

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JNANA SANGAMA”, BELAGAVI-590018



A MINI PROJECT REPORT ON

WEAPON BASED MANAGEMENT SYSTEM

Submitted in partial fulfillment of the
requirements for the award of degree of

Bachelor of Engineering

In

Computer Science and Engineering

By

SUJAY GS

[1KS18CS103]

Under the guidance of

Dr. Dayanand R.B

Prof. Dept. Of CSE

Mr. Kumar K

Asst. Prof, Dept. Of CSE



Department of Computer Science & Engineering

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

2020-2021

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Computer Science & Engineering



CERTIFICATE

This is to certify that mini project work entitled **“WEAPON BASED MANAGEMENT SYSTEM”** carried out by **Mr. SUJAY GS** bearing USN **1KS18CS103** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering in Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi**, during the year 2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini Project work prescribed for the said degree for the 5th semester.

Dr . Rekha. B. Venkatapur

Prof & HOD, CS & E Department

Dr . K.V.A Balaji

Principal/CEO KSIT

Dr. Dayanand R.B

Prof. Dept. Of CSE

Mr. Kumar K

Asst. Prof, Dept. Of CSE

Name of the Examiners

Signature with date

- 1.
- 2.

ACKNOWLEDGEMENT

I take this opportunity to thank one and all involved in building this project. Firstly I would like to thank the college for providing us an opportunity to work on the project.

I would also like to thank the management of **K. S. Institute of Technology** for providing all the resources required for the project.

I wish to acknowledge my sincere gratitude to our beloved Principal, **Dr. K.V.A.Balaji** for his encouragement and providing all facilities for the accomplishment of this project.

This project would not have been possible without the support of our beloved **Prof & HOD, Dr. Rekha.B.Venkatapur, Dept. of CSE.**

I am also highly grateful to my project guides, **Dr. Dayanand R.B** and **Mr. Kumar K,Dept. of CSE** who have been very generous in assisting and supporting, to do this Project **“Weapon Based Management System”**.

I also would like to thank all other teaching and non-teaching staff members who have extended their support and co-operation while bringing up this project.

SUJAY GS
(1KS18CS103)

ABSTRACT

The Phoenix Firearms is basically a weapon management system of a private organization that designs, develops, manages their own weapons and weaponizes the licensed government and private organization. Quality and safety always come first when it comes to the purchase of weapons. Phoenix Firearms do not entertain any illegal activities. In 21st century the weapons and military technology have been the pride of nation. Phoenix Firearms focuses on privatization of arms industry for the future development of weapons and technology to fight against threats that are imposed our world and maintained global peace.

CONTENTS

1	INTRODUCTION	7-11
1.1	OVERVIEW	7
1.2	PROBLEM STATEMENT	8
1.3	DATABASE MANAGEMENT SYSTEM	8
1.4	SQL	8
1.5	FRONT END DESIGN	9-10
1.6	CONNECTIVITY	11
2	REQUIREMENTS SPECIFICATION	12
2.1	OVERALL DESCRIPTION	12
2.2	SPECIFIC REQUIREMENTS	12
2.2.1	SOFTWARE REQUIREMENTS	12
2.2.2	HARDWARE REQUIREMENTS	12
3	DETAILED DESIGN	13-22
3.1	ENTITY RELATIONSHIP DIAGRAM	13-14
3.2	RELATIONAL SCHEMA	15-19
3.3	DESCRIPTION OF TABLES	20-21
3.4	DESCRIPTION OF FUNCTIONALITIES	22

4	IMPLEMENTATION	23-32
4.1	IMPLEMENTATION USING MYSQL	23-25
4.2	TRIGGERS AND STORED PROCEDURES	26
4.3	IMPLEMENTATION USING PHP	27-32
5	TESTING	33-34
5.1	TESTING	33-34
6	SNAPSHOTS	35-38
6.1	FRONT PAGE	35
6.2	GENERAL LOGIN PAGE	35
6.3	GENERAL REGISTRATION PAGE	36
6.4	GENERAL FRONT PAGE	36
6.5	WEAPON DETAILS	37
6.6	GUARD DETAILS	37
6.7	INVOICE	38
6.8	ADMIN LOGIN PAGE	38
	APPLICATIONS	40
	CONCLUSION	41
	FUTURE ENHANCEMENTS	42

Chapter 1

INTRODUCTION

1.1 OVERVIEW

India has the 3rd largest military in the world and is the 6th biggest defence spender. India is also one of the largest importers of conventional defence equipment and spends around 30% of its total defence budget on capital acquisitions. 60% of defence related requirements are currently met through imports. The Make in India initiative by the Government is focusing its efforts on increasing indigenous defense manufacturing and becoming self-reliant. The opening up of the defence sector for private sector participation is helping foreign original equipment manufacturers enter into strategic partnerships with Indian companies and leverage opportunities in the domestic market as well as global markets. The weapon equipments are supplied from Russia currently but through the initiative of “Make in India” the weapons can be manufactured in India and can be utilized by the defense personnel.

The Phoenix Firearms is basically a weapon management system of a private organization that designs, develops, manages their own weapons and weaponizes the licensed government and private organization. Quality and safety always come first when it comes to the purchase of weapons. Phoenix Firearms do not entertain any illegal activities. In 21st century the weapons and military technology have been the pride of nation. Phoenix Firearms focuses on privatization of arms industry for the future development of weapons and technology to fight against threats that are imposed our world and maintained global peace.

Equipments manufactured are vigorously challenged to meet stringent requirements to endure the harsh electrical conditions of today’s battlefields. End-to-end testing capabilities, expert test technicians & engineers, and single-source testing solutions helps meeting the demanding contract deadlines. Responsive team of experienced engineers and project support staff, are on-hand to advise, guide, and manage the whole process.

1.2 PROBLEM STATEMENT

The 'Phoenix Firearms' is a weapon management system of a private organization that designs, develops, manages their own weapons and weaponizes the licensed government and private organization .

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified ,and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases.

Originally based upon relational algebra and tuple relational calculus, SQL consists data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

1.5 FRONT END DESIGN

1.5.1 Screen layout Design for forms

HTML (Hyper Text Markup Language)

HTML is a standard markup language for creating web pages and we applications with Cascading Style Sheet (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

CSS (Cascading Style Sheet)

CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a corner stone technology of the World Wide Web, alongside HTML and JavaScript.

1.5.1.1 Screen layout design for Web pages and

forms HTML <p> TAG:

The HTML <p> element represents a paragraph of text. Paragraphs are usually represented in visual media as blocks of text that are separated from adjacent blocks by vertical blank space and/or first-line indentation. Paragraphs are block level elements.

HTML <div> TAG:

The HTML <div> element is the generic container for flow content and does not inherently represent anything. Use it to group elements for purposes such as styling (using the class or id attributes), marking a section of a document in a different language (using the lang attribute), and so on.

**HTML
 TAG:**

The HTML
 element produces a line break in text (carriage- return). It is useful for writing a poem or an address, where the division of lines is significant.

HTML <input> TAG:

The HTML <input> element is used to create interactive controls for web-based forms in order to accept data from the user. An <input> works varies considerably depending on the value of its type attribute, hence the different types are covered in their own separate reference pages. If this attributes is not specified, the default type adopted type is text.

The **HTML <thead>** element defines a set of rows defining the head of the columns of the table.

The **HTML <tbody>** element groups one or more <tr> elements as the body of a <table> element.

HTML <form> TAG:

The HTML <form> element represents a document section that contains interactive controls to submit information to a web server. It is possible to use the :valid and :invalid CSS pseudo-classes to style a <form>element. The HTTP method that the browser uses to submit the form. Possible values are:

post: Corresponds to the HTTP POST method ; form data are included in the body of the form and sent to the server.

get: Corresponds to the HTTP GET method; form data are appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server. Use this method when the form has no side-effects and contains only ASCII characters. This value can be overridden by a form method attribute on a <button> or <input>element.

action: The URI of a program that processes the form information. This value can be overridden by a form action attribute on a <button> or <input> element.

1.6 CONNECTIVITY

16.1 Connecting to a MySQL database

We need our MySQL server address (if the database is on the same server as the web server it will most likely be **localhost** or **127.0.0.1**), `username`, `password` and `database name`. Create a **filename.php** file and open and close the php code with tags before the `html`, we can put regular `html` after it. Open the file in a browser and we should be able to see nothing apart from the title tag, if we could see the error in `username/password` or `database name` which means there is something wrong.

```
// Create connection
```

```
<?php
$connect_error = 'sorry, server is down';
mysqli_connect('localhost', 'root', '') or die($connect_error);
mysqli_select_db('phoenix') or die($connect_error);
?>
```

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about weapons and military personnel details in an efficient, organized and time saving manner. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Weapon Management System are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

- Text Editor – Visual Studio Code
- Web Browser – Firefox 50 or later, Google Chrome – 60 or later
- Database support - MySQL 8.0
 - MySQL Server 8.0
 - MySQL Workbench
- Operating system – Windows 10 / Ubuntu 20.04.3
- Server deployment – Xampp Server 7.4.12-0

2.2.2 HARDWARE REQUIREMENTS

- Processor – Pentium IV or above
- RAM – 2 GB or more
- Hard disk – 3GB or more
- Monitor – VGA of 1024x768 screen resolution
- Keyboard and Mouse

Chapter 3

DETAILED DESIGN

3.1 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

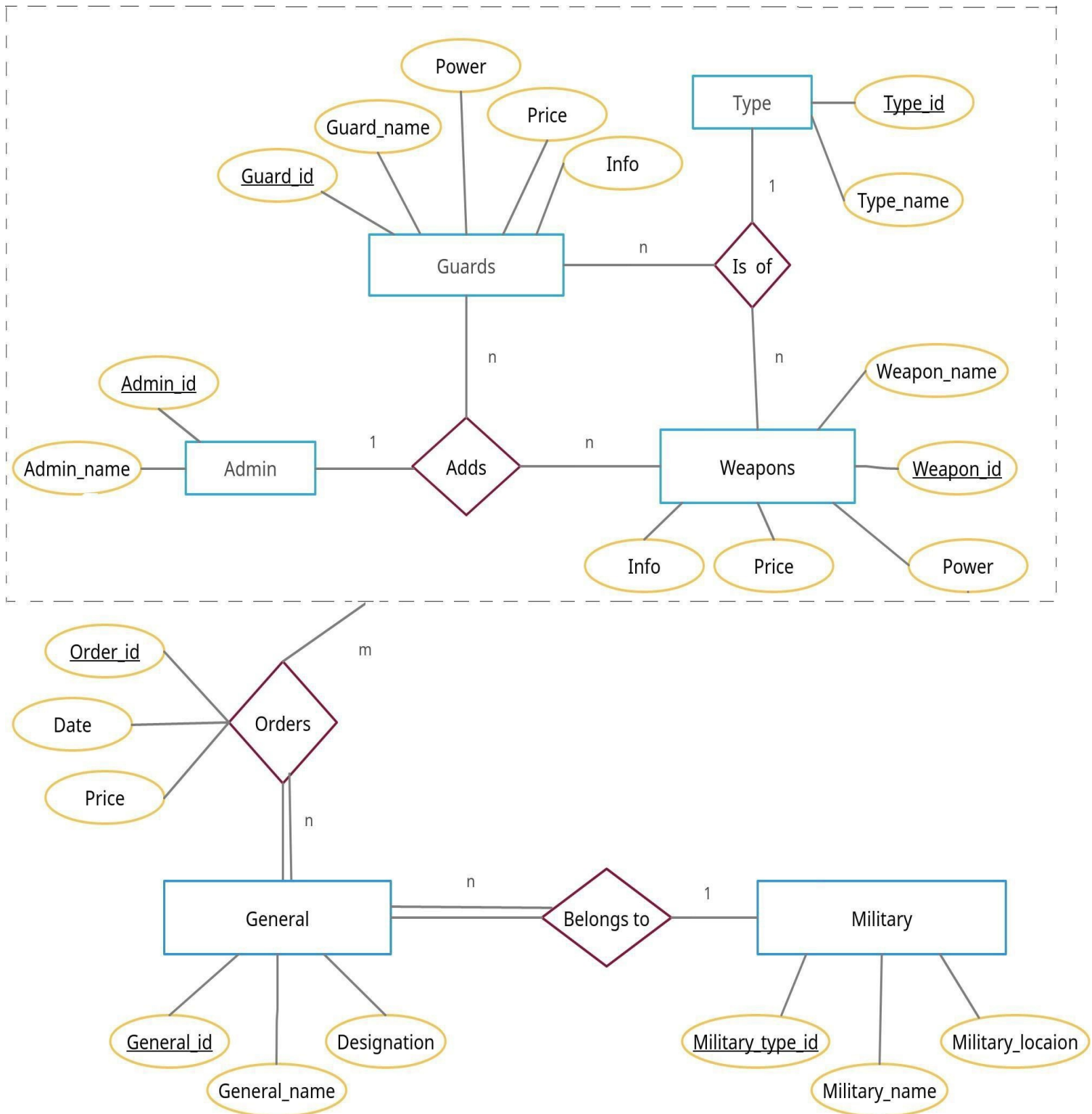
An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

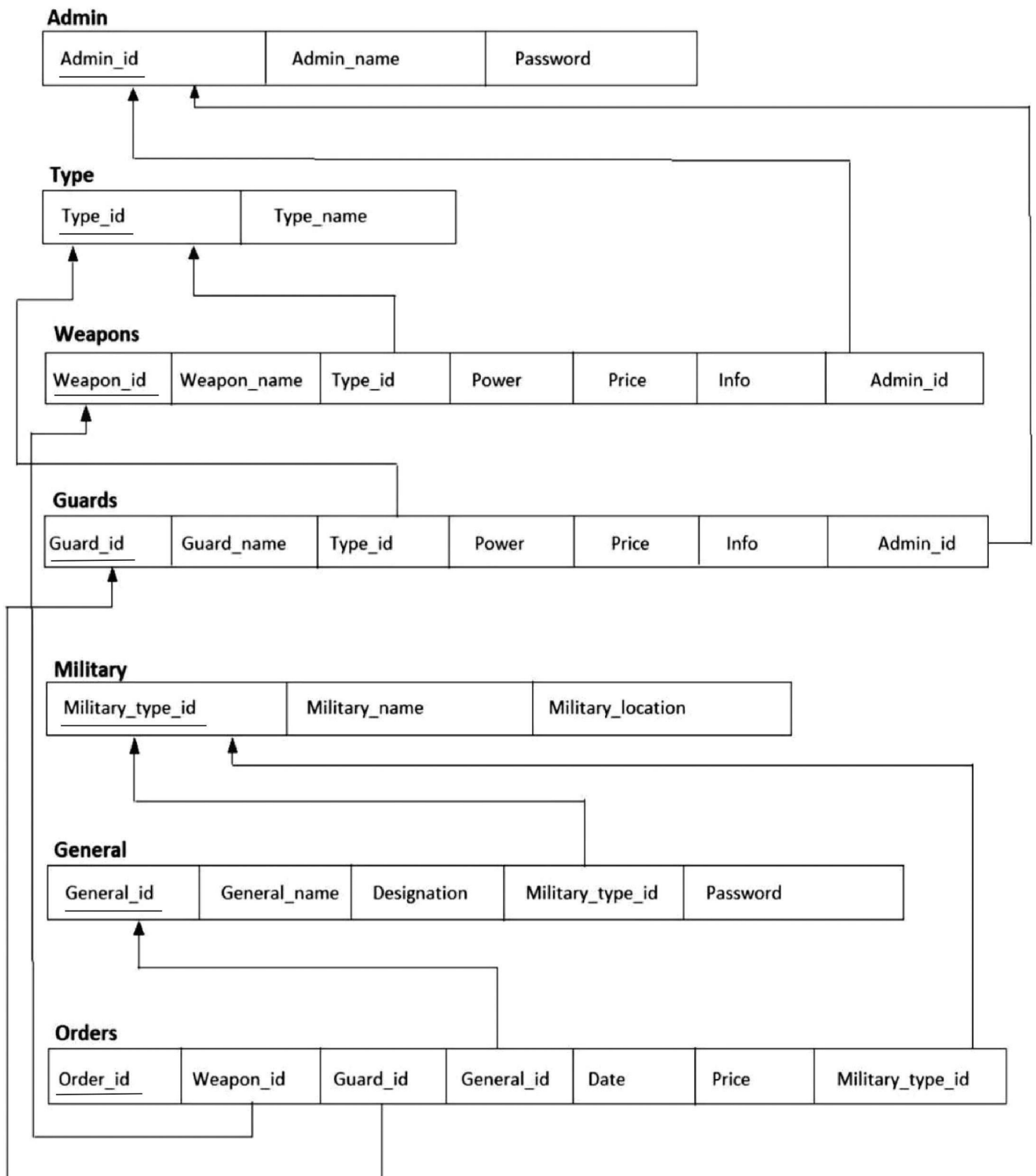
3.1.1 Conceptual Database Design (ER-Diagram)



3.2 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.


3.2.1 Logical Database Design (ER-Mapping)



3.2.2 NORMALIZATION

Admin

<u>Admin_id</u>	Admin_name	Password
-----------------	------------	----------



Admin_id-> Admin_name,Password


This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key and all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there is no transitive dependency for non-prime attributes.

Type

<u>Type_id</u>	Type_name
----------------	-----------



Type_id-> Type_name


This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key and all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there is no transitive dependency for non-prime attributes.

Weapons

<u>Weapon_id</u>	Weapon_name	Type_id	Power	Price	Info	Admin_id
------------------	-------------	---------	-------	-------	------	----------



Weapon_id->Weapon_name,Type_id,Power,Price,Info,Admin_id

This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key and all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there is no transitive dependency for non-prime attributes.

Guards

<u>Guard_id</u>	Guard_name	Type_id	Power	Price	Info	Admin_id
	↑	↑	↑	↑	↑	↑

Guard_id->Guard_name,Type_id,Power,Price,Info,Admin_id

This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key and all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there is no transitive dependency for non-prime attributes.

Military

<u>Military_type_id</u>	Military_name	Military_location
	↑	↑

Military_type_id->Military_name,Military_location

This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key and all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there is no transitive dependency for non-prime attributes.

General

<u>General_id</u>	General_name	Designation	Military_type_id	Password
	↑	↑	↑	↑

General_id->General_name,Designation,Military_type_id>Password

This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key an all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there no transitive dependency for non-prime attributes.

Orders

<u>Order_id</u>	Weapon_id	Guard_id	General_id	Date	Price	Military_type_id
	↑	↑	↑	↑	↑	↑

Order_id->Weapon_id,Guard_id,General_id,Date,Price,Military_type_id

This table is normalized in 1NF because there are all atomic values.

This table is normalized in 2NF because there is only one primary key an all the other attributes depend on this primary key only and there is no partial dependency.

This table is normalized in 3NF because there no transitive dependency for non-prime attributes

3.3 DESCRIPTION OF TABLES

The database consists of seven tables:

1.Admin: Admin has the control over the application.

- Admin_id: Unique admin id assigned to admin.
- Admin_name: Name of the admin.
- Password: Password of the admin.

2.Type: It is the type of the weapon/body armor.

- Type_id: It is the unique id assigned to each of the weapon/body armor.
- Type_name :Name of the type of weapon/body armor..

3.Weapons: Stores the weapon details.

- Weapon_id: Unique id assigned to weapon.
- Weapon_name: Name of the weapon.
- Type_id: Weapon type
- Power: Power of the weapon.
- Price: Price assigned to weapon.
- Info: Information related to the weapon.
- Admin_id: Unique admin id assigned to admin.

4.Guards: Stores the body armor details.

- Guard_id: Unique id assigned to guard.
- Guard_name: Name of the guard.
- Type_id: Guard type.
- Power: Power of the Guard.
- Price: Price assigned to guard.
- Info: Information related to the guard.
- Admin_id: Unique admin id assigned to admin.

5.Military: It stores the location details of the military personnel.

- Military_Type_id: Unique military id based on the location.
- Military_name: Location of the military personnel.
- Military_Location: Military base station name.

6.General: Stores the details of general.

- General_id: Unique id assigned to general.
- General_name: Name of the general.
- Designation: Designation of the military personnel.
- Military_Type_id: Unique military id based on the location.

7. Orders: It stores the order details of the military personnel.

- Order_id: Unique order id generated using triggers.
- Weapon_id: Unique id assigned to weapon.
- Guard_id: Unique id assigned to guard.
- General_id: Unique id assigned to general.
- Date: Date on which the order was placed.
- Price: Price of the order placed.

3.4 DESCRIPTION OF FUNCTIONALITIES

1. Admin Module:

The admin module will be used by the admins of this company, admin can hire view all the orders placed, add weapons/body armor and add generals.

Orders:

By using this functionality Admin can view the orders placed.

Add Weapon:

By using this functionality Admin can add new weapons.

Add Guard:

By using this functionality Admin can add new guards.

Profile:

By using this functionality Admin can view their profile.

Logout:

By using this functionality Admin can logout.

2. Organization Module:

This module is accessed by the military personnel. They can view weapons on the board to purchase, they can view their profile and their previous purchases.

Weapons & Orders:

By using this functionality military personnel can have a look into weapons and order the same.

My Orders:

By using this functionality military personnel can view the details of orders placed by them.

Profile:

By using this functionality military personnel can view their profile.

Logout:

By using this functionality military personnel can logout.

Chapter 4

IMPLEMENTATION USING PHP AND MYSQL

4.1 IMPLEMENTATION USING My SQL

The back end of the web application is basically the brains behind the front end. It comprises three components: server, application and database. It is a link between the server and the user. Most of the coding for the web application can be found in the back end and the quality of this code will determine how the website functions. MySQL is used as a back-end technology.

Creation of Tables:

```
CREATE TABLE `admin` (  
  `admin_id` int(6) NOT NULL,  
  `admin_name` varchar(20) NOT NULL,  
  `admin_password` varchar(20) NOT NULL  
)
```

```
CREATE TABLE `general` (  
  `general_id` varchar(20) NOT NULL,  
  `general_name` varchar(20) NOT NULL,  
  `designation` varchar(20) NOT NULL,  
  `military_type_id` varchar(10) NOT NULL,  
  `set_password` varchar(20) NOT NULL,  
  `confirm_password` varchar(20) NOT NULL  
)
```

```
CREATE TABLE `guards` (  
  `guard_id` varchar(10) NOT NULL,  
  `guard_name` varchar(10) NOT NULL,  
  `type_id` varchar(10) NOT NULL,  
  `power` varchar(10) NOT NULL,  
  `price` float NOT NULL,  
  `info` mediumtext NOT NULL,  
  `image` text NOT NULL  
)
```

```
CREATE TABLE `military` (  
  `military_type_id` varchar(10) NOT NULL,  
  
  `military_name` varchar(40) NOT NULL,
```

```
`military_location` varchar(20) NOT NULL)
```

```
CREATE TABLE `orders` (  
  `order_id` varchar(10) NOT NULL,  
  `weapon_id` varchar(6) DEFAULT NULL,  
  `guard_id` varchar(10) DEFAULT NULL,  
  `general_id` varchar(6) NOT NULL,  
  `date` date NOT NULL,  
  `price` float NOT NULL,  
  `military_type_id` varchar(10) NOT NULL  
)
```

```
CREATE TABLE `type` (  
  `type_id` varchar(6) NOT NULL,  
  `type_name` varchar(20) NOT NULL  
)
```

```
CREATE TABLE `weapons` (  
  `weapon_id` varchar(6) NOT NULL,  
  `weapon_name` varchar(10) NOT NULL,  
  `type_id` varchar(6) NOT NULL,  
  `admin_id` int(6) DEFAULT NULL,  
  `power` varchar(10) NOT NULL,  
  `price` float NOT NULL,  
  `info` mediumtext NOT NULL,  
  `image` text NOT NULL  
)
```

```
ALTER TABLE `admin`  
ADD PRIMARY KEY (`admin_id`);
```

```
ALTER TABLE `general`  
ADD PRIMARY KEY (`general_id`),  
ADD KEY `military_type_id` (`military_type_id`);
```

```
ALTER TABLE `guards`  
ADD PRIMARY KEY (`guard_id`),  
ADD KEY `type_id` (`type_id`);
```

```
ALTER TABLE `military`  
ADD PRIMARY KEY (`military_type_id`);
```

```
ALTER TABLE `orders`  
ADD PRIMARY KEY (`order_id`),
```

```
ADD KEY `weapon_id` (`weapon_id`),  
ADD KEY `general_id` (`general_id`),  
ADD KEY `military_type_id` (`military_type_id`),  
ADD KEY `guard_id` (`guard_id`);
```

```
ALTER TABLE `type`  
ADD PRIMARY KEY (`type_id`);
```

```
ALTER TABLE `weapons`  
ADD PRIMARY KEY  
(`weapon_id`), ADD KEY `type_id`  
(`type_id`), ADD KEY `admin_id`  
(`admin_id`);
```

```
ALTER TABLE `general`  
ADD CONSTRAINT `general_ibfk_1` FOREIGN KEY (`military_type_id`)  
REFERENCES `military` (`military_type_id`);
```

```
ALTER TABLE `guards`  
ADD CONSTRAINT `guards_ibfk_1` FOREIGN KEY (`type_id`)  
REFERENCES `type` (`type_id`);
```

```
ALTER TABLE `orders`  
ADD CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`weapon_id`)  
REFERENCES `weapons` (`weapon_id`),  
ADD CONSTRAINT `orders_ibfk_3` FOREIGN KEY (`general_id`)  
REFERENCES `general` (`general_id`),  
ADD CONSTRAINT `orders_ibfk_4` FOREIGN KEY (`military_type_id`)  
REFERENCES `military` (`military_type_id`),  
ADD CONSTRAINT `orders_ibfk_5` FOREIGN KEY (`guard_id`)  
REFERENCES `guards` (`guard_id`);
```

```
ALTER TABLE `weapons`  
ADD CONSTRAINT `weapons_ibfk_1` FOREIGN KEY (`type_id`)  
REFERENCES `type` (`type_id`),  
ADD CONSTRAINT `weapons_ibfk_2` FOREIGN KEY (`admin_id`)  
REFERENCES `admin` (`admin_id`);  
COMMIT;
```


4.2 TRIGGERS AND STORED PROCEDURE:

Triggers:

```
CREATE TRIGGER `inc`  
AFTER INSERT ON `orders`  
FOR EACH ROW update counter set cntr=cntr+1 where cntkey=1
```

Stored Procedure:

```
CREATE PROCEDURE viewsac()  
BEGIN  
Select * from Admin;  
END
```

4.3 IMPLEMENTATION USING PHP

General_login.php

```
<?php
session_start();
$hostname = "localhost"; $username="root";
$password = ""; $database="phoneix";
$con = mysqli_connect($hostname,$username,$password);
if(! $con) {
die('Connection Failed'.mysqli_error());
}
mysqli_select_db($con,$database);
if($_SERVER["REQUEST_METHOD"] == "POST") {
$gi = $_POST['general_id']; $mi=$_POST['military_type_id'];
$sp = $_POST['set_password'];

$s="SELECT * FROM general WHERE general_id='$gi' AND
military_type_id = ' $mi' AND set_password='$sp'"or
die("unable to connect".mysqli_error($con));

$result = mysqli_query($con,$s);
$row = mysqli_fetch_array($result,MYSQLI_ASSOC);

$_SESSION['general_id'] = $row['general_id'];
$_SESSION['general_name'] = $row['general_name'];
$_SESSION['designation'] = $row['designation'];
$_SESSION['military_type_id'] = $row['military_type_id'];
if (mysqli_num_rows($result) == 1)
{ header('Location:http://localhost/project/about.php');
}
else{
function function_alert2($message)
{ echo"<script>alert('$message'); history.go(-1);
</script>";
}
function_alert2("Check your General ID/Password.");
}
}
?>
```

general_register.php

```

<?php
$hostname = "localhost"; $username = "root";
$password = ""; $database = "phoneix";
$con = mysqli_connect($hostname, $username, $password);
if (!$con) {
    die("Connection Failed" . mysqli_error());
}

mysqli_select_db($con, $database);

function function_alert($message) {
    echo "<script>alert('$message');
    history.go(-1);
    </script>";
}

if (isset($_POST["submit"])) {
    $gi = $_POST["general_id"]; $name = $_POST["general_name"];
    $d = $_POST["designation"]; $mi = $_POST["military_type_id"];
    $sp = $_POST["set_password"]; $cp = $_POST["confirm_password"];
    $sql_u = "SELECT * FROM general WHERE general_id='$gi'";

    $res_u = mysqli_query($con, $sql_u);

    if (mysqli_num_rows($res_u) > 0) {
        function_alert("General ID already exists");
    } else {
        if ($sp == $cp) {
            $sql = "insert into
general(general_id,general_name,designation,military_type_id,set_password,confirm_password)
values('$gi','$name','$d','$mi','$sp','$cp')";
            $res = mysqli_query($con, $sql);

            if ($res) {
                header("Location: http://localhost/project/index.html");
            } else {
                function_alert("There is problem in inserting records");
            }
        } else {
            function_alert("Passwords do not match");
        }
    }
}
?>

```

admin_login.php

```

<?php
session_start();
$hostname = "localhost"; $username = "root";
$password = ""; $database = "phoneix";
$con=mysqli_connect($hostname,$username,$password);
if(! $con){
die('Connection Failed'.mysqli_error());
}

mysqli_select_db($con,$database);

if($_SERVER["REQUEST_METHOD"] == "POST") {
    $ai = $_POST['admin_id']; $an = $_POST['admin_name'];
    $ap = $_POST['admin_password'];

    $s = "SELECT * FROM admin WHERE admin_id = '$ai' AND admin_name = '$an' AND
admin_password = '$ap'" or die( "unable to connect".mysqli_error($con));

    $result = mysqli_query($con,$s);
    $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
    $_SESSION['admin_id'] = $row['admin_id'];
    $_SESSION['admin_name'] = $row['admin_name'];
    if (mysqli_num_rows($result) == 1) {
        header('Location: http://localhost/project/admin orders.php');
    }
    else{
        function function_alert2($message){
            echo"<script>alert('$message');
            history.go(-1);
            </script>";
        }
        function_alert2("Enter the correct details.");
    }
}
?>

```

admin_addweapons.php

```

<?php
session_start();
$hostname = "localhost"; $username = "root";
$password = ""; $database = "phoneix";
$con = mysqli_connect($hostname,$username,$password);
if(! $con) {
die('Connection Failed'.mysqli_error());}

```

```
mysqli_select_db($con,$database);

function function_alert($message){
    echo"<script>alert('$message');
    history.go(-1);
    </script>";
}

if(isset($_POST['submit'])) {
    $wi = $_POST['weapon_id']; $wn = $_POST['weapon_name'];
    $ti = $_POST['type_id']; $p = $_POST['power'];
    $pr = $_POST['price']; $info = $_POST['info'];
    $image = $_POST['image'];

    $sql_wi = "SELECT * FROM weapons WHERE weapon_id='$wi'";
    $res_wi = mysqli_query($con, $sql_wi);

    if (mysqli_num_rows($res_wi) > 0)
    { function_alert("Weapon ID already exists");
    }
    else{
        $sql = "insert into weapons(weapon_id,weapon_name,type_id,power,price,info,image)
        values('$wi','$wn','$ti','$p','$pr','$info','$image')";
        $result = mysqli_query($con,$sql);

        $filename = $_FILES["image"]["name"];
        $tempname = $_FILES["image"]["tmp_name"];
        $folder = "image/".$filename;
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        if($result){
            function_alert("Weapon inserted successfully");
        }
        else{
            function_alert("Failed to insert");
        }
    }
    mysqli_close($con);
?>
```

admin_addgeneral.php

```
<?php
session_start();
$hostname = "localhost"; $username = "root";
$password = ""; $database = "phoneix";

$con = mysqli_connect($hostname,$username,$password);
if(! $con) {
die('Connection Failed'.mysqli_error());
}

mysqli_select_db($con,$database);

$sql = "Select * from general;";
$result = mysqli_query($con,$sql);
while($row = mysqli_fetch_assoc($result)) {
    echo"<tr>
        <td>".$row['general_name']. "</td>
        <td>".$row['general_id']. "</td>
        <td>".$row['designation']. "</td>
        <td>".$row['military_type_id']. "
    </tr>";
}
?>
```

order_weapons.php

```
<?php
session_start();
$hostname = "localhost"; $username = "root";
$password = ""; $database = "phoneix";
$con = mysqli_connect($hostname,$username,$password);
if(! $con) {
die('Connection Failed'.mysqli_error());
}
mysqli_select_db($con,$database);

function function_alert($message){
    echo"<script>alert('$message');
    history.go(-1);
    </script>";
}

if(isset($_POST['submit'])) {
    $wi = $_POST['weapon_id']; $gi = $_POST['general_id'];
    $d = date('Y/m/d');
```

```

$p = "SELECT price FROM weapons WHERE weapon_id = '$wi'";
$mi = $_POST['military_type_id'];

$r = mysqli_query($con,$p);
$row = mysqli_fetch_assoc($r);
$v = $row['price'];
$result2 = mysqli_query($con,"select cntr from counter where cntkey=1");
    $k = mysqli_fetch_assoc($result2);
    $s = "#phx".strval(sprintf("%03d",$k['cntr']));

if($gi == $_SESSION['general_id'] && $mi == $_SESSION['military_type_id']){

$sql = "insert into orders(order_id,weapon_id,general_id,date,price,military_type_id)
values('$s','$wi','$gi','$d','$v','$mi')";
$res = mysqli_query($con, $sql);
if($res){
    function_alert("Weapon Order place Successfully!!!");
}
else{
    function_alert("Error while ordering.");
}
}
else{
    function_alert("Invalid General ID/Military Location");
}
}
?>

```

logout.php

```

<?php
    session_start();
    // Destroy session
    if(session_destroy()) {
        header("Location: http://localhost/project/index.html");
    }
?>

```

Chapter 5

TESTING

5.1 TESTING

Test case description	Test case	Expected Output	Obtained Output	Result
Register	The information has to be stored in database if all the fields are filled	The information is stored in database.	Accepted	Pass
Register	If user Military ID \neq database[Military ID], then no redirect	It should not redirect to the next page	Rejected	Pass
Login	If General ID = database[General ID] and password = database[password], then redirect	It has to redirect to next page	Accepted	Pass
Login	If General ID \neq database[General ID] and password \neq database[password], then no redirect	It should not redirect to the next page	Rejected	Pass
Buy	If General ID = database[General ID] and Military ID = database[Military ID] then generate invoice and store in database	Generate invoice and store in database	Accepted	Pass
Buy	If General ID \neq database[General ID] and Military ID \neq database[Military ID] then abort booking	Abort booking	Rejected	Pass

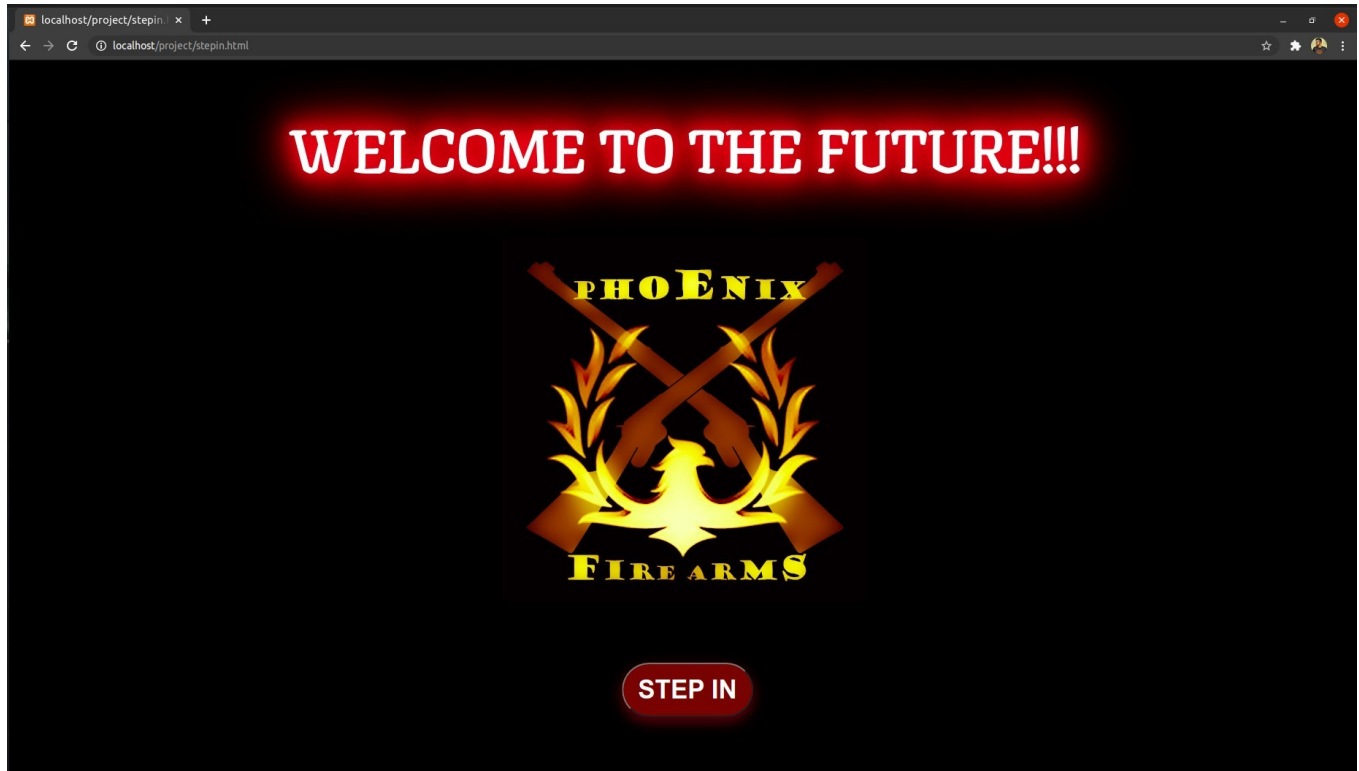
Test case description	Test case	Expected Output	Obtained Output	Result
Admin	The information has to be fetched from database and showed in the table format	It has to show the order details of respective people	Accepted	Pass
Admin	Adds weapons and guards and stores in database	The information is stored in database	Accepted	Pass
Admin	Fresher details should be stored in database	The details is stored in database	Accepted	Pass

Chapter 6

SNAPSHOTS

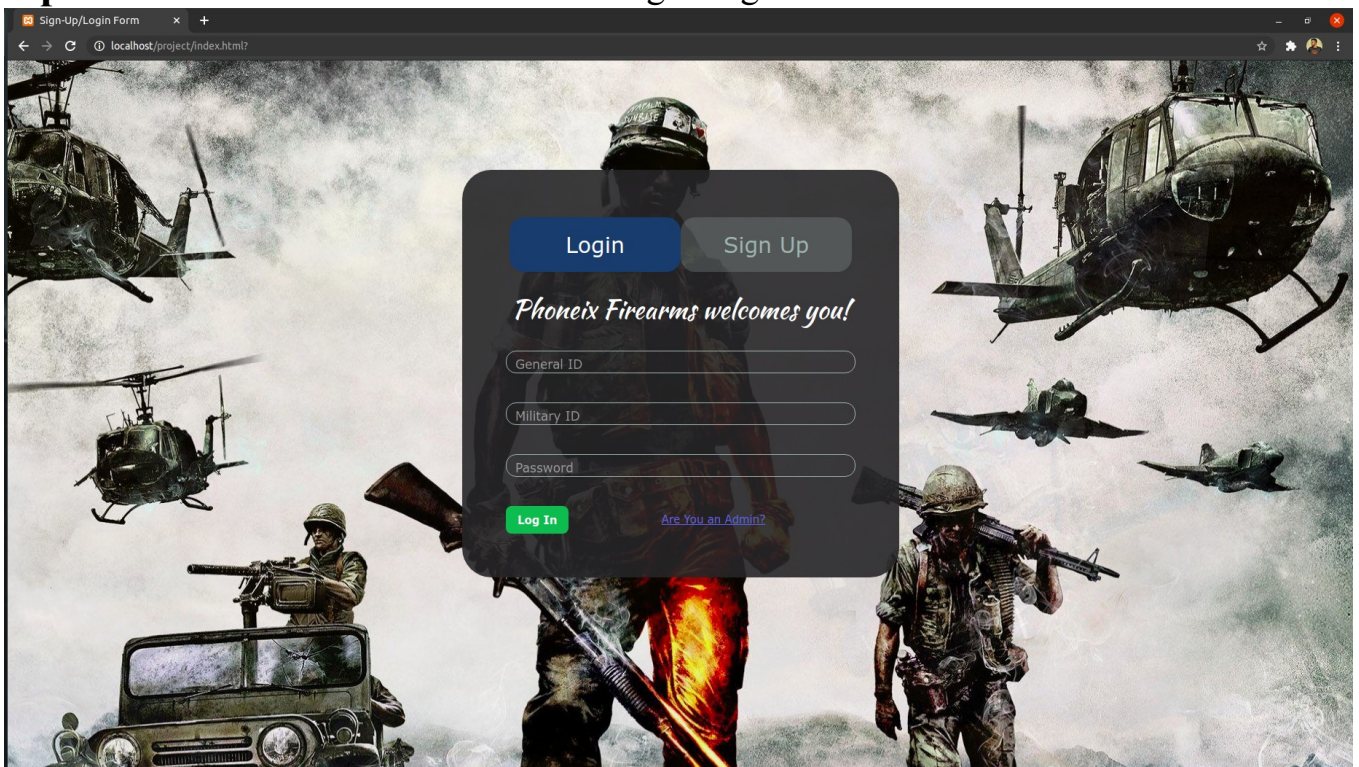
Snapshot-1:

Front page

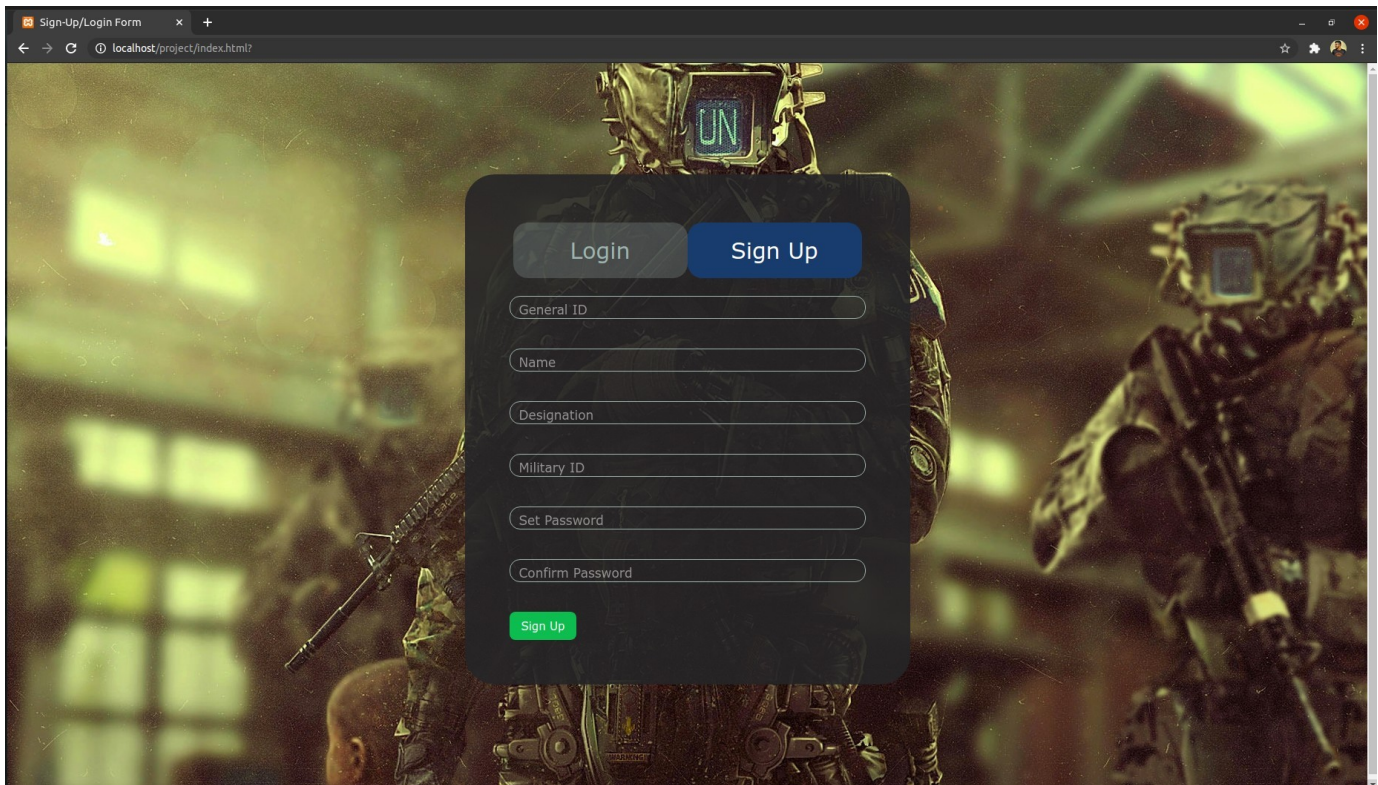


Snapshot-2:

General Login Page

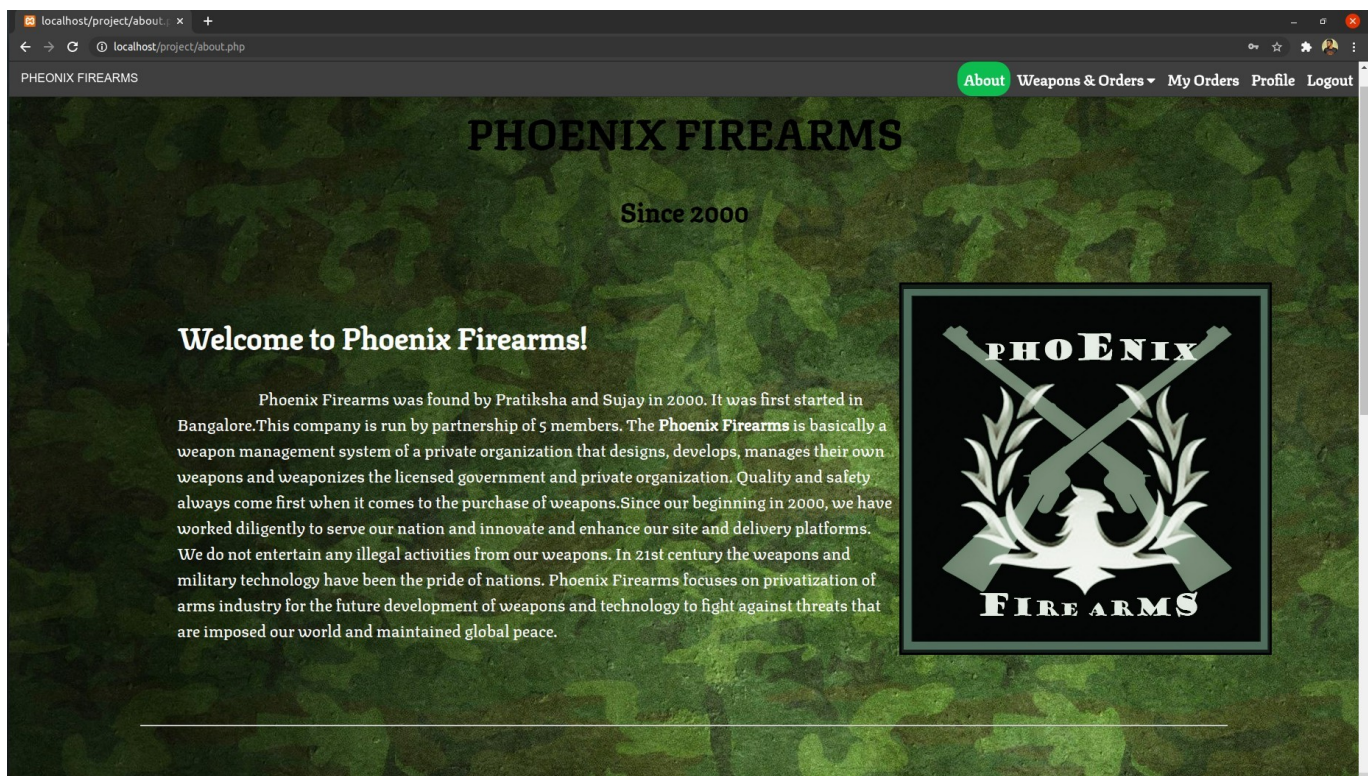


Snapshot-3: General Registration Page



The screenshot shows a web browser window with the address bar displaying 'localhost/project/index.html?'. The page features a dark, tactical-themed background with a soldier in camouflage gear. A central, semi-transparent dark box contains the registration form. At the top of this box are two buttons: 'Login' (light blue) and 'Sign Up' (dark blue). Below these are six input fields: 'General ID', 'Name', 'Designation', 'Military ID', 'Set Password', and 'Confirm Password'. At the bottom of the form is a green 'Sign Up' button.

Snapshot-4: General Front Page



The screenshot shows the front page of the Phoenix Firearms website. The header includes the site name 'PHONEIX FIREARMS' and a navigation menu with links: 'About', 'Weapons & Orders', 'My Orders', 'Profile', and 'Logout'. The main content area has a green camouflage background. It features the text 'PHONEIX FIREARMS' and 'Since 2000' at the top. Below this is a 'Welcome to Phoenix Firearms!' section followed by a paragraph of text: 'Phoenix Firearms was found by Pratiksha and Sujay in 2000. It was first started in Bangalore. This company is run by partnership of 5 members. The Phoenix Firearms is basically a weapon management system of a private organization that designs, develops, manages their own weapons and weaponizes the licensed government and private organization. Quality and safety always come first when it comes to the purchase of weapons. Since our beginning in 2000, we have worked diligently to serve our nation and innovate and enhance our site and delivery platforms. We do not entertain any illegal activities from our weapons. In 21st century the weapons and military technology have been the pride of nations. Phoenix Firearms focuses on privatization of arms industry for the future development of weapons and technology to fight against threats that are imposed our world and maintained global peace.' To the right of the text is a logo featuring a white silhouette of a phoenix rising from flames, with two crossed rifles behind it. The word 'PHONEIX' is at the top and 'FIRE ARMS' is at the bottom of the logo.


Snapshot-5:

Weapon Details

PHONEIX FIREARMS

About Weapons & Orders My Orders Profile Logout

GUN TYPE: ASSAULT RIFLE



Weapon Name	BLN-007
Weapon ID	GAR1
Power	10,000J
Price	4.3L
Type ID	GAR
Information	The BLN-007 is a 7.62x39mm Assault Rifle. This is a common modernised variant of the AK-47 rifle developed in the year 2000. It is a selective fire, gas operated with a rotating bolt, firing in either semi-automatic or fully automatic, and has a cyclic Rate of fire of around 600 RPM . The weight of the loaded magazine around 1.806 lb .

Snapshot-6:

Guard Details


localhost/project/defense x +

localhost/project/defensive%20weaponary.php

PHONEIX FIREARMS

About Weapons & Orders My Orders Profile Logout


GUARD TYPE: BODY ARMOUR



Guard Name	ASL
Guard ID	BEBA1
Power	B-P
Price	1.95L
Type ID	BEBA
Information	ASL vest was in use since 1949 by the Indian Army. It can stop pistol rounds with less diameter. Designed to protect against small explosions. It has front, back, side and shoulder ballistic protection with ballistic collar and removable groin protector. It includes a steel core with hardness rating ranging from RC20 mild steel up to RC25 medium hard steel .

Snapshot-7:

Invoice



PHONEIX FIREARMS
Bangalore
India

INVOICE TO:
IMFM19: Sitabuldi Fort
Nagpur

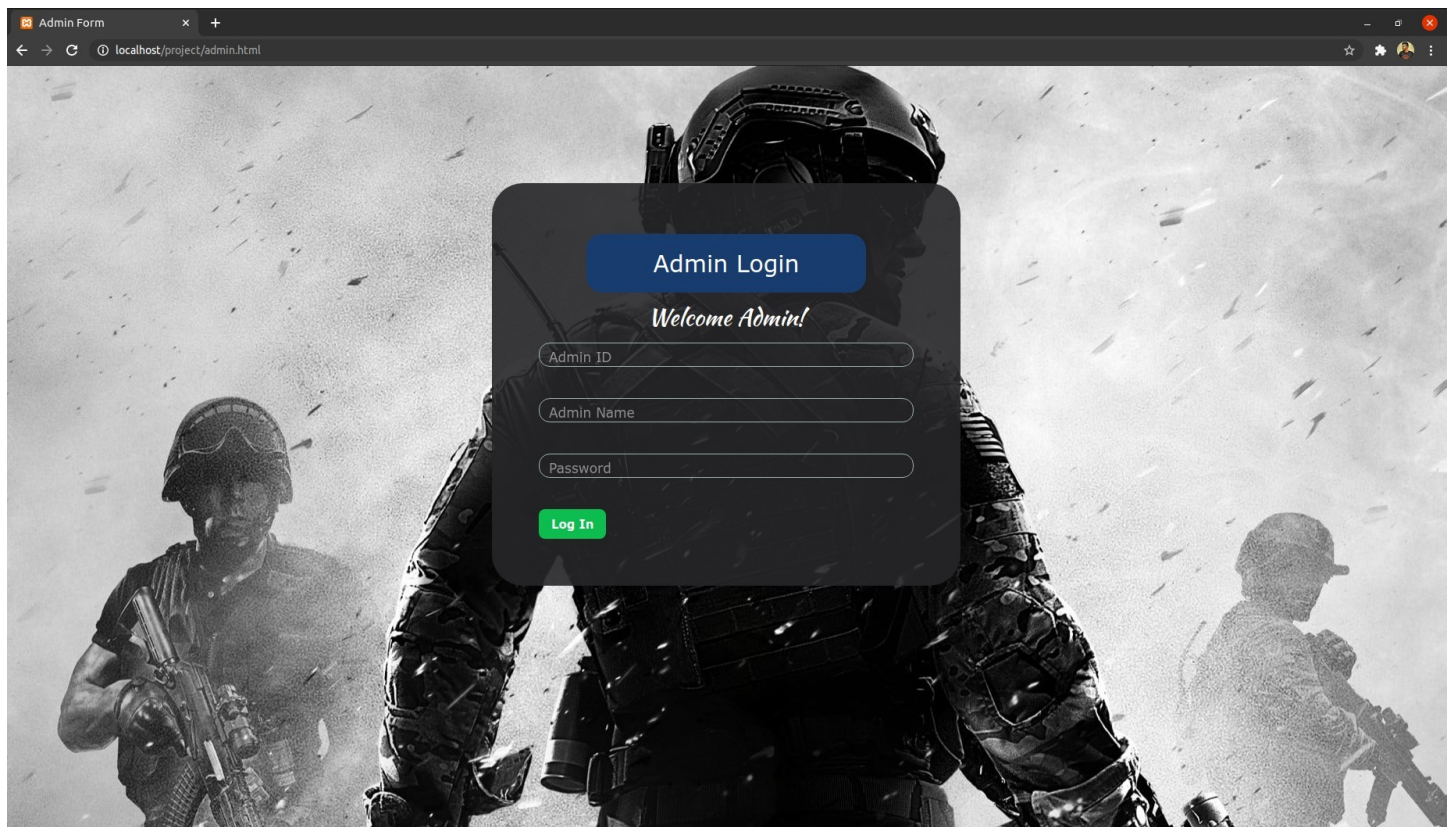
Weapon ID:
GSMG3
Date: 24-12-2020

WEAPON ID	WEAPON NAME	PRICE TOTAL
GSMG3	PSB-L07	Rs 3.15L

SUBTOTAL	GST 20%	Grand Total
Rs 3.15 L	Rs 0.63 L	Rs 3.78 L

Snapshot-8:

Admin Login Page



Admin Form

localhost/project/admin.html

Admin Login

Welcome Admin!

Admin ID

Admin Name

Password

Log In

Chapter 7

APPLICATIONS

Applications:

- ◆ Phoenix Firearms helps in privatization of arms industry and opens the door for future technology and minds to build a better world.
- ◆ Phoenix Firearms helps in militarizing, reinforcing military organization, country's defense system to uphold the law and order and to defend themselves from enemies and anti-social elements.
- ◆ Helps to fight against the main threat in the world of terrorism. Through weaponization of private anti terrorist organization with right and efficient technology global peace can be achieved.
- ◆ Provides control over the threats imposed by other countries.
- ◆ Cost appears to be a major incentive when purchased from other nations.
- ◆ As military technologies become more complex and more expensive, Phoenix Firearms stands by the concept of 'Make In India' to support the economy of the country.

Chapter 8

CONCLUSION

Conclusion:

The Weapon Based Management System provides easier maintenance of various weapons that military personnel will invest in. It allows simplified operation and is a time saving platform with the ability to view orders placed. The application has been completed successfully and tested with suitable test cases. It is user friendly and contains suitable options. This is developed using HTML5, CSS, JavaScript, PHP and SQL. The goals achieved by this project are:

- Centralized database.
- Easier buying, selling of various weapons.
- User friendly environment.
- Efficient management of weapons.
- Ability to view orders placed and invoice is generated.

Chapter 9

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

- Inclusion of license for weapons and the availability of weapons for a particular license.
- Better security implementations between the organization and military personnel.
- Tracking the location of the weapon until it reaches the destination.