Visvesvaraya Technological University

Jnana Sangama, Belagavi – 590018, Karnataka

A Mini Project Report
on
"VIDEO GAMES MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirement for the DBMS Laboratory with mini project (18CSL58) of V semester

Bachelor of Engineering
In
Computer Science and Engineering

Submitted By
Prajna N (1GA19CS109)
Nayan V Bhandari (1GA19CS095)
Under the Guidance of

Mrs. Reshma S
Assistant Professor,
Dept. of CSE

Mrs. Vanishree M L
Assistant Professor,
Dept. of CSE

GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering
(Accredited by NBA 2019-2022)
Raja Rajeshwari Nagar, Bengaluru – 560 098

2021-2022

GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Raja Rajeshwari Nagar, Bengaluru – 560 098

Certificate

This is to certify that V Semester Mini project entitled **"VIDEO GAMES MANAGEMENT SYSTEM"** is a bona fide work carried out by **PRAJNA N (1GA19CS109), NAYAN V BHANDARI (1GA19CS095)** as a partial fulfillment for the award of Bachelor's Degree in Computer Science and Engineering for DBMS Laboratory with Mini Project [18CSL58] as prescribed by **Visvesvaraya Technological University**, **Belagavi** during the year 2021-2022.

---	---	---
Mrs. Reshma S Assistant Professor, Dept of CSE, GAT, Bengaluru.	Mrs. Vanishree M L Assistant Professor, Dept of CSE, GAT, Bengaluru.	Dr. Bhagyashri R Hanji Professor & Head, Dept of CSE, GAT, Bengaluru.

External Exam

Name of the Examiner	Signature with date
1._____	_____
2._____	_____

ABSTRACT

A video games database management system is the company's way of managing its stocks. It helps in efficient management of resources. It ensures that the logistics are in check and the company has reach to everyone linked to it through its products to ensure there are no discrepancies in the progress of the company. It ensures protection of patents and copyright by storing the crucial details of the products.

This project focuses on the retail stores and the products they hoard in stock or sell it to the customers who walk-in to the store. Our employees have an ID and a password to login to our website and then fill in the details, which would be stored in the company's database for future reference. Then an invoice is printed out to the customer.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned our efforts with success.

We consider ourselves proud, to be part of **Global Academy of Technology** family, the institution which stood by our way in endeavors.

We express our deep and sincere thanks to our Principal **Dr. N. Ranapratap Reddy** for his support.

We are grateful to **Dr. Bhagyashri R Hanji,** Professor and Head, Dept. of Computer Science & Engineering who is source of inspiration and of invaluable help in channelizing my efforts in right direction.

We wish to thank our internal guide **Mrs. Reshma S** and **Mrs. Vanishree M L**, Assistant Professor, Dept of CSE for guiding and correcting various documents with attention and care. She has taken lot of pain to go through the document and make necessary corrections as and when needed.

We would like to thank the faculty members and supporting staff of the Department of CSE, GAT for providing all the support for completing the Project work.

Finally, we are grateful to our parents and friends for their unconditional support and help during our Project work.

NAYAN V BHANDARI (1GA19CS095)

PRAJNA N (1GA19CS109)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

Structured Query Language is a domain specific language used in programming and designed for managing data held in a relational database management system. It consists of many types of statements which are informally classed as sub languages, a few of which are:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Query language (DQL)
- Data Control Language (DCL)

INTRODUCING DATABASES

To understand SQL, it's important to have a basic understanding of how databases work. If you're comfortable with terms like table, relation and query, feel free to flow right ahead! If not, you may wish to read the article Database Fundamentals before moving on.

A database is an organized collection of data, generally stored and accessed electronically from a computer system. It supports the storage and manipulation of data. In other words, databases are used by an organization as a method of storing, managing and retrieving information.

There are many advantages of databases:

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity and independence from application programs
- Improved data access to users through the use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs

INTRODUCTION TO QUERY

A query can be either a request for data results from your database or for action on the data, or for both. A query can give you an answer to a simple question, perform calculations, combine data from different tables, add, delete, or change data from a database.

Let's take an example suppose there is a table student which has student_id and student_name as attributes, if we want to find a student whose id is 56, then we can simply use a query as follows:

Select * from student where student_id = 56;

THE RANGE OF SQL STATEMENTS

SQL provides a wide range of statements, of which SELECT is just one.

Here are some examples of other common SQL statements:

SQL INSERT and SQL DELETE: Inserts or deletes a record from a table.

SQL UPDATE: Modifies records in a table.

SQL CREATE and SQL DROP: Creates or deletes a table.

In addition to these SQL statements, you can use SQL clauses, among them the WHERE clause used in the previous examples. These clauses serve to refine the type of data to act on. In addition to the WHERE clause, here are other commonly-used clauses:

AND or OR

Combine multiple conditions to refine a SQL statement

LIKE: Compares a value to similar values using a wildcard.

ORDER BY: Sorts data in ascending or descending order.

If you are interested in further exploring SQL, SQL Fundamentals is a multi-part tutorial that explores the components and aspects of SQL in more detail.

INTRODUCTION TO FRONT END SOFTWARE

PHP

BACKGROUND

PHP is a server-side scripting system

- PHP stands for "PHP: Hypertext Pre-processor"
- Syntax based on Perl, Java, and C
- Very good for creating dynamic content
- Powerful, but somewhat risky!
- If you want to focus on one system for dynamic content, this is a good one to choose

HISTORY

- Started as a Perl hack in 1994 by RasmusLerdorf (to handle his resume), developed to PHP/FI 2.0
- By 1997 up to PHP 3.0 with a new parser engine by ZeevSuraski and AndiGutmans
- Version 8.1.2 is current version
- PHP is one of the premier examples of what an open-source project can be PHP Scripts
- Typically file ends in. php--this is set by the web server configuration
- Separated in files with the tag
- PHP commands can make up an entire file, or can be contained in html--this is a choice.
- Program lines end in ";" or you get an error
- Server recognizes embedded script and executes

PROJECT REPORT OUTLINE

The report is arranged in the following way:

Chapter 1: Introduction to SQL about its database, sql query state, AND or OR and range if sql statements and introduction to front end software PHP

Chapter 2: Requirement specification of hardware and software

Chapter 3: Objective of the Project, design of project and developing

Chapter 4: Implementation of ER diagram and it's description

Chapter 5: Front End Design, connecting to database using PHP, Front end code of the Project

Chapter 6: Testing of project by different cases, it's process and testing objectives
Chapter 7: Outcome of the Project

Conclusions

References

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Operating System : Windows7 or higher

Database : MariaDB

Tools : WampServer3.2 or higher

2.2 HARDWARE SPECIFICATION

Processor : Any Processor above 500 MHz

AM : 4.00GB

Hard Disk : 1TB

Compact Disk : 700Mb

Input device : Keyboard

Output device : Laptop Display Screen

CHAPTER 3

OBJECTIVE OF THE PROJECT

The main objective of this application is to:

1. Have a clear picture about the products in stock.
2. Know the whereabouts of a purchase.
3. Be well informed about the customer details.
4. Print out the invoices and hand it over to customers.
5. Store all the crucial data in the company's database.

CHAPTER 4

IMPLEMENTATION

4.1 ER DIAGRAM

The following ER Diagram shows the entity relationship the entity relationships of Videogames Management System

The entities used are:

- GAME
- DEVELOPER
- DEVICECOMPATIBILITY
- UPDATES
- CUSTOMER
- INVOICE
- ADMINLOGIN

The attributes of Entity GAME are ID, Name, Version, Creation_date, MRP, where ID is the primary key

The attributes of Entity DEVELOPER are D_ID, Name, Location, where D_ID is the primary key

he attributes of Entity DEVICECOMPATIBILITY are RAM, Memory, Processor

The attributes of Entity UPDATES are new_version, existing_version, new_version_date

The attributes of Entity CUSTOMER are C_ID, Fname, Lname, Phno, Address, where C_ID is the primary key

The attributes of Entity INVOICE are Billing_id, selling_price, billing_date, where Billing_id is the primary key

The attributes of Entity ADMINLOGIN are Login_id, password, where login_id is the primary key

The relationships are as follows:

1) **DEVELOPS**: between the entities are DEVELOPER and GAME i.e, DEVELOPER DEVELOPS GAME with cardinality 1:N, partial participation from DEVELOPER and total participation from GAME
2) **REQUIRES MINIMUM** : between the entities are DEVICECOMPATIBILITY and GAME i.e, GAME REQUIRES MINIMUM DEVICECOMPATIBILITY with cardinality 1:1, total participation from DEVICECOMPATIBILITY and total participation from GAME
3) **HAS**: between the entities are UPDATES and GAME i.e, GAME HAS UPDATES with cardinality 1:N, partial participation from GAME and total participation from UPDATES
4) **BUYS**: between the entities are CUSTOMER and GAME i.e, CUSTOMER BUYS GAME with cardinality 1:N, partial participation from GAME and total participation from CUSTOMER
5) **GETS**: between the entities are CUSTOMER and INVOICE i.e, CUSTOMER GETS INVOICE with cardinality 1:1, total participation from invoice and total participation from CUSTOMER

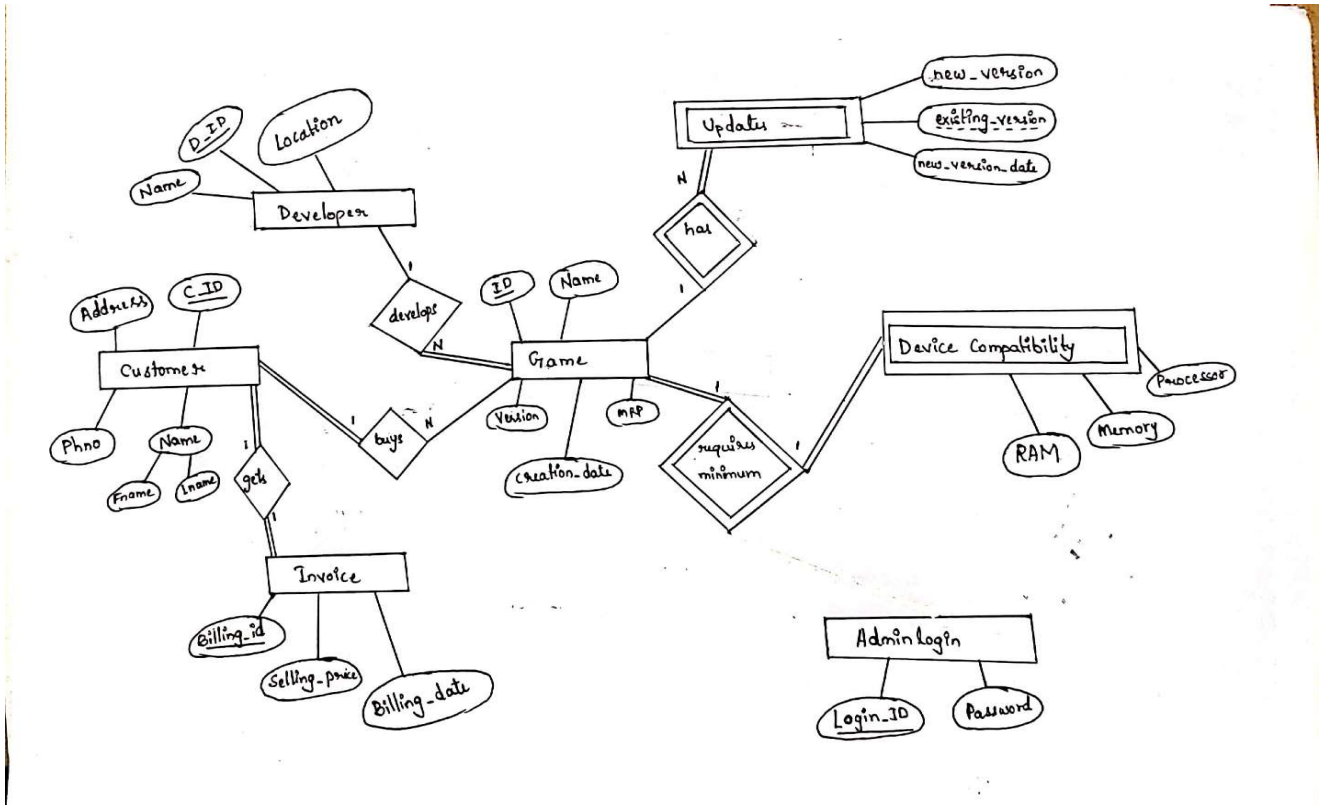
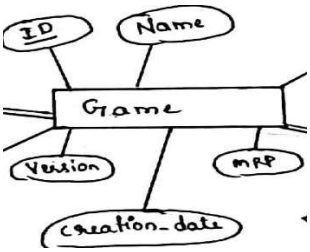


Fig. 4.1: ER Diagram of VIDEOGAMES MANAGEMENT SYSTEM

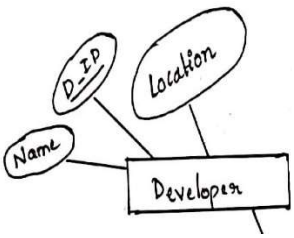
4.2 MAPPING OF ER DIAGRAM TO SCHEMA DIAGRAM

Mapping: The conceptual/internal mapping defines the correspondence between the conceptual view and the store database. It specifies how conceptual record and fields are represented at the internal level. There could be one mapping between conceptual and internal levels.

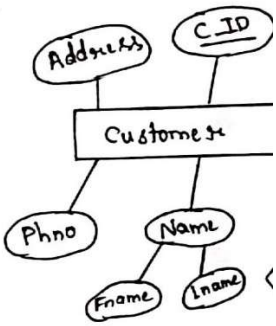
STEP 1: Mapping of Regular (Strong) Entity types



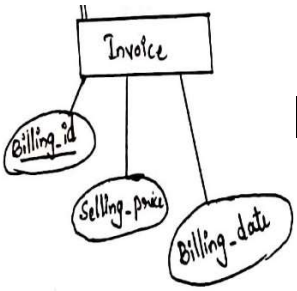
ID	Name	Version	Creation_date	MRP



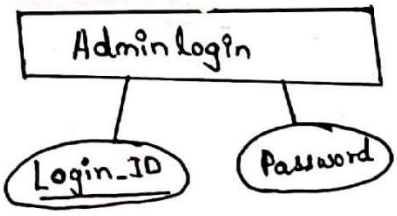
D_ID	Name	Location



C_ID	Fname	Lname	Phno	Address



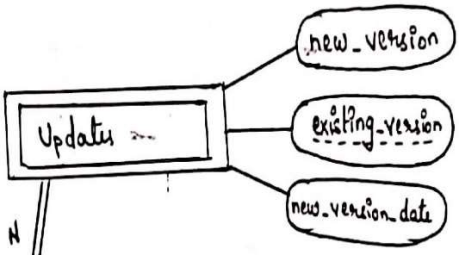
Billing_id	Selling_price	Billing_date



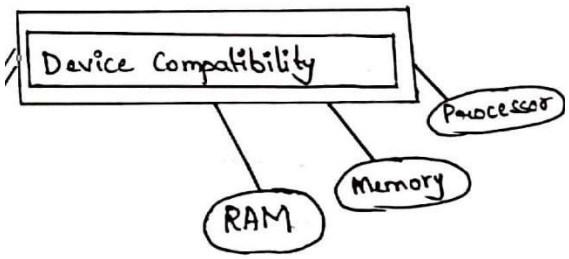
Login_id	Password

Fig. 4.2.1: Step 1 Mapping of ER Diagram to Schema Diagram

Step 2: Mapping of weak entities



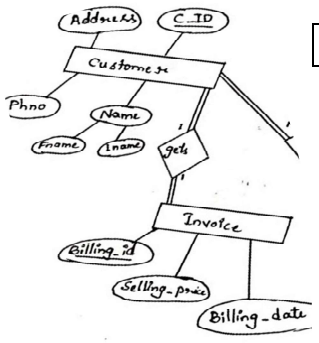
Game_id	New_version	Existing_version	New_version_date



Game_id	RAM	Memory	Processor

Fig. 4.2.2: Step 2 Mapping of ER Diagram to Schema Diagram

Step 3: Mapping of Binary 1:1 relationship

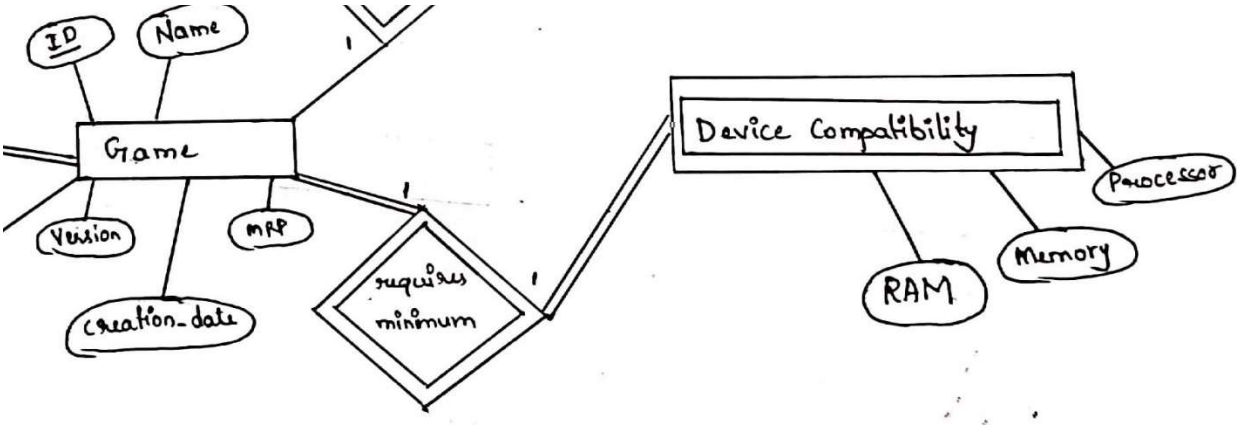


Customer:

C_ID	Fname	Lname	Phno	Address

Invoice:

Billing_id	Selling_price	Billing_date	C_ID



Game:

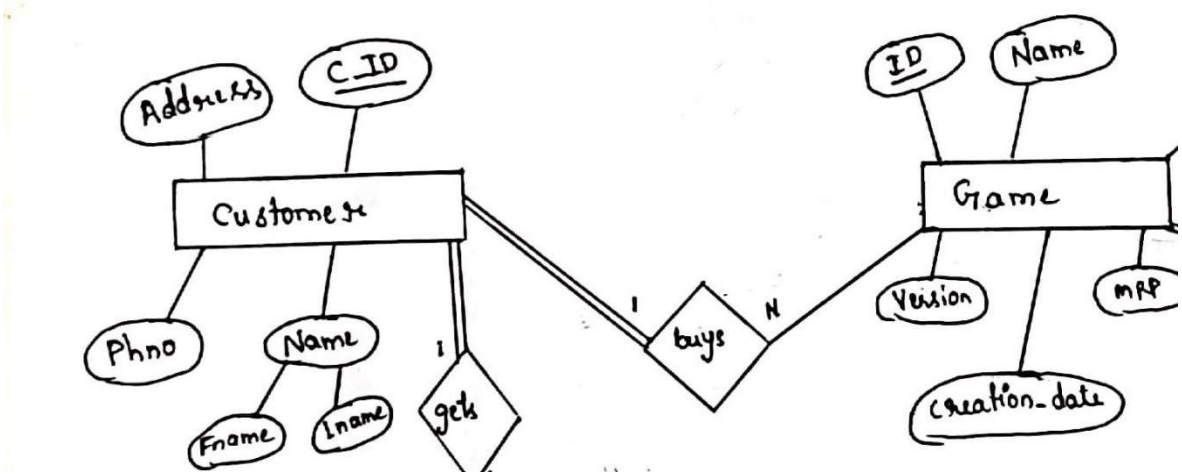
ID	Name	Version	Creation_date	MRP

DeviceCompatibility:

Game_id	RAM	Memory	Processor

Fig 4.2.3: Step 3 Mapping of ER Diagram to Schema Diagram

Step 4: Mapping of 1:N relationship

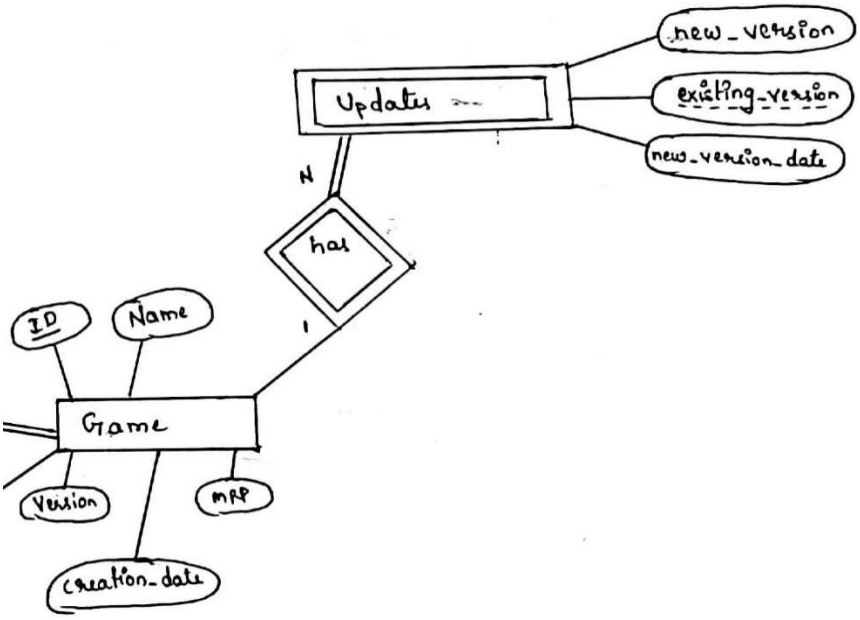


Game:

ID	Name	Version	Creation_date	MRP

Customer:

C_ID	Fname	Lname	Phno	Address

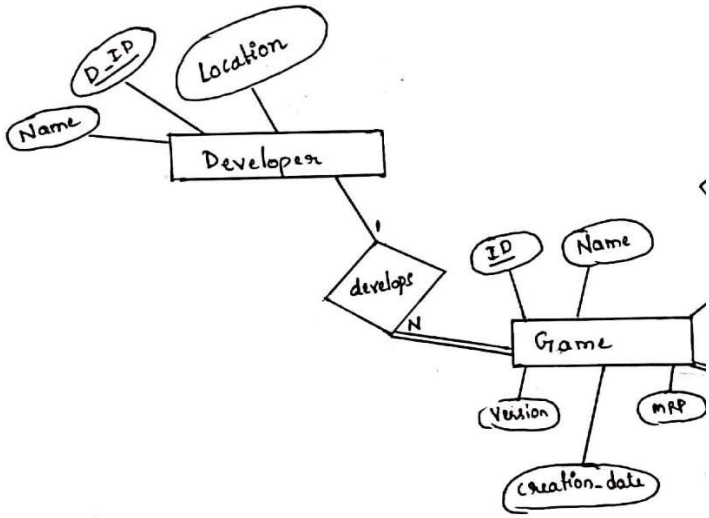


Game:

ID	Name	Version	Creation_date	MRP

Updates:

Game_id	New_version	Existing_version	New_version_date



Game:

ID	Name	Version	Creation_date	MRP

Developer:

D_ID	Name	Location	Game_id

Fig. 4.2.4: Step 4 Mapping of ER Diagram to Schema Diagram

Step 5: Mapping of M:N relationship

No M:N relationship types

Step 6: Mapping of Multivalued attributes

No multivalued attributes

Step 5: Mapping of N-ary relationship

No N-ary relationship types

4.3 Mapping of ER Schema to relations

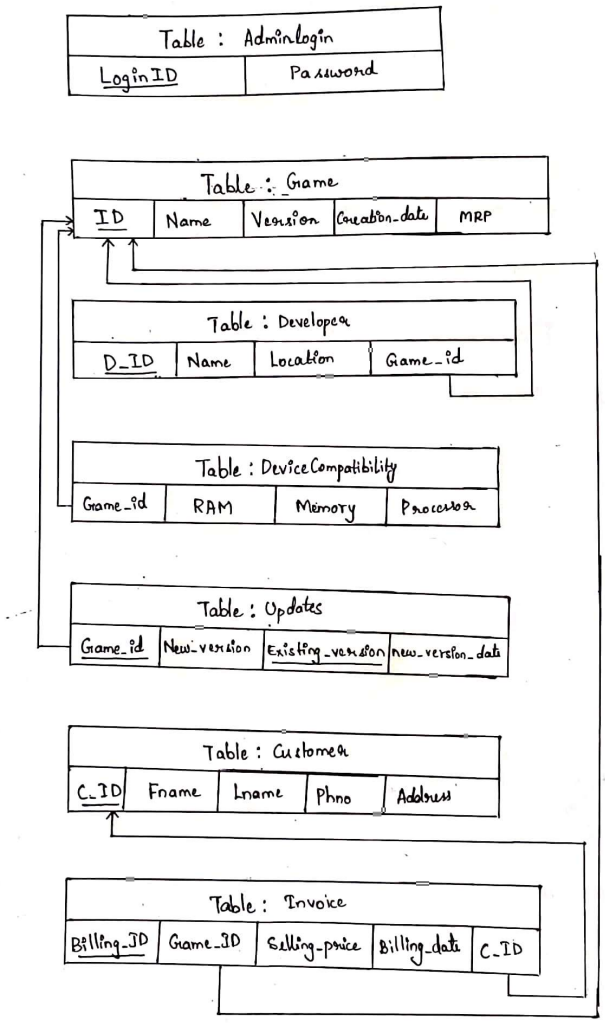


Fig 4.3: Schema Diagram of Videogames Management System

4.4 Creation of Tables

CREATE TABLE GAME

(ID varchar(5) PRIMARY KEY,

Name varchar(20) NOT NULL,

Version varchar(5),

Creation_date date,

MRP dec(7,2));

Table 4.4.1: Description of Table GAME

#	Name	Type	Collation	Attributes	Null	Default
1	**ID**	varchar(5)	hp8_english_ci		No	*None*
2	**Name**	varchar(20)	hp8_english_ci		No	*None*
3	**Version**	varchar(5)	hp8_english_ci		Yes	*NULL*
4	**Creation_date**	date			Yes	*NULL*
5	**MRP**	decimal(7,2)			Yes	*NULL*

CREATE TABLE DEVELOPER

(D_ID varchar(5) PRIMARY KEY,

Name varchar(20) NOT NULL,

Location varchar(20) NOT NULL,

Game_id varchar(5) NOT NULL,

FOREIGN KEY (Game_id) REFERENCES GAME(ID)ON DELETE CASCADE);

Table 4.4.2: Description of Table DEVELOPER

#	Name	Type	Collation	Attributes	Null	Default
1	**D_ID**	varchar(5)	hp8_english_ci		No	*None*
2	**Name**	varchar(20)	hp8_english_ci		No	*None*
3	**Location**	varchar(20)	hp8_english_ci		No	*None*
4	**game_id**	varchar(5)	hp8_english_ci		No	*None*

CREATE TABLE DEVICECOMPATIBILITY

(Game_id varchar(5),

RAM int(3) CHECK (RAM>=8),

Memory varchar(10),

Processor varchar(10),

FOREIGN KEY (Game_id) REFERENCES GAME(ID)ON DELETE CASCADE);

Table 4.4.3: Description of Table DEVICECOMPATIBILITY

#	Name	Type	Collation	Attributes	Null	Defau
1	**Game_id**	varchar(5)	hp8_english_ci		Yes	*NULL*
2	**RAM**	int(3)			Yes	*NULL*
3	**Memory**	varchar(20)	hp8_english_ci		Yes	*NULL*
4	**Processor**	varchar(20)	hp8_english_ci		Yes	*NULL*

CREATE TABLE UPDATES

(Game_id varchar(5),

New_version varchar(5),

Existing_version varchar(5),

New_version_date date,

FOREIGN KEY (Game_id) REFERENCES GAME(ID)ON DELETE CASCADE,

PRIMARY KEY(Game_id,Existing_version));

Table 4.4.4: Description of Table UPDATES

#	Name	Type	Collation	Attributes	Null	Default
1	**game_id**	varchar(5)	hp8_english_ci		No	*None*
2	**new_version**	varchar(5)	hp8_english_ci		Yes	*NULL*
3	**existing_version**	varchar(5)	hp8_english_ci		No	*None*
4	**new_version_date**	date			Yes	*NULL*

CREATE TABLE CUSTOMER

(C_ID varchar(5) PRIMARY KEY,

Fname varchar(20) NOT NULL,

Lname varchar(20),

Phno number(10) NOT NULL,

Address varchar(50) NOT NULL);

Table 4.4.5: Description of Table CUSTOMER

#	Name	Type	Collation	Attributes	Null	Default
1	**C_ID**	varchar(5)	hp8_english_ci		No	*None*
2	**Fname**	varchar(20)	hp8_english_ci		No	*None*
3	**Lname**	varchar(20)	hp8_english_ci		Yes	*NULL*
4	**Phno**	bigint(10)			No	*None*
5	**Address**	varchar(50)	hp8_english_ci		No	*None*

CREATE TABLE INVOICE

(Billing_id varchar(5) PRIMARY KEY,

Game_id varchar(5) NOT NULL,

Selling_price number(7,2),

Billing_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

C_ID varchar(5),

FOREIGN KEY (C_ID) REFERENCES CUSTOMER(C_ID)ON DELETE CASCADE,

FOREIGN KEY (Game_id) REFERENCES GAME(ID)ON DELETE CASCADE);

Table 4.4.6: Description of Table INVOICE

#	Name	Type	Collation	Attributes	Null	Default
1	**billing_id**	varchar(5)	hp8_english_ci		No	*None*
2	**game_id**	varchar(5)	hp8_english_ci		No	*None*
3	**Selling_price**	decimal(7,2)			Yes	*NULL*
4	**Billing_date**	datetime			Yes	current_timestamp()
5	**C_ID**	varchar(5)	hp8_english_ci		Yes	*NULL*

CREATE TABLE ADMINLOGIN

(Login_id varchar(20) Primary Key,

Password varchar(20));

Table 4.4.7: Description of Table LOGINADMIN

#	Name	Type	Collation	Attributes	Null	Default
1	LoginID	varchar(20)	hp8_english_ci		No	*None*
2	Password	varchar(20)	hp8_english_ci		Yes	*NULL*

4.5 INSERTION OF TUPLES

TABLE GAME:

Insert into Game values(‘VG01’, ‘League of legends’,1, ‘2009-10-27’,1500.0);

Insert into Game values(‘VG02’,’Valorant’,1,’2020-06-02’,3000.0);

Insert into Game values(‘VG03’,’Watch Dogs’,1,’2014-05-27’,1000.0);

Insert into Game values(‘VG04’,’CALL OF DUTY’,1,’2003-10-29’,2850.0);

Insert into Game values(‘VG05’,’ASSASSINS CREED’,1,’2007-09-13’,1200.0);

Table 4.5.1: Insertion in Table GAME

ID	Name	Version	Creation_date	MRP
VG01	League of legends	1	2009-10-27	1500.00
VG02	Valorant	1	2020-06-02	3000.00
VG03	Watch Dogs	1	2014-05-27	1000.00
VG04	CALL OF DUTY	1	2003-10-29	2850.00
VG05	ASSASSINS CREED	1	2007-09-13	1200.00

TABLE DEVELOPER:

Insert into Developer values('D1','Riot Games’, 'Los Angeles','VG01');

Insert into Developer values('D2','Riot Games’, 'Los Angeles','VG02');

Insert into Developer values('D3','Ubisoft Montreal','Montreuil','VG03');

Insert into Developer values('D4','Activision','California','VG04');

Insert into Developer values('D5','Ubisoft Montreal','Montreuil','VG05');

Table 4.5.2: Insertion in Table DEVELOPER

D_ID	Name	Location	game_id
D1	Riot Games	Los Angeles	VG01
D2	Riot Games	Los Angeles	VG02
D3	Ubisoft Montreal	Montreuil	VG03
D4	Activision	California	VG04
D5	Ubisoft Montreal	Montreuil	VG05

TABLE DEVICECOMPATIBILITY:

Insert into DeviceCompatibility values("VG01", 8, "1 TB HDD", "AMD Ryzen 5");

Insert into DeviceCompatibility values("VG02", 8, "1 TB HDD", "Intel core i5-12600K");

Insert into DeviceCompatibility values("VG03", 8, "1 TB HDD","AMD Ryzen 7");

Insert into DeviceCompatibility values("VG04", 8, "256 GB SSD and 1 TB HDD", "Intel core i7-9700K");

Insert into DeviceCompatibility values("VG05", 8,"1 TB HDD", "AMD Ryzen 5");

Table 4.5.3: Insertion in Table DEVICECOMPATIBILITY

Game_id	RAM	Memory	Processor
VG01	8	1 TB HDD	AMD Ryzen 5
VG02	8	1 TB HDD	Intel core i5-12600K
VG03	8	1 TB HDD	AMD Ryzen 7
VG04	8	256 GB SSD and 1 TB HDD	Intel core i7-9700K
VG05	8	1 TB HDD	AMD Ryzen 5

TABLE LOGINADMIN:
Insert into Loginadmin values('vgindia01', 'justemployees')

Table 4.5.4: Insertion in Table LOGINADMIN

LoginID	Password
vgindia01	justemployees

4.6 Creation of Triggers

Before customer table:

BEGIN

SET NEW.Fname = UPPER(NEW.Fname);

SET NEW.Lname = UPPER(NEW.Lname);

END

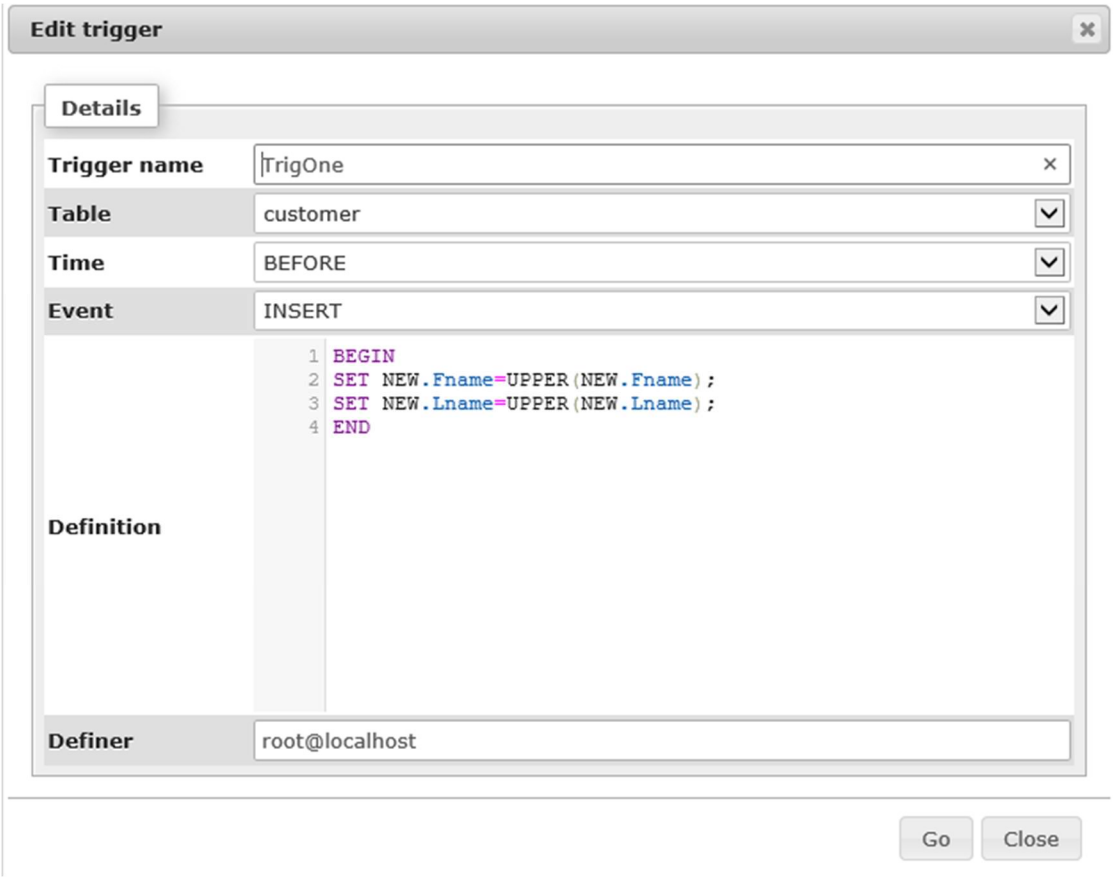


Fig. 4.6: Creation of Triggers

CHAPTER 5

FRONT END DESIGN

5.1 CONNECTIVITY TO DATABASE

PHP CODE FOR MAKING CONNECTION WITH THE DATABASE

```
<?php

$conn = "";

try {
	$servername = "localhost:3306";
	$dbname = "videogame";
	$username = "root";
	$password = "";

	$conn = new PDO(
		"mysql:host=$servername; dbname=videogame",
		$username, $password
	);

$conn->setAttribute(PDO::ATTR_ERRMODE,
						PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e) {
	echo "Connection failed: " . $e->getMessage();
}

?>
```

5.2 FRONT END CODE

PHP CODE FOR HOME PAGE:

```
<?php
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Home</title>
<link rel="stylesheet" href="css/forcontact.css" type="text/css">
<span>
<font face="cinzel" size="5" font-color="black">
<h2>
<h2 align ="right"style="color:#000000;font-size:9px;" >
<body link="black" alink="black" vlink="black">
<a href="home2.php">HOME</a>      
<a href="LoginAdmin.php">LOGIN</a>    
</font>
</span>
</h2></h2>
</body></body>
<h1>
<span>
<h1 align="left" style="color:#000066;font-size:80px">
ANASH
   
<img src="http://localhost//videogames dbms//images//companylogo.jpg"
height=75 width=105 >
</h1>
<img src="http://localhost//videogames dbms//images//homeimg.jpg" height=710
width=1899>
<br>
<div id="footer">
<p align ="center" style="color:#000000">
                    © 2021 by Product Sales Management System. All Rights
Reserved
</p>
</div>
</html>
```

PHP CODE FOR LOGIN PAGE:

```
<?php
include_once('connection.php');
?>
<!DOCTYPE html>
<html lang="en">

<head>
	<meta charset="UTF-8">
	<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<meta http-equiv="X-UA-Compatible" content="ie=edge">
	<link rel="stylesheet" href="css\LoginPage.css">
	<title>Login Page</title>
</head>

<body>
	<form action="validate.php" method="post">
		<div class="login-box">
			<h1>Login</h1>

			<div class="textbox">
				<i class="fa fa-user" aria-hidden="true"></i>
				<input type="text" placeholder="Login ID"
					name="LoginID" value="">
			</div>

			<div class="textbox">
				<i class="fa fa-lock" aria-hidden="true"></i>
				<input type="password" placeholder="Password"
					name="Password" value="">
			</div>

			<input class="button" type="submit"
				name="login" value="Sign In">
			</div>
	</form>
</body>

</html>
```

PHP CODE FOR VALIDATE PAGE:

```
<?php

include_once('connection.php');

function test_input($data) {

        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
}

if ($_SERVER["REQUEST_METHOD"]== "POST") {

        $LoginID = test_input($_POST["LoginID"]);
        $Password = test_input($_POST["Password"]);
        $stmt = $conn->prepare("SELECT * FROM adminlogin");
        $stmt->execute();
        $users = $stmt->fetchAll();

        foreach($users as $user) {

                if(($user['LoginID'] == $LoginID) &&
                        ($user['Password'] == $Password)) {
                                header("Location: contact.php");
                }
                else {

                        echo '<script>alert("Invalid Credentials")</script>';
                        header( "refresh:1; url=LoginAdmin.php" );


                }
        }

}

?>
```

PHP CODE FOR CUSTOMER DETAILS PAGE:

```
<?php
session_start();
include_once('connection.php');
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<meta charset="utf-8">

<link rel="stylesheet" href="css/style.css" type="text/css">
<style>
input[type=button],
     input[type=submit] {
       background-color: #000066;
       border: none;
       color: #fff;
       padding: 15px 30px;
       text-decoration: none;
       margin: 4px 2px;
       cursor: pointer;
            font-size:25px;

     }
</style>
</head>

<body>
<div class="container py-4">
<div class="row" style ="margin-left:150px:">
<div class ="col-md-8">
?php

if(isset($_SESSION['success'])&&$_SESSION['success']!='')
{
            echo '<div class="alert alert-success text-center font-weight-bold"
role="alert">'.$_SESSION['success'].'</div>';
            unset($_SESSION['success']);
}
if(isset($_SESSION['failure'])&&$_SESSION['failure']!='')
{
```

```
        echo '<div class="alert alert-danger text-center font-weight-bold"
role="alert">'.$_SESSION['failure'].'</div>';
        unset($_SESSION['success']);
}

?>
<div align="center" >
<div class="card">
<h1 class="text-primary font-weight-bold text-center">Database Insertion</h1>
<div class = "card-body">
<form action="userinfo.php" method="POST">
<div class="form-group">
<label class="font-wight-bold text-success">Customer ID</label>
<input type="text" name="C_ID" class="form-control"><br><br><br><br>
</div>
<div class="form-group">
<label class="font-wight-bold text-success">First Name</label>
<input type="text" name="Fname" class="form-control"><br><br><br><br>
</div>
<div class="form-group">
<label class="font-wight-bold text-success">Last Name</label>
<input type="text" name="Lname" class="form-control"><br><br><br><br>
</div>
<div class="form-group">
<label class="font-wight-bold text-success">Phone Number</label>
<input type="text" name="Phno" class="form-control"><br><br><br><br>
</div><div class="form-group">
<label class="font-wight-bold text-success">Address</label>
<input type="text" name="Address" class="form-control"><br><br><br><br>
</div>
<div class="text-center">
<input class="button" type="submit" name="submit" value="Submit">
</div>
</div>
</form>
</form>
</body>
</html>
```

PHP CODE FOR VALIDATING CUSTOMER DETAILS:

```
<?php
session_start();
$conn=mysqli_connect('localhost','root','','videogame');

if($conn)
{
	echo "connected to database";
}
else
{
	echo "Not connected to database";
}
if(isset($_POST['submit']))
{
	$C_ID=$_POST['C_ID'];
	$Fname=$_POST['Fname'];
	$Lname=$_POST['Lname'];
	$Phno=$_POST['Phno'];
	$Address=$_POST['Address'];
}

$query="INSERT INTO Customer(C_ID,Fname,Lname,Phno,Address) VALUES
('$C_ID','$Fname','$Lname','$Phno','$Address')";
$result=mysqli_query($conn, $query);

if($result)
{
		$_SESSION['success']="";
		header('Location:contact2.php');
	}
	else
	{
		$_SESSION['failure']="";
			echo'<script>alert("Invalid information, please enter
again")</script>';
			header( "refresh:1; url=contact.php" );

	}

?>
```

PHP CODE FOR CUSTOMER PURCHASE DETAILS:

```
<?php
session_start();
include_once('connection.php');
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="css/style.css" type="text/css">
<style>
input[type=button],
     input[type=submit] {
       background-color: #000066;
       border: none;
       color: #fff;
       padding: 15px 30px;
       text-decoration: none;
       margin: 4px 2px;
       cursor: pointer;
            font-size:25px;
     }
</style></head>
<body>
<div class="container py-4">
<div class="row" style ="margin-left:150px:">
<div class ="col-md-8">
<?php

if(isset($_SESSION['success'])&&$_SESSION['success']!='')
{
            echo '<div class="alert alert-success text-center font-weight-bold"
role="alert">'.$_SESSION['success'].'</div>';
            unset($_SESSION['success']);
}
if(isset($_SESSION['failure'])&&$_SESSION['failure']!='')
{
            echo '<div class="alert alert-danger text-center font-weight-bold"
role="alert">'.$_SESSION['failure'].'</div>';
            unset($_SESSION['success']);
}
```

```
?>
<div align="center" >
<div class="card">
<h1 class="text-primary font-weight-bold text-center">Database Insertion</h1>
            <br><br>

                        <div class = "card-body">

                <form action="userinfo2.php" method="POST">

                                <div class="form-group">
                                <label class="font-wight-bold text-success">Billing
ID</label>
                                        <input type="text" name="billing_id"
class="form-control"><br><br><br><br>
                                </div>

                                <div class="form-group">
                                <label class="font-wight-bold text-success">Game
ID</label>
                                        <input type="text" name="game_id" class="form-
control"><br><br><br><br>
                                </div>

                                <div class="form-group">
                                <label class="font-wight-bold text-success">Selling
Price</label>
                                        <input type="text" name="Selling_price"
class="form-control"><br><br><br><br>
                                </div>
                                <!--
                                <div class="form-group">
                                <label class="font-wight-bold text-success">Customer
ID</label>
                                <input type="text" name="C_ID" class="form-
control"><br><br><br><br>
                                </div>
                                  <div class="text-center">
                                <input class="button" type="submit" name="submit"
value="Submit">
                                </div></div></form></form>
</body>
</html>
```

PHP CODE FOR VALIDATING THE CUSTOMER PURCHASE DETAILS:

```
<?php
session_start();
$conn=mysqli_connect('localhost','root','','videogame');

if($conn)
{
        echo "connected to database";
}
else
{
        echo "Not connected to database";
}
if(isset($_POST['submit']))
{

        $billing_id=$_POST['billing_id'];
        $game_id=$_POST['game_id'];
        $Selling_price=$_POST['Selling_price'];
        $_SESSION['c_id'] = "$C_ID";

}

$query1="INSERT INTO invoice(billing_id,game_id,Selling_price,c_id)
VALUES ('$billing_id','$game_id','$Selling_price','$c_id')";
$result=mysqli_query($conn, $query1);
if($result)
{

                $_SESSION['success']="";
                header('Location:invoice.php');
        }
        else
        {
                $_SESSION['failure']="";
                        echo'<script>alert("Invalid information, please enter
again")</script>';
                        header( "refresh:1; url=contact2.php" );
        }

?>
```

PHP CODE FOR GENERATING INVOICE:

```
<!DOCTYPE html>
<html>
<head>
  <title>Invoice</title>
</head>
<body>
<div align="center">
 <div id="content"
              style="text-align: center;
              vertical-align: middle;
              border-color: #000000;
              border-width: 4px;
              border-style: solid;
              margin-top: 1%;
              margin-bottom: 2%;
                     margin-left: 35%;
                     margin-right: 40%;">
 <h2>Invoice</h2>
<div align="center">
<table border="2">
<tbody>
  <tr>
    <td>Customer_ID</td>
    <td>First Name</td>
    <td>Last Name</td>
    <td>Phone Number</td>
    <td>Address</td>
  </tr>
<?php
include "dbconn.php"; // Using database connection file here
$sql = "SELECT * FROM customer ORDER BY C_ID DESC LIMIT 1";
$records = mysqli_query($db,$sql) or die( mysqli_error($db));
while($data = mysqli_fetch_array($records))
{
?>
  <tr>
    <td><?php echo $data['C_ID']; ?></td>
    <td><?php echo $data['Fname']; ?></td>
    <td><?php echo $data['Lname']; ?></td>
    <td><?php echo $data['Phno']; ?></td>
```

```
   <td><?php echo $data['Address']; ?></td>
  </tr>
<?php
}
?>
</tbody>
</table>
<table border="2">
<tbody>
  <tr>
    <td>Billing ID</td>
    <td>Game ID</td>
    <td>Selling Price</td>
    <td>Billing Date & Time</td>
  </tr><br><br>
<?php
include "dbconn.php"; // Using database connection file here
$sql = "SELECT * FROM invoice ORDER BY Billing_date DESC LIMIT 1";
$records = mysqli_query($db,$sql) or die( mysqli_error($db));
while($data = mysqli_fetch_array($records))
{
?>
  <tr>
    <td><?php echo $data['billing_id']; ?></td>
    <td><?php echo $data['game_id']; ?></td>
    <td><?php echo $data['Selling_price']; ?></td>
    <td><?php echo $data['Billing_date']; ?></td></tr>
<?php
}
?>
</tbody>
</table>
<?php mysqli_close($db); // Close connection
 ?>
<br><h4 align ="center">
Thankyou for shopping in Panash!<br>
We hope to see you soon<br>
Keep calm and game on!<br>
</div></div></div></h4></body>
<?php
header( "refresh:20; url=contact.php" );
?>
</html>
```

CHAPTER 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

6.2 TESTING OBJECTIVES

The main objectives of testing process are as follows.

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

6.3 TEST CASES

The test cases provided here test the most important features of the project.

6.3.1 Test cases of the project

Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a record	New tuple should be inserted	Query OK 1 row affected or inserted	PASS
2	Search a record	Search from existing records	Query OK 1 row affected or searched	PASS
3	Delete a record	Delete an existing record	Query OK 1 row affected or deleted	PASS
4	Create Trigger	Trigger created	Query OK trigger created	PASS
5	Create Assertions	Assertion created	Query OK assertion created	PASS

Table 6.3: Test cases of the project

Chapter 7

RESULTS

This section describes the screens of the “Project title”. The snapshots are shown below for each module.

7.1 Snapshots

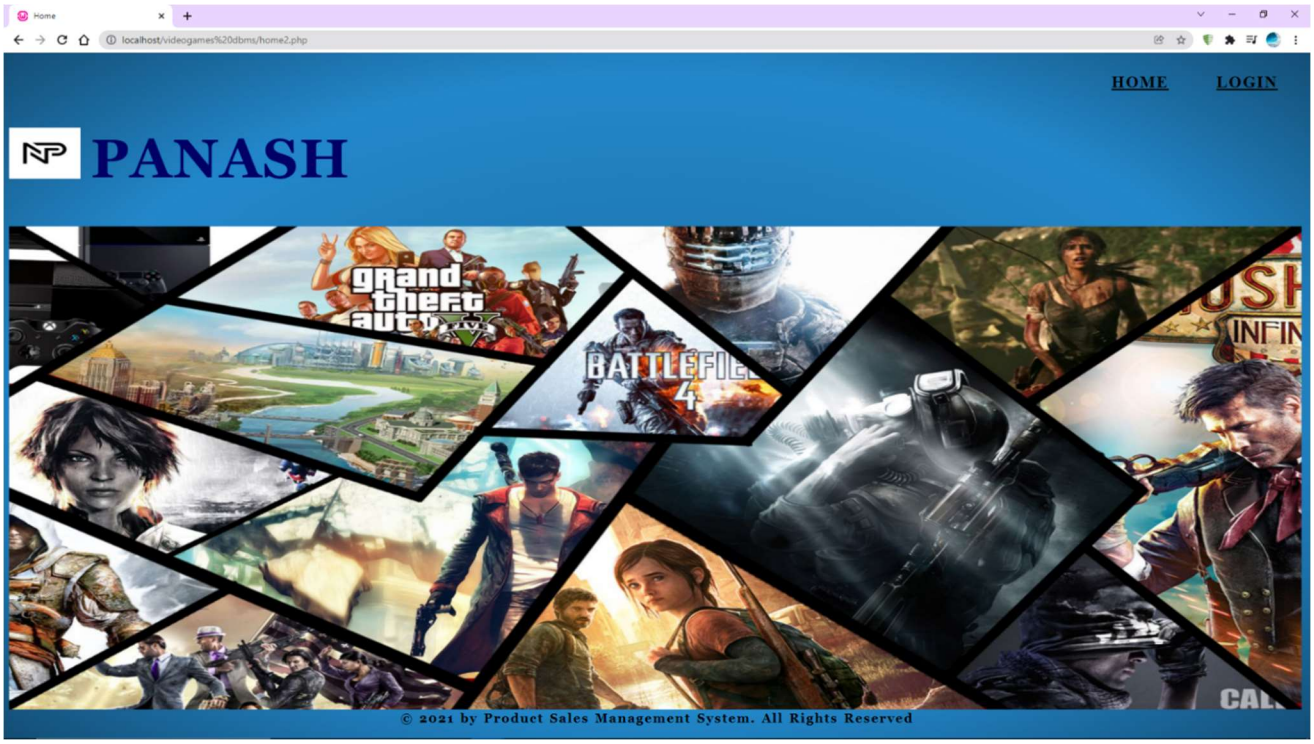


Fig 7.1.1: Snapshot1 Homepage

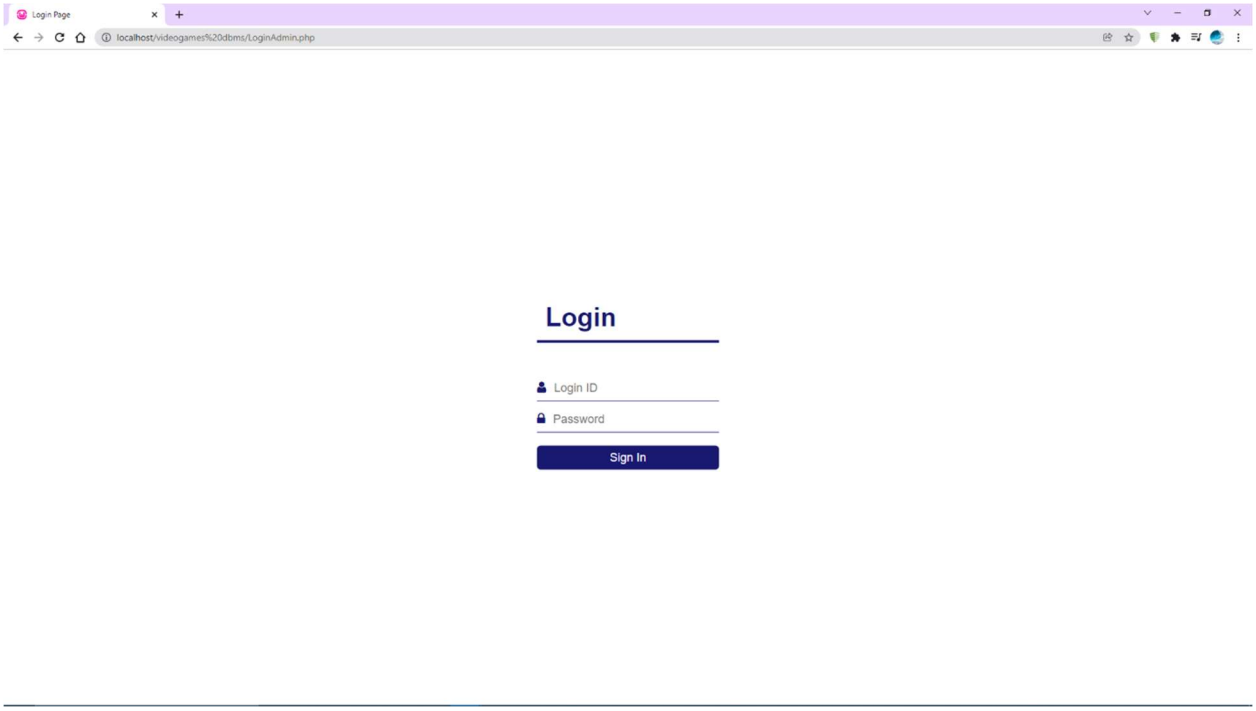


Fig. 7.1.2: Snapshot 2 Login Page

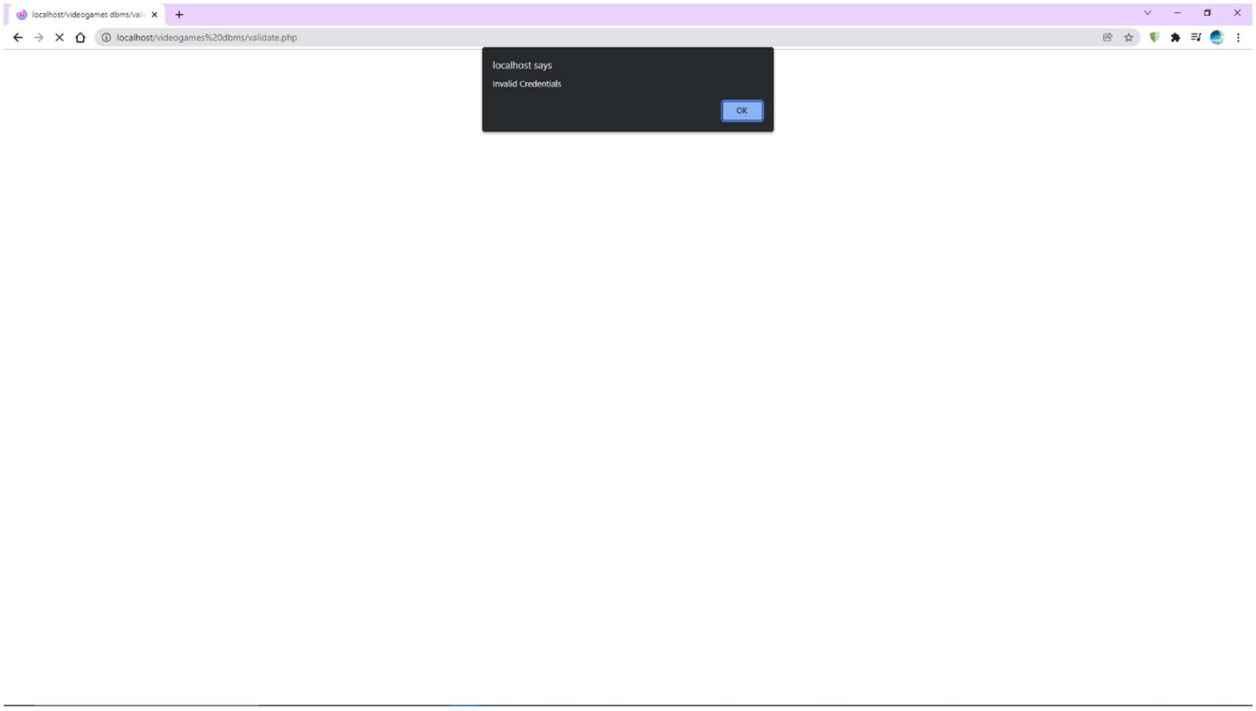


Fig. 7.1.3: Snapshot 3 Invalid Login credentials

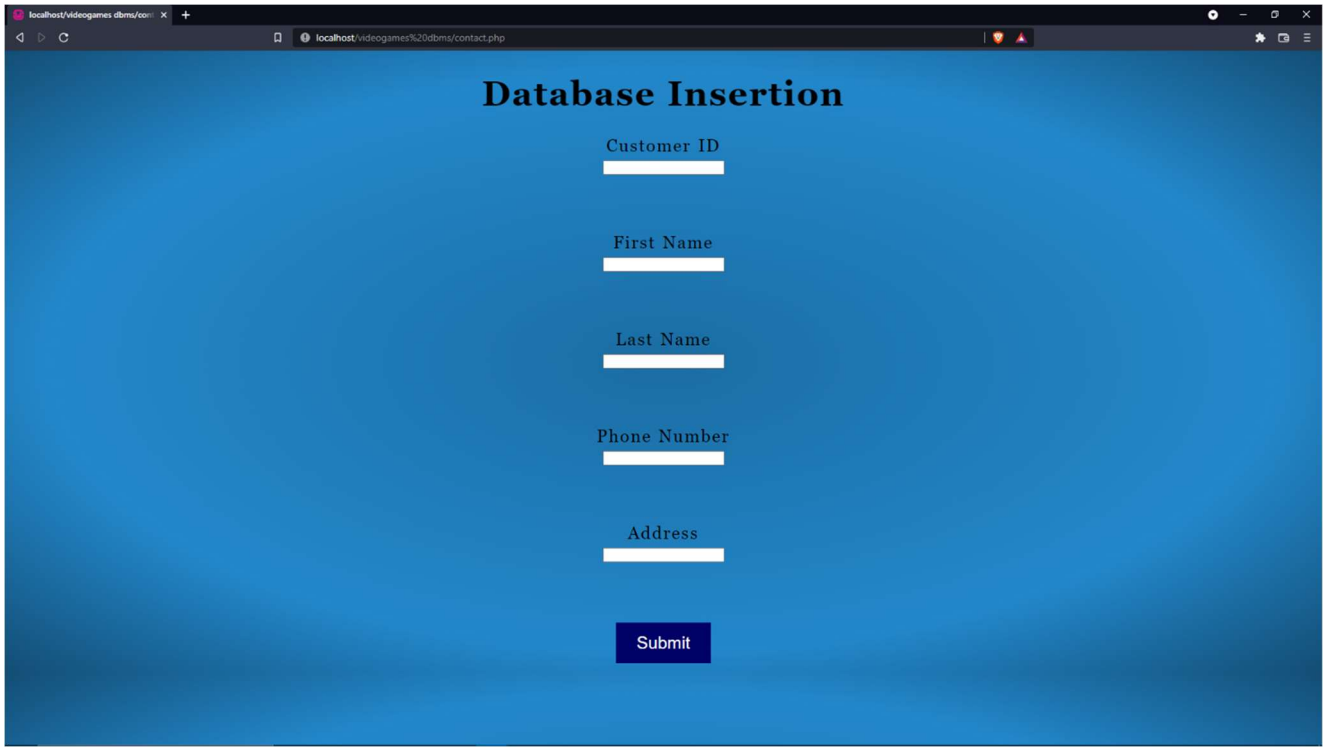


Fig. 7.1.4: Snapshot 4 Customer Details Page

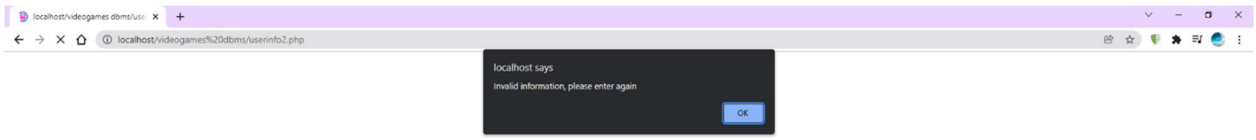


Fig. 7.1.5: Snapshot 5 Invalid Customer Details Page

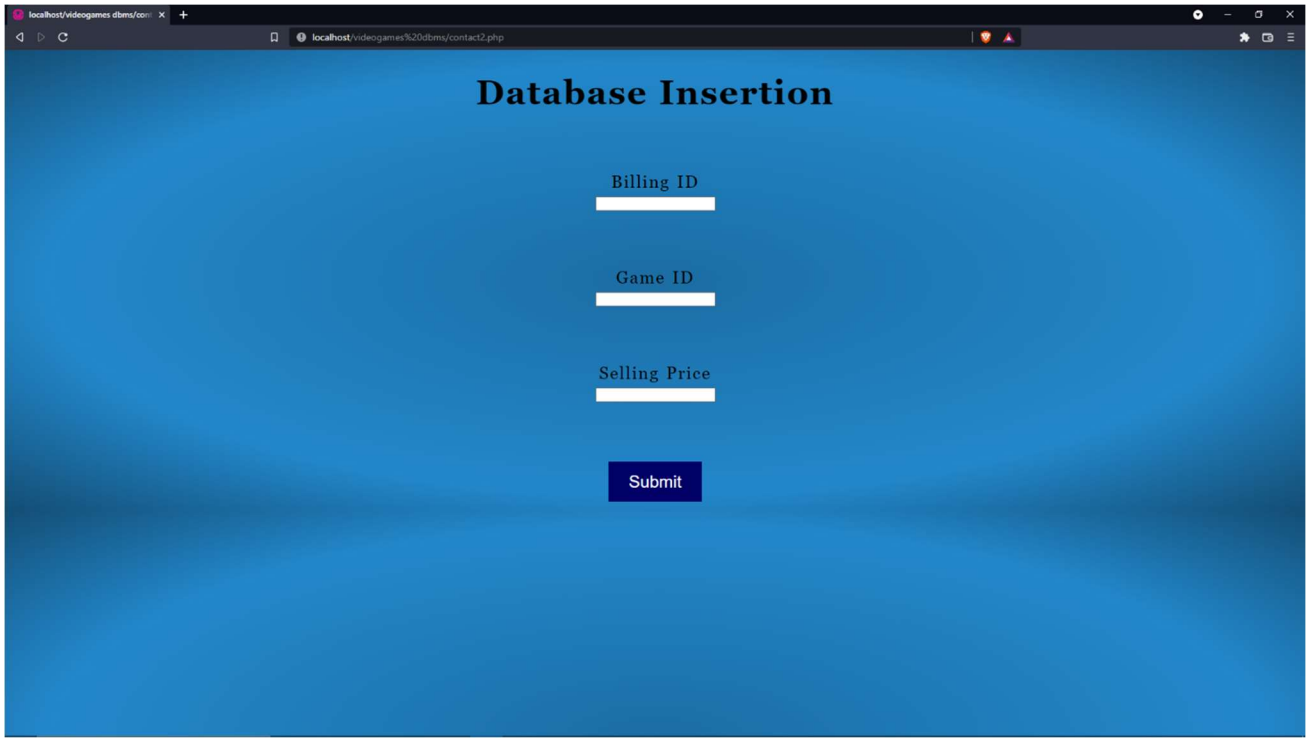


Fig. 7.1.6: Snapshot 6 Customer Purchase Details

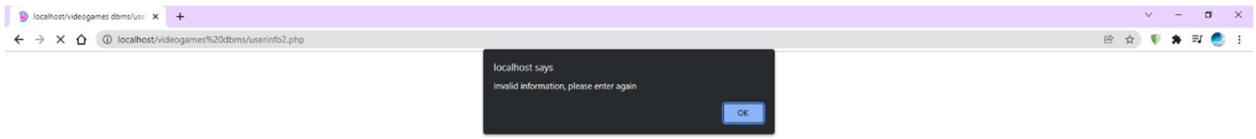


Fig. 7.1.7: Snapshot 7 Invalid Purchase Details

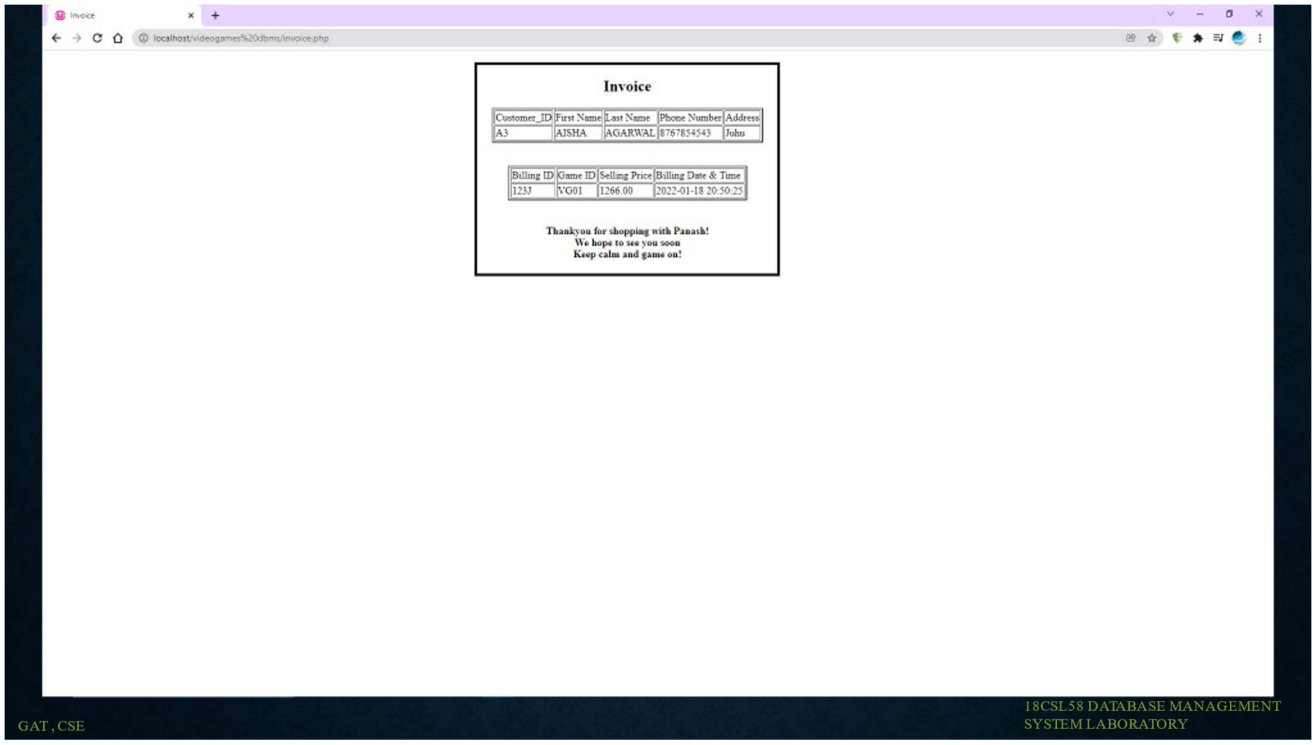


Fig. 7.1.8: Snapshot 8 Invoice Page

CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project "VIDEOGAMES MANAGEMENT SYSTEM" was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

The project also provides us the opportunity of interacting with our teachers and to gain from their best experience.

REFERENCES

- Fundamentals of Database System, Ramez Elmasri and Shamkant B.Navathe, 7th Edition, 2017, Pearson.
- Database Management System, Ramakrishna and Gehrke, 3rd Edition, 2014, McGraw Hill.
- The Complete Reference PHP from w3schools.com and geeksforgeeks.com.
- Website: http://php.net/manual/en/language.references.php & videos in youtube.com