIMARY KEY AUTO_INCREMENT,
->     item_id INT UNIQUE,
->     start_time DATETIME,
->     end_time DATETIME,
->     status VARCHAR(50),
->     FOREIGN KEY (item_id) REFERENCES item(item_id)
->         ON DELETE CASCADE
-> );
Query OK, 0 rows affected (0.032 sec)

MariaDB [auction_db]> CREATE TABLE bid (
->     bid_id INT PRIMARY KEY AUTO_INCREMENT,
->     bid_amount DECIMAL(10,2) NOT NULL,
->     bid_time DATETIME,
->     user_id INT,
->     auction_id INT,
->     FOREIGN KEY (user_id) REFERENCES user(user_id)
->         ON DELETE CASCADE,
->     FOREIGN KEY (auction_id) REFERENCES auction(auction_id)
->         ON DELETE CASCADE
-> );
Query OK, 0 rows affected (0.035 sec)

MariaDB [auction_db]> CREATE TABLE payment (
->     payment_id INT PRIMARY KEY AUTO_INCREMENT,
->     amount DECIMAL(10,2),
->     payment_date DATETIME,
->     payment_method VARCHAR(50),
->     bid_id INT UNIQUE,
->     FOREIGN KEY (bid_id) REFERENCES bid(bid_id)
->         ON DELETE CASCADE
-> );
Query OK, 0 rows affected (0.035 sec)
```

```
MariaDB [auction_db]> SHOW TABLES;
```

Tables_in_auction_db	
auction	
bid	
item	
payment	
user	

```
5 rows in set (0.001 sec)
```

```
MariaDB [auction_db]> desc user;
```

Field	Type	Null	Key	Default	Extra
user_id	int(11)	NO	PRI	NULL	auto_increment
user_name	varchar(100)	NO		NULL	
email	varchar(100)	YES	UNI	NULL	
phone_no	varchar(15)	YES		NULL	
address	varchar(255)	YES		NULL	

```
5 rows in set (0.028 sec)
```

```
MariaDB [auction_db]> desc item;
```

Field	Type	Null	Key	Default	Extra
item_id	int(11)	NO	PRI	NULL	auto_increment
item_name	varchar(100)	NO		NULL	
description	text	YES		NULL	
base_price	decimal(10,2)	NO		NULL	
category	varchar(50)	YES		NULL	

```
5 rows in set (0.027 sec)
```

```
MariaDB [auction_db]> desc auction;
```

Field	Type	Null	Key	Default	Extra
auction_id	int(11)	NO	PRI	NULL	auto_increment
item_id	int(11)	YES	UNI	NULL	
start_time	datetime	YES		NULL	
end_time	datetime	YES		NULL	
status	varchar(50)	YES		NULL	

```
5 rows in set (0.025 sec)
```

```
MariaDB [auction_db]> desc bid;
```

Field	Type	Null	Key	Default	Extra
bid_id	int(11)	NO	PRI	NULL	auto_increment
bid_amount	decimal(10,2)	NO		NULL	
bid_time	datetime	YES		NULL	
user_id	int(11)	YES	MUL	NULL	
auction_id	int(11)	YES	MUL	NULL	

```
5 rows in set (0.031 sec)
```

```
MariaDB [auction_db]> desc payment;
```

Field	Type	Null	Key	Default	Extra
payment_id	int(11)	NO	PRI	NULL	auto_increment
amount	decimal(10,2)	YES		NULL	
payment_date	datetime	YES		NULL	
payment_method	varchar(50)	YES		NULL	
bid_id	int(11)	YES	UNI	NULL	

```
5 rows in set (0.021 sec)
```

```
MariaDB [auction_db]> INSERT INTO user (user_name, email, phone_no, address)
```

```
-> VALUES
```

```
-> ('Rahul Sharma', 'rahul@gmail.com', '9876543210', 'Delhi'),
```

```
-> ('Ashish', 'ashish@gmail.com', '8081173943', 'Mumbai');
```

```
Query OK, 2 rows affected (0.008 sec)
```

```
Records: 2 Duplicates: 0 Warnings: 0
```

```
MariaDB [FleetManagement]> SELECT v.Vehicle_Number
```

```
-> FROM Vehicle v
```

```
-> LEFT JOIN Trip t ON v.Vehicle_ID = t.Vehicle_ID
```

```
-> WHERE t.Trip_ID IS NULL;
```

```
Empty set (0.003 sec)
```

```
MariaDB [auction_db]> SELECT * FROM user;
```

user_id	user_name	email	phone_no	address
1	Rahul Sharma	rahul@gmail.com	9876543210	Delhi
2	Ashish	ashish@gmail.com	8081173943	Mumbai

```
2 rows in set (0.001 sec)
```

```
MariaDB [auction_db]> INSERT INTO item (item_name, description, base_price, category)
```

```
-> VALUES ('Laptop', 'Dell i5 11th Gen', 40000.00, 'Electronics');
```

```
Query OK, 1 row affected (0.005 sec)
```

```
MariaDB [auction_db]>
```

```
MariaDB [auction_db]> INSERT INTO item (item_name, description, base_price, category)
```

```
-> VALUES ('Mobile', 'Samsung Galaxy', 15000.00, 'Electronics');
```

```
Query OK, 1 row affected (0.006 sec)
```

```
MariaDB [auction_db]> SELECT * FROM item;
```

item_id	item_name	description	base_price	category
1	Laptop	Dell i5 11th Gen	40000.00	Electronics
2	Mobile	Samsung Galaxy	15000.00	Electronics

```
2 rows in set (0.001 sec)
```

```
MariaDB [auction_db]> INSERT INTO auction (item_id, start_time, end_time, status)
```

```
-> VALUES (1, NOW(), DATE_ADD(NOW(), INTERVAL 2 DAY), 'Active');
```

```
Query OK, 1 row affected (0.004 sec)
```

```
MariaDB [auction_db]>
```

```
MariaDB [auction_db]> INSERT INTO auction (item_id, start_time, end_time, status)
```

```
-> VALUES (2, NOW(), DATE_ADD(NOW(), INTERVAL 3 DAY), 'Active');
```

```
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [auction_db]> SELECT * FROM auction;
```

```
+-----+-----+-----+-----+-----+
+
| auction_id | item_id | start_time          | end_time            | status
+-----+-----+-----+-----+-----+
+
|          1 |        1 | 2026-02-23 20:03:29 | 2026-02-25 20:03:29 | Active
|          2 |        2 | 2026-02-23 20:03:30 | 2026-02-26 20:03:30 | Active
+-----+-----+-----+-----+-----+
+
2 rows in set (0.001 sec)
```

```
MariaDB [auction_db]> INSERT INTO bid (bid_amount, bid_time, user_id, auction_id)
-> VALUES (42000.00, NOW(), 1, 1);
Query OK, 1 row affected (0.006 sec)
```

```
MariaDB [auction_db]>
MariaDB [auction_db]> INSERT INTO bid (bid_amount, bid_time, user_id, auction_id)
-> VALUES (43000.00, NOW(), 2, 1);
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [auction_db]>
MariaDB [auction_db]> INSERT INTO bid (bid_amount, bid_time, user_id, auction_id)
-> VALUES (16000.00, NOW(), 1, 2);
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [auction_db]> SELECT * FROM bid;
```

```
+-----+-----+-----+-----+-----+
+
| bid_id | bid_amount | bid_time            | user_id | auction_id |
+-----+-----+-----+-----+-----+
+
|      1 | 42000.00 | 2026-02-23 20:03:51 |      1 |          1 |
|      2 | 43000.00 | 2026-02-23 20:03:51 |      2 |          1 |
|      3 | 16000.00 | 2026-02-23 20:03:53 |      1 |          2 |
+-----+-----+-----+-----+-----+
+
3 rows in set (0.001 sec)
```

```
MariaDB [auction_db]> INSERT INTO payment (amount, payment_date, payment_method, bid_id)
-> VALUES (43000.00, NOW(), 'UPI', 2);
Query OK, 1 row affected (0.006 sec)
```

```

MariaDB [auction_db]>
MariaDB [auction_db]> INSERT INTO payment (amount, payment_date, payment_method, bid_id)
-> VALUES (16000.00, NOW(), 'Credit Card', 3);
Query OK, 1 row affected (0.007 sec)

```

```

MariaDB [auction_db]> SELECT * FROM payment;
+-----+-----+-----+-----+-----+
| payment_id | amount | payment_date | payment_method | bid_id |
+-----+-----+-----+-----+-----+
|          1 | 43000.00 | 2026-02-23 20:04:09 | UPI | 2 |
|          2 | 16000.00 | 2026-02-23 20:04:10 | Credit Card | 3 |
+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

```

```

MariaDB [auction_db]> SELECT u.user_name, b.bid_amount, b.bid_time
-> FROM bid b
-> JOIN user u ON b.user_id = u.user_id;
+-----+-----+-----+
| user_name | bid_amount | bid_time |
+-----+-----+-----+
| Rahul Sharma | 42000.00 | 2026-02-23 20:03:51 |
| Ashish | 43000.00 | 2026-02-23 20:03:51 |
| Rahul Sharma | 16000.00 | 2026-02-23 20:03:53 |
+-----+-----+-----+
3 rows in set (0.001 sec)

```

```

MariaDB [auction_db]> SELECT auction_id, MAX(bid_amount) AS highest_bid
-> FROM bid
-> GROUP BY auction_id;
+-----+-----+
| auction_id | highest_bid |
+-----+-----+
|          1 | 43000.00 |
|          2 | 16000.00 |
+-----+-----+
2 rows in set (0.001 sec)

```

```

MariaDB [auction_db]> SELECT u.user_name, p.amount, p.payment_method
    -> FROM payment p
    -> JOIN bid b ON p.bid_id = b.bid_id
    -> JOIN user u ON b.user_id = u.user_id;

```

user_name	amount	payment_method
Rahul Sharma	16000.00	Credit Card
Ashish	43000.00	UPI

2 rows in set (0.001 sec)

```

MariaDB [auction_db]> SELECT i.item_name, a.status, a.start_time, a.end_time
    -> FROM auction a
    -> JOIN item i ON a.item_id = i.item_id;

```

item_name	status	start_time	end_time
Laptop	Active	2026-02-23 20:03:29	2026-02-25 20:03:29
Mobile	Active	2026-02-23 20:03:30	2026-02-26 20:03:30

2 rows in set (0.001 sec)

```

MariaDB [auction_db]> SELECT * FROM user;

```

user_id	user_name	email	phone_no	address
1	Rahul Sharma	rahul@gmail.com	9876543210	Delhi
2	Ashish	ashish@gmail.com	8081173943	Mumbai

2 rows in set (0.001 sec)

```

MariaDB [auction_db]> SELECT * FROM item;

```

item_id	item_name	description	base_price	category
1	Laptop	Dell i5 11th Gen	40000.00	Electronics
2	Mobile	Samsung Galaxy	15000.00	Electronics

2 rows in set (0.001 sec)

2 rows in set (0.001 sec)

MariaDB [auction_db]> SELECT * FROM auction;

auction_id	item_id	start_time	end_time	status
1	1	2026-02-23 20:03:29	2026-02-25 20:03:29	Active
2	2	2026-02-23 20:03:30	2026-02-26 20:03:30	Active

2 rows in set (0.001 sec)

MariaDB [auction_db]> SELECT * FROM auction;

auction_id	item_id	start_time	end_time	status
1	1	2026-02-23 20:03:29	2026-02-25 20:03:29	Active
2	2	2026-02-23 20:03:30	2026-02-26 20:03:30	Active

2 rows in set (0.001 sec)

MariaDB [auction_db]> SELECT * FROM payment;

payment_id	amount	payment_date	payment_method	bid_id
1	43000.00	2026-02-23 20:04:09	UPI	2
2	16000.00	2026-02-23 20:04:10	Credit Card	3

2 rows in set (0.001 sec)

CONCLUSION

The Auction bidding platform successfully demonstrates the practical implementation of Database Management System concepts using ER modelling and relational schema design.

The system efficiently organizes and manages data related to users, items, auctions, bids, and payments.

Primary keys and foreign keys ensure data integrity and establish proper relationships between tables. The use of one-to-one and one-to-many relationships accurately represents real-world auction scenarios .

Normalization techniques reduce redundancy and maintain consistency. SQL queries enable creation, insertion, updating, deletion, and retrieval of data effectively

Overall, this project provides a clear understanding of ER modelling, relational schema conversion, and SQL implementation in building a structured database system for an Auction bidding platform.