**Chapter 1**

**Introduction**

Assembling a computer yourself is a good way to learn how they work. Our Website is an option to build a desktop computer by selecting the PC parts like Processor, Motherboard, GPU, RAM, SSD, and other PC parts as per your choice and requirements.

**1.1 Objectives**

* Building a PC allows you to handpick every component that goes into your machine. When you have total control over your computer's internal components, the final product can have a better overall build quality.
* To provide assistance to the users.
* To inspire the users to do good PC.

**1.2 Scope of the project**

It is focused on studying of PC BUILD in and to make sure that the users can build good Desktops. This also will produce:

* Less effort and the primary cost and focus primarily on creating, managing, and running a secure desktop maintenance.
* Increasing number of users will find it easier and more convenient to build the desktop.

**Chapter 2**

**SYSTEM REQUIREMENTS**

**2.1 Hardware Requirements**

Processor : i5 Core Processor

Clock speed : 1.19 GHz

RAM : 8 GB 3.2

**2.2 Software Requirements**

Operating System : 64-bit Operating System, Windows 10 or Ubuntu 22.04

Database : MySQL

Web Server : XAMPP

IDE : Visual Studio Code

Scripting Language : PHP , JavaScript

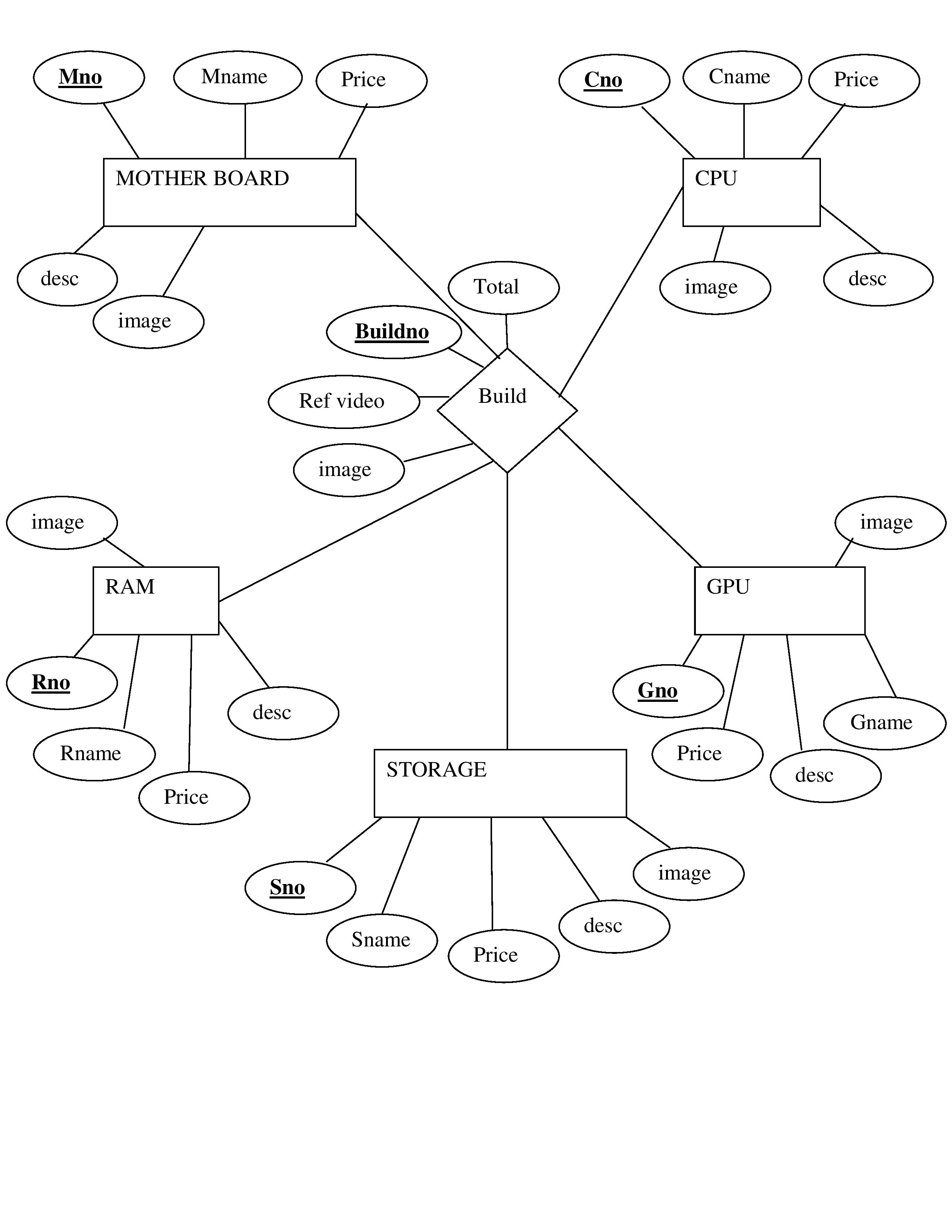
Front End : HTML, CSS, Bootstrap

**Chapter 3**

**DESIGN**

**3.1 ER Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. In other words, ER diagrams illustrate the logical structure of databases.

FIG3.1\_ER DIAGRAM

**3.2 Schema Diagram**

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram shows us the database design.

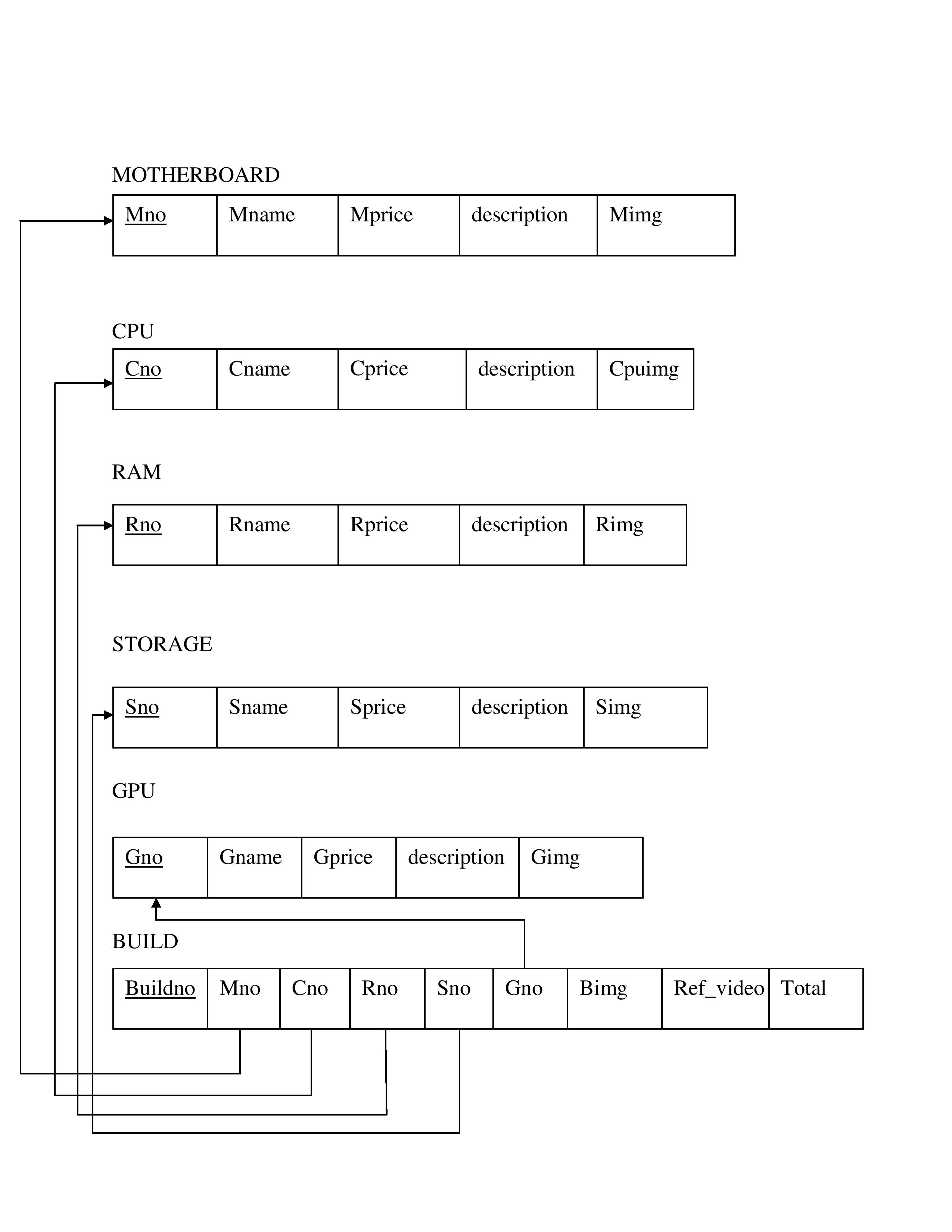


FIG 3.2 SCHEMA DIAGRAM

**Chapter 4**

**IMPLEMENTATION**

**4.1 Introduction**

This project is designed and implemented using MySQL database along with PHP for back-end implementation and HTML and CSS for front-end design. IDE used is Visual Studio Code.

**MySQL**

The back-end of the web application is basically the brains behind the front-end. It comprises of 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 determines how the website functions. In this project MySQL is used as a back-end technology. MySQL is a multi-threaded, multi-user SQL Database Management System. The basic program run as server providing multi-user access to a number of databases. MySQL is currently the world’s most popular and widely used open source database technology and data storage system. MySQL offers great reliability and ease of use. MySQL runs on virtually all platforms, including Linux, UNIX, and Windows.

**Hypertext Preprocessor (PHP)**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general purpose scripting language that is primarily designed for web development and can be embedded into HTML. PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page. PHP commands can be embedded directly into an HTML source document external file to process data or it can be used in combination with various web template systems, web content management systems and web frameworks. It has also evolved to include a command line interface capability and can be used in standalone graphical applications.

A good benefit of using PHP is that it can interact with many different database languages including MySQL. Both PHP and MySQL are compatible with an Apache server which is also free to license. PHP can also run on Windows, Linux and UNIX servers. Due to all these languages being free it is cheap and easy to setup and create a website using PHP. PHP also has very good online documentation with a good framework of. functions in place.

**Hypertext Markup Language (HTML)**

HTML is the web’s core language for creating documents and applications for everyone to use, anywhere. It is standardizing system for tagging text files to achieve font, color, graphic and hyperlink effects on World Wide Web pages. HTML elements form the building blocks of all websites. The markup tells the web browsers how to display web pages. Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of websites. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

**Cascading Style Sheets (CSS)**

CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a simple mechanism for adding style (e.g. fonts, colors, spacing etc.) to web documents. CSS defines how HTML elements are displayed. CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web application and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors and fonts. This separation can improve content accessibility, provide more flexibility. This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods.

**XAMPP**

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.

**4.2 Database Design**

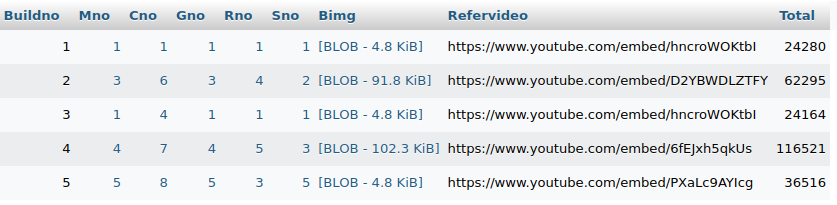
****

Table 4.1 Build Table.

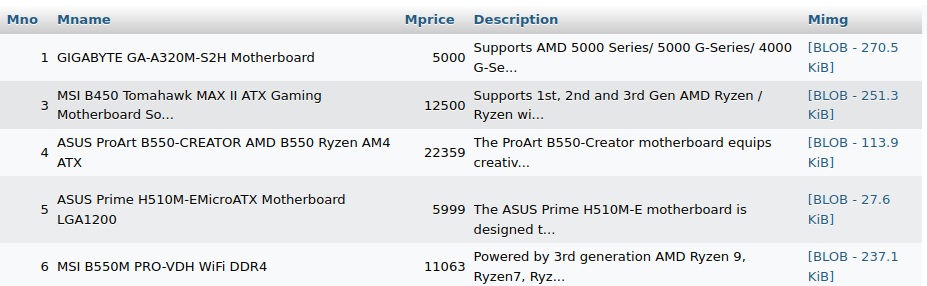


Table 4.2 Motherboard Table.

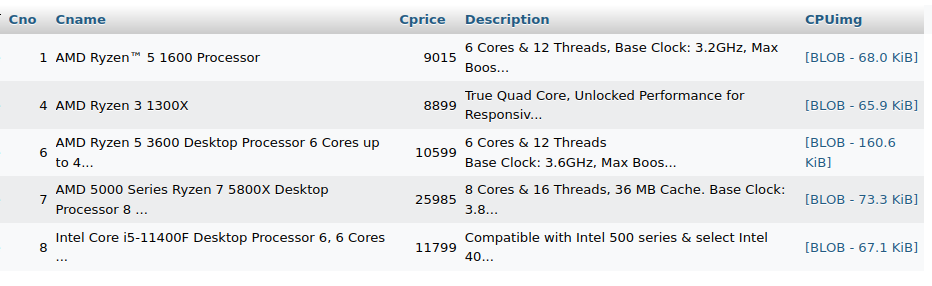


Table 4.3 CPU Table.

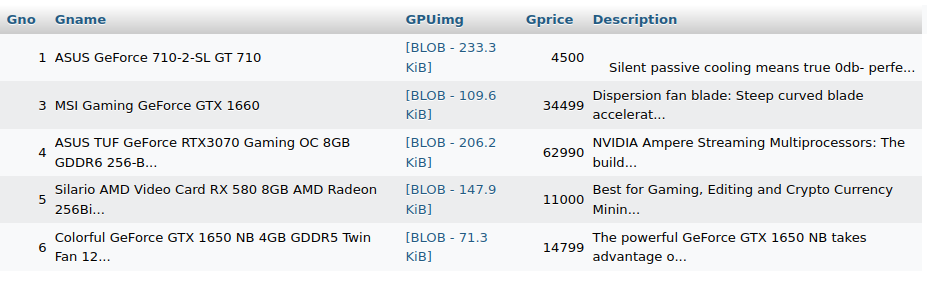


Table 4.4 GPU Table.

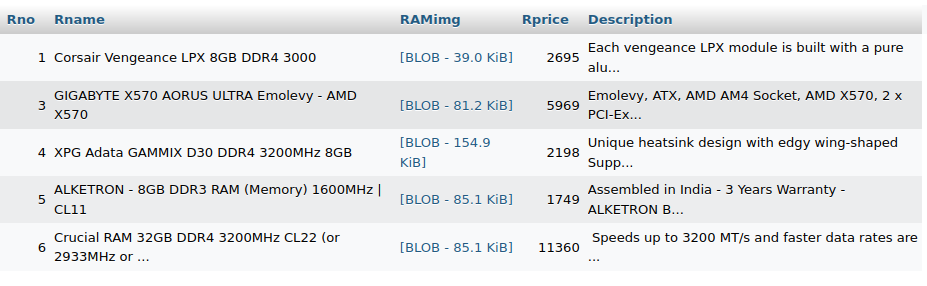


Table 4.5 RAM Table.

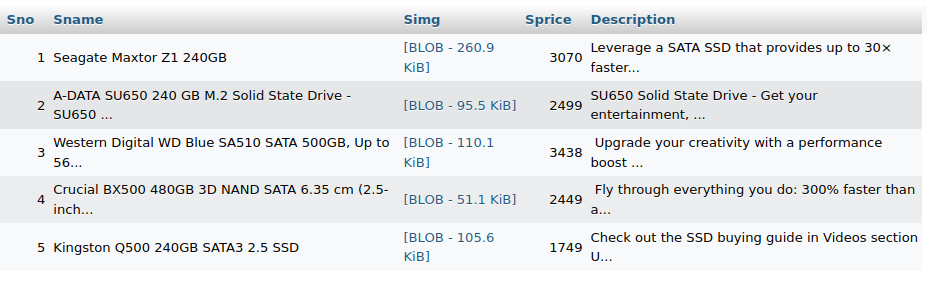


Table 4.6 STORAGE Table.

**4.3 CODE IMPLEMENTATION**

**database Connection**

<?php

$sname= "Localhost";

$name= "root";

$password = "";

$db\_name = "PC\_BUILD";

$conn = new mysqli($sname, $name, $password, $db\_name);

if (!$conn) {

echo 'Could not connect';

}

?>

**CreatIon of table**

CREATE TABLE `BUILD` (

`Buildno` int(11) NOT NULL,

`Mno` int(11) NOT NULL,

`Cno` int(11) NOT NULL,

`Gno` int(11) NOT NULL,

`Rno` int(11) NOT NULL,

`Sno` int(11) NOT NULL,

`Bimg` mediumblob NOT NULL,

`Refervideo` varchar(500) NOT NULL,

`Total` bigint(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `CPU` (

`Cno` int(11) NOT NULL,

`Cname` varchar(500) NOT NULL,

`Cprice` int(11) NOT NULL,

`Description` varchar(500) NOT NULL,

`CPUimg` mediumblob NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `GPU` (

`Gno` int(11) NOT NULL,

`Gname` varchar(500) NOT NULL,

`GPUimg` mediumblob NOT NULL,

`Gprice` int(11) NOT NULL,

`Description` varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `MotherBoard` (

`Mno` int(11) NOT NULL,

`Mname` varchar(100) NOT NULL,

`Mprice` int(11) NOT NULL,

`Description` varchar(500) NOT NULL,

`Mimg` longblob NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `RAM` (

`Rno` int(11) NOT NULL,

`Rname` varchar(500) NOT NULL,

`RAMimg` mediumblob NOT NULL,

`Rprice` int(11) NOT NULL,

`Description` varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `STORAGE` (

`Sno` int(11) NOT NULL,

`Sname` varchar(500) NOT NULL,

`Simg` mediumblob NOT NULL,

`Sprice` int(11) NOT NULL,

`Description` varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**HEADER PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PC BUILD</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+AMvyTG2x" crossorigin="anonymous" />

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/css/bootstrap.min.css" integrity="sha384-r4NyP46KrjDleawBgD5tp8Y7UzmLA05oM1iAEQ17CSuDqnUK2+k9luXQOfXJCJ4I" crossorigin="anonymous">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css" />

<link href="https://api.mapbox.com/mapbox-gl-js/v2.1.1/mapbox-gl.css" rel="stylesheet" />

<link rel="stylesheet" href="style.css" />

<title>PC BUILD</title>

<style>

.extragap {

margin-top: 5.1%;

}

a {

text-decoration: none;

}

.extragap1 {

background-color:black;

margin-top: 5%;

}

p {

color: black;

}

</style>

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg bg-primary navbar-dark py-3 fixed-top">

<div class="container">

<a href="index.php" class="navbar-brand">PC BUILD</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navmenu">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navmenu">

<ul class="navbar-nav ms-auto">

<li class="nav-item">

<a href="index.php" class="nav-link">HOME</a>

</li>

<li class="nav-item">

<a href="product.php" class="nav-link">BUILD</a>

</li>

<li class="nav-item">

<a href="question.php" class="nav-link">QUESTIONS</a>

</li>

<li class="nav-item">

<a href="about.php" class="nav-link">ABOUT US</a>

</li>

</ul>

</div>

</div>

</nav>

**FOOTER PAGE**

<section class="bg-secondary extragap">

</section>

<footer class="p-5 bg-dark text-light text-center position-relative bottom-0 end-0">

<div class="container">

<p class="lead text-light">Copyright &copy; 2022-23 <br> FSD MINI PROJECT</p>

<a href="#" class="position-absolute bottom-0 end-0 p-5">

<i class="bi bi-arrow-up-circle h1"></i>

</a>

</div>

</footer>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9Bn9Plx0x4" crossorigin="anonymous"></script>

</body>

</html>

**HOME PAGE**

<?php

require\_once 'includes/header.php';

?>

<!-- Showcase -->

<section class="bg-dark text-light p-5 p-lg-0 pt-lg-5 text-center text-sm-start">

<div class="container">

<div class="d-sm-flex align-items-center justify-content-between">

<div>

<p class="lead my-4 text-light">

<h3>Want a Better PC? Try Building Your Own.</h3><br>

Assembling a computer yourself is a good way to learn how they work.

</p>

<a href="product.php" class="btn btn-primary btn-lg">CLICK HERE FOR THE LATEST COMBINATIONS!</a>

</div>

<img class="img-fluid w-50 d-none d-sm-block" src="img/pc.jpg" alt="" />

</div>

</div>

</section>

<!--another-->

<section class="bg-secondary text-light p-5">

<div class="container">

<div class="d-md-flex justify-content-between align-items-center">

<h3 class="mb-3 mb-md-0">Latest Offers</h3>

</div>

</div>

</section>

<?php

require\_once 'conn.php';

$sql = 'SELECT \* FROM BUILD';

$result = mysqli\_query($conn, $sql);

?>

<!-- Boxes -->

<section class="p-5">

<div class="container">

<div class="row text-center g-4">

<?php

while ($row = $result->fetch\_array(SQLITE3\_ASSOC)) {

$Buildno = $row['Buildno'];

$Bimg = $row['Bimg'];

$Total = $row['Total'];

?>

<div class="card" style="width: 18rem;">

<!-- <img src="img/pc2.png" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($Bimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Buildno ?></h5>

<p class="card-text"><?php echo $Total ?></p>

<a href="product\_view.php?no=<?php echo $Buildno; ?>" class="btn btn-primary">view</a>

</div>

</div>

<?php } ?>

</div>

</div>

</section>

<!-- MORE Sections -->

<section id="learn" class="p-5" style="background-color:#F4B746;">

<div class="container">

<div class="row align-items-center justify-content-between">

<div class="col-md">

<img src="./img/guidebook.png" class="img" alt="" width="700" height="500" />

</div>

<div class="col-md p-5">

<h2>Guide</h2>

<p class="lead">

Do what you can, with what you have, where you are

</p>

<p>

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Cumque

reiciendis eius autem eveniet mollitia, at asperiores suscipit

quae similique laboriosam iste minus placeat odit velit quos,

nulla architecto amet voluptates?

</p>

<a href="guide.php" class="btn btn-light mt-3">

<i class="bi bi-chevron-right"></i> Read More

</a>

</div>

</div>

</div>

</section>

<?php require\_once 'includes/footer.php'; ?>

**PRODUCT PAGE**

<?php

require\_once 'includes/header.php';

require\_once 'conn.php';

$sql = 'SELECT \* FROM BUILD';

$result = mysqli\_query($conn, $sql);

?>

<div class="extragap"></div>

<section class="bg-dark text-light">

<div class="container">

all our products are certified.

</div>

</section>

<div class="container align-center card-group">

<?php

while ($row = $result->fetch\_array(SQLITE3\_ASSOC)) {

$Buildno = $row['Buildno'];

$Bimg = $row['Bimg'];

$Total = $row['Total'];

?>

<div class="card gap-2" style="width: 18rem;">

<!-- <img src="img/pc.jpg" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" width="30%" height="50%" alt="..." src="data:image/png;base64,' . base64\_encode($Bimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title">BUILD <?php echo $Buildno ?></h5>

<p class="card-text"> <span>&#8377;</span> <?php echo $Total ?></p>

<a href="product\_view.php?no=<?php echo $Buildno; ?>" class="btn btn-primary">view</a>

</div>

</div>

<?php } ?>

</div>

<?php require\_once 'includes/footer.php'; ?>

<?php

require\_once 'includes/header.php';

?>

**MOTHERBOARD PAGE**

<?php

$no = $\_GET['no'];

require\_once 'conn.php';

$sql = "SELECT \* FROM MotherBoard WHERE Mno='$no'";

$result = mysqli\_query($conn, $sql);

while ($row = $result->fetch\_array(SQLITE3\_ASSOC)) {

$Mno = $row['Mno'];

$Mname = $row['Mname'];

$Price = $row['Mprice'];

$Description = $row['Description'];

$Mimg = $row['Mimg'];

// echo '<img class="d-block w-100" alt="..." src="data:image/png;base64,'.base64\_encode($img).'"/>';

}

?>

<div class="extragap"></div>

<section class="bg-dark text-light">

<div class="container">

all our products are certified.

</div>

</section>

<div class="container">

<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">

<div class="carousel-indicators">

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>

<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>

</div>

<div class="carousel-inner">

<div class="carousel-item active">

<!-- <img src="img/mb1.png" class="d-block w-100" alt="..."> -->

<?php echo '<img class="d-block w-100" alt="..." src="data:image/png;base64,' . base64\_encode($Mimg) . '">'; ?>

</div>

<div class="carousel-item">

<!-- <img src="img/mb1.png" class="d-block w-100" alt="..."> -->

<?php echo '<img class="d-block w-100" alt="..." src="data:image/png;base64,' . base64\_encode($Mimg) . '">'; ?>

</div>

<div class="carousel-item">

<!-- <img src="img/mb1.jpg" class="d-block w-100" alt="..."> -->

<?php echo '<img class="d-block w-100" alt="..." src="data:image/png;base64,' . base64\_encode($Mimg) . '">'; ?>

</div>

</div>

<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="visually-hidden">Previous</span>

</button>

<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="visually-hidden">Next</span>

</button>

</div>

</div>

<div class="container">

<h1><?php echo $Mname; ?></h1>

<p><?php echo $Description; ?></p>

<h1><mark> <span>&#8377;</span> <?php echo $Price; ?>/-</mark></h1>

</div>

<?php require\_once 'includes/footer.php'; ?>

**DISPLAY USING SELECT QUERIES**

<?php

require\_once 'includes/header.php';

?>

<style>

.card-img-top {

width: 100%;

height: 15vw;

object-fit: cover;

}

</style>

<?php

$no = $\_GET['no'];

require\_once 'conn.php';

$sql = "SELECT \* FROM BUILD b, CPU c, GPU g, MotherBoard m, RAM r, STORAGE s WHERE b.Mno=m.Mno and b.Cno=c.Cno and b.Gno=g.Gno and b.Rno=r.Rno and b.Sno=s.Sno and Buildno='$no'";

$result = mysqli\_query($conn, $sql);

while ($row = $result->fetch\_array(SQLITE3\_ASSOC)) {

$Buildno = $row['Buildno'];

$Mno = $row['Mno'];

$Cno = $row['Cno'];

$Rno = $row['Rno'];

$Sno = $row['Sno'];

$Gno = $row['Gno'];

$Mimg = $row['Mimg'];

$CPUimg = $row['CPUimg'];

$GPUimg = $row['GPUimg'];

$RAMimg = $row['RAMimg'];

$Simg = $row['Simg'];

$Mname = $row['Mname'];

$Cname = $row['Cname'];

$Gname = $row['Gname'];

$Sname = $row['Sname'];

$Rname = $row['Rname'];

$Mprice = $row['Mprice'];

$Cprice = $row['Cprice'];

$Gprice = $row['Gprice'];

$Rprice = $row['Rprice'];

$Sprice = $row['Sprice'];

$Refervideo = $row['Refervideo'];

$Total = $row['Total'];

}

?>

<div class="extragap"></div>

<div class="card-group">

<div class="card">

<a href="./motherboard.php?no=<?php echo $Mno; ?>">

<!-- <img src="./img/mb1.png" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($Mimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Mname; ?></h5>

<p class="card-text">MotherBoard</p>

<p class="card-text"><small class="text-muted">Price:<?php echo $Mprice; ?></small></p>

</div>

</a>

</div>

<div class="card">

<a href="./cpu.php?no=<?php echo $Cno; ?>">

<!-- <img src="./img/cpu1.jpg" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($CPUimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Cname; ?></h5>

<p class="card-text">CPU</p>

<p class="card-text"><small class="text-muted">Price:<?php echo $Cprice; ?></small></p>

</div>

</a>

</div>

<div class="card">

<a href="./ram.php?no=<?php echo $Rno; ?>">

<!-- <img src="./img/ram1.jpg" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($RAMimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Rname; ?></h5>

<p class="card-text">RAM</p>

<p class="card-text"><small class="text-muted">Price:<?php echo $Rprice; ?></small></p>

</div>

</a>

</div>

<div class="card">

<a href="./gpu.php?no=<?php echo $Gno; ?>">

<!-- <img src="./img/gpu1.jpeg" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($GPUimg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Gname; ?></h5>

<p class="card-text">GPU</p>

<p class="card-text"><small class="text-muted">Price:<?php echo $Gprice; ?></small></p>

</div>

</a>

</div>

<div class="card">

<a href="./storage.php?no=<?php echo $Sno; ?>">

<!-- <img src="./img/ssd1.jpg" class="card-img-top" alt="..."> -->

<?php echo '<img class="card-img-top" alt="..." src="data:image/png;base64,' . base64\_encode($Simg) . '">'; ?>

<div class="card-body">

<h5 class="card-title"><?php echo $Sname; ?></h5>

<p class="card-text">STORAGE</p>

<p class="card-text"><small class="text-muted">Price:<?php echo $Sprice; ?></small></p>

</div>

</a>

</div>

</div>

<div class="container">

<div class="embed-responsive embed-responsive-21by9">

<iframe width="560" height="315" src="<?php echo $Refervideo ?>" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>

</div>

</div>

<?php require\_once 'includes/footer.php'; ?>

**UPDATE QUERIES**

UPDATE BUILD b, CPU c, GPU g, MotherBoard m, RAM r, STORAGE s

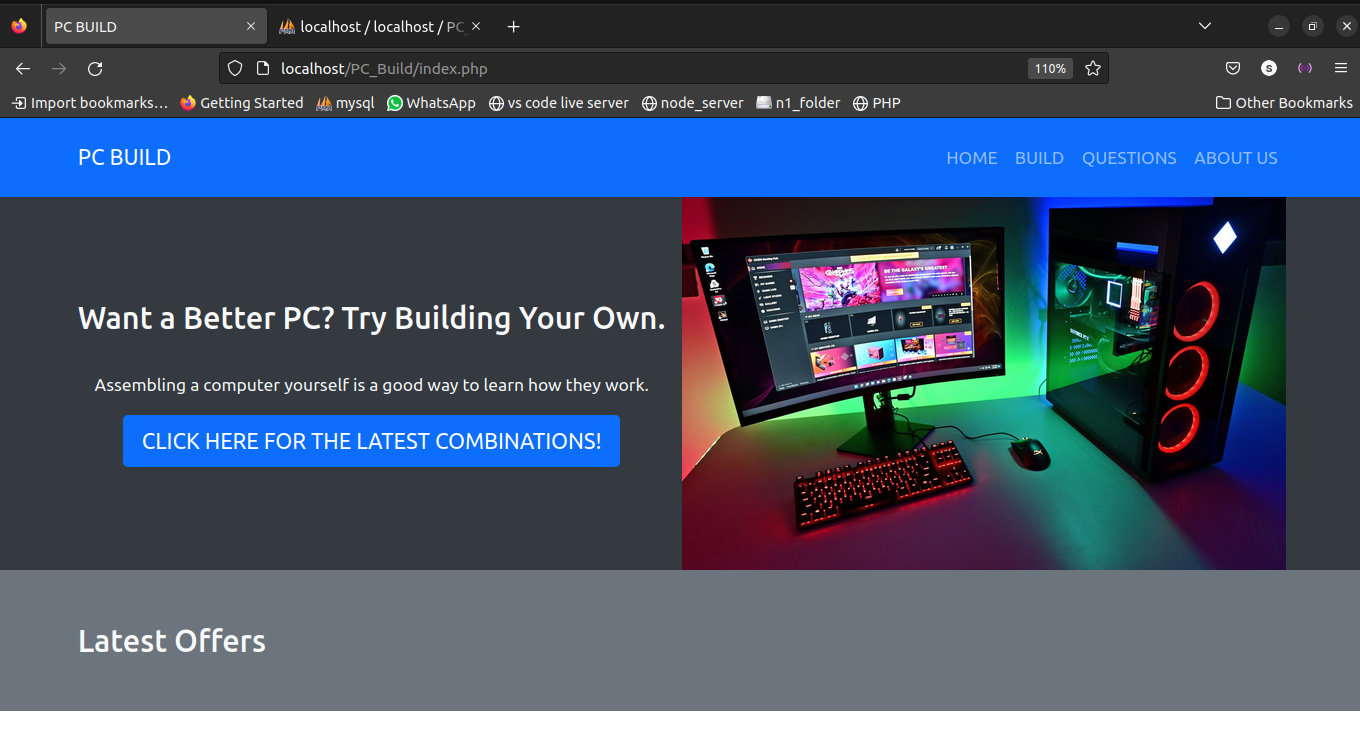
set b.Total=(Mprice + Cprice+ Gprice+ Rprice+ Sprice)

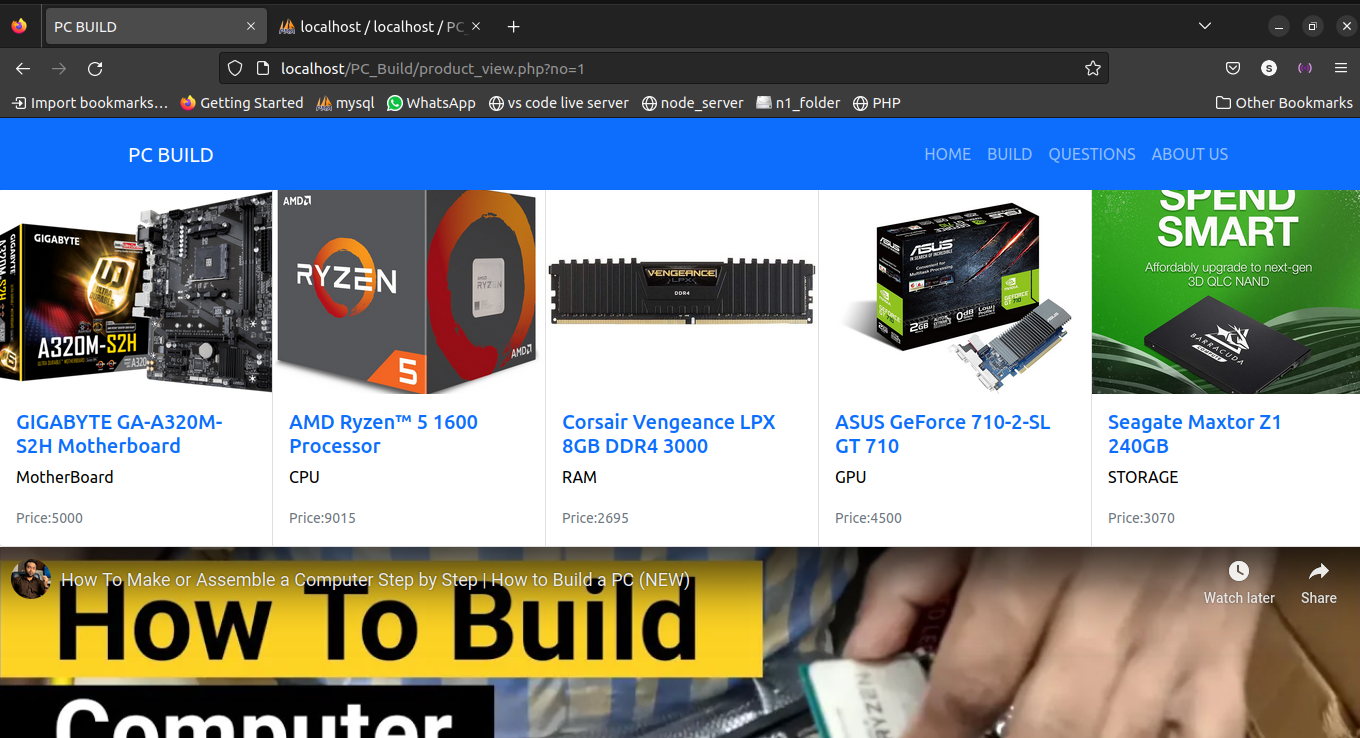
where b.Cno=c.Cno AND b.Mno=m.Mno AND b.Rno=r.Rno AND b.Gno=g.Gno AND b.Sno=s.Sno;

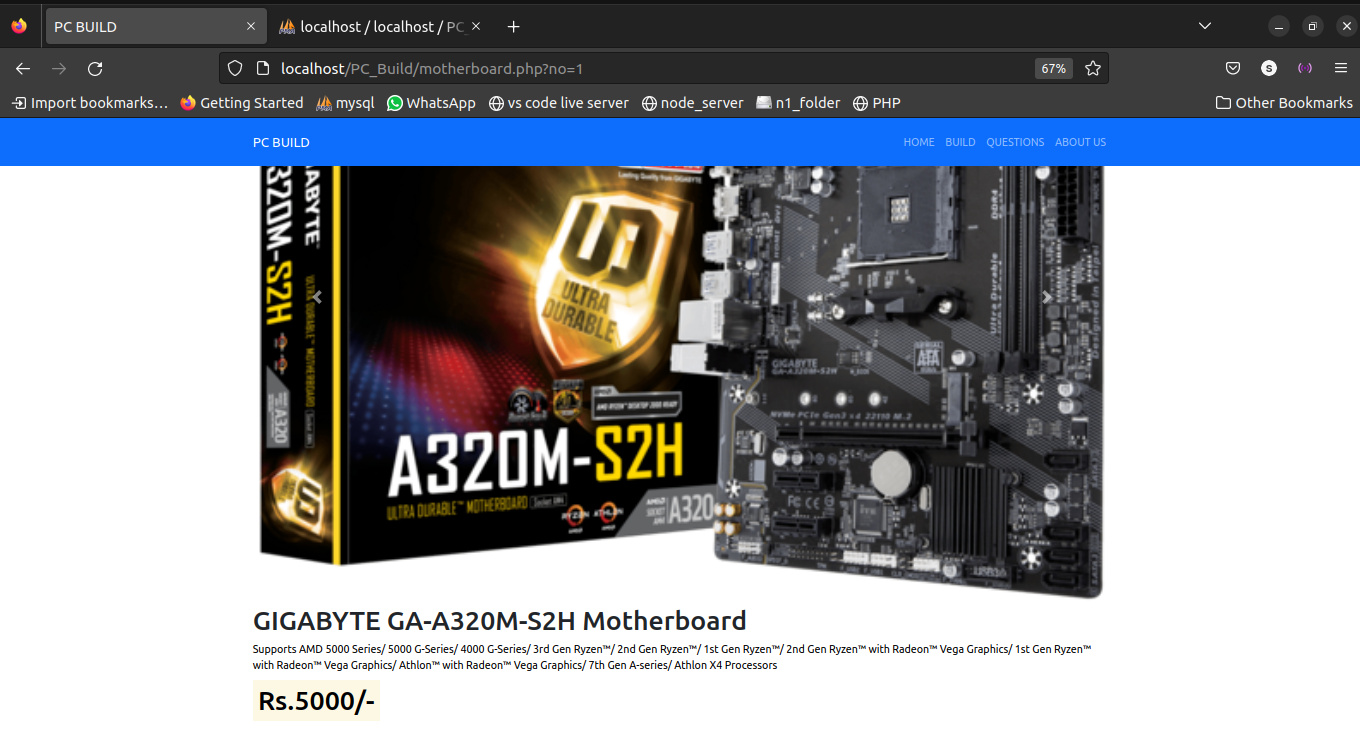
**Chapter 5**

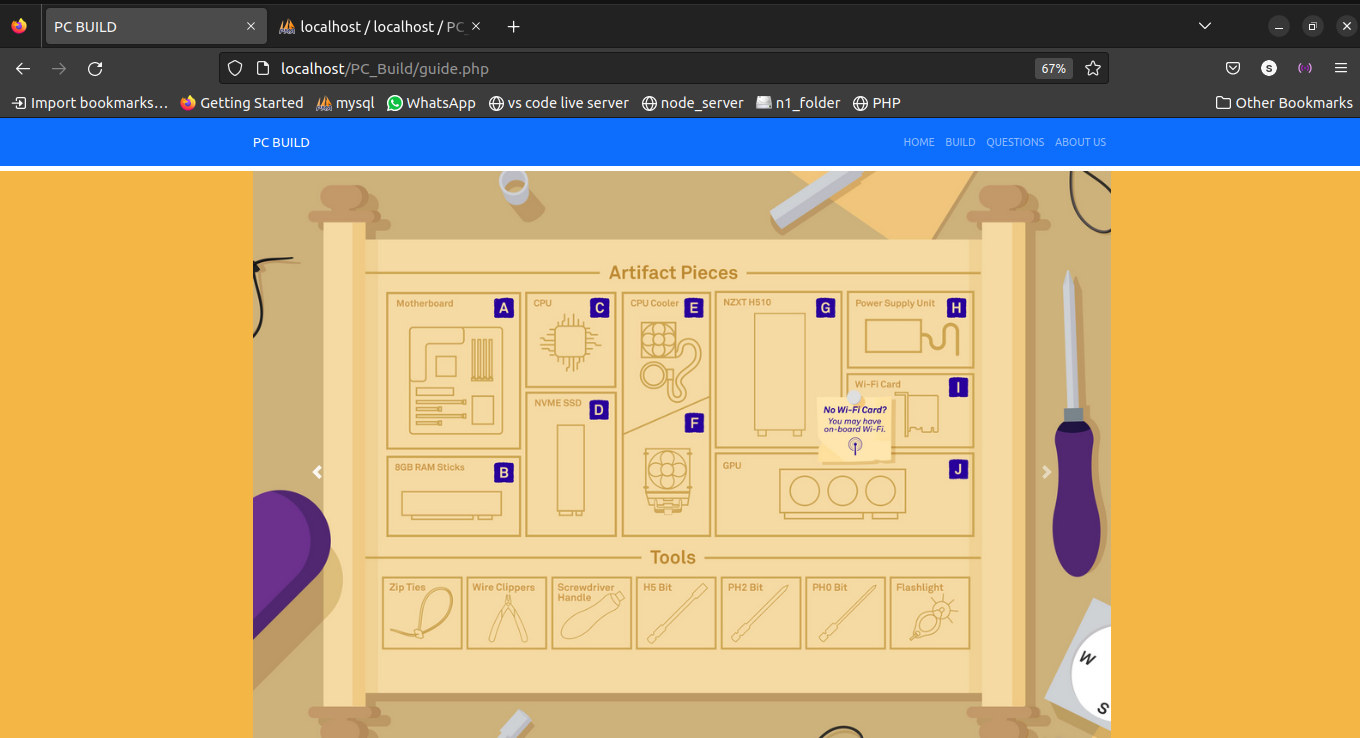
**Interpretation of result**

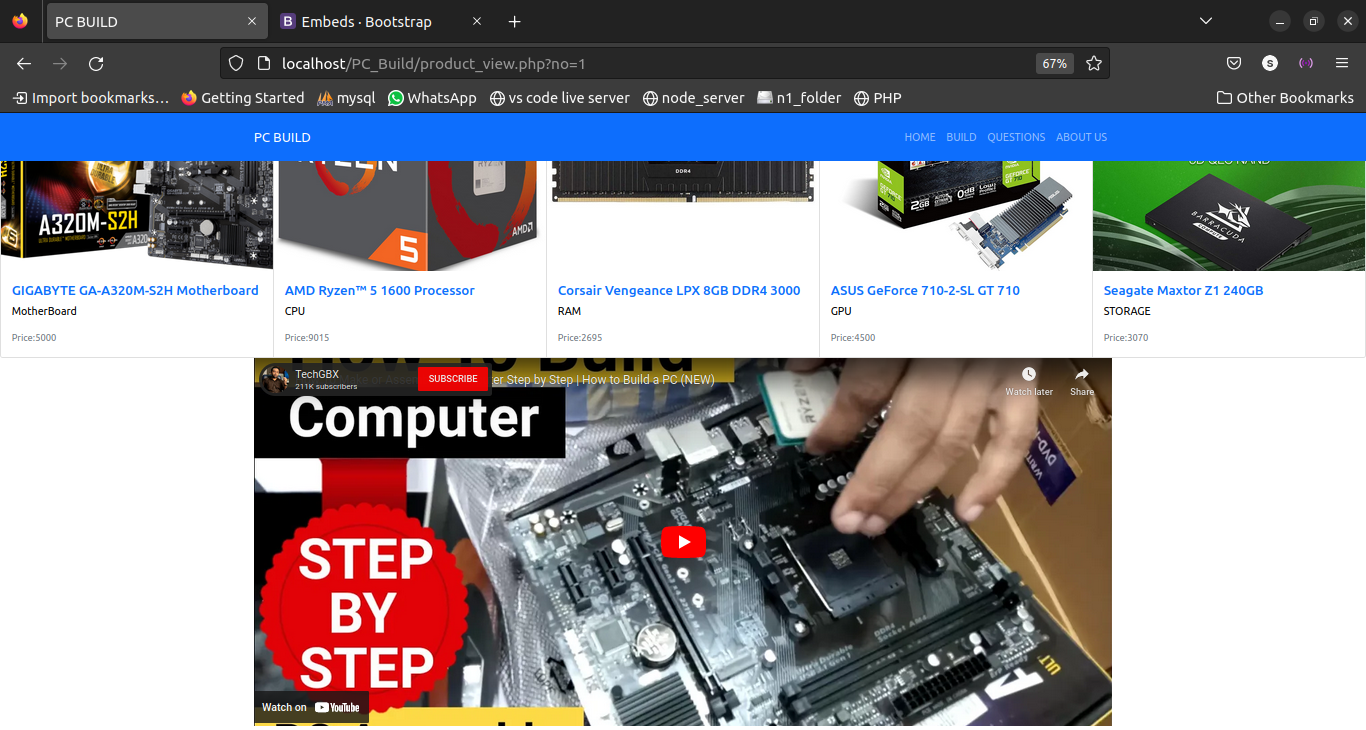
Results:

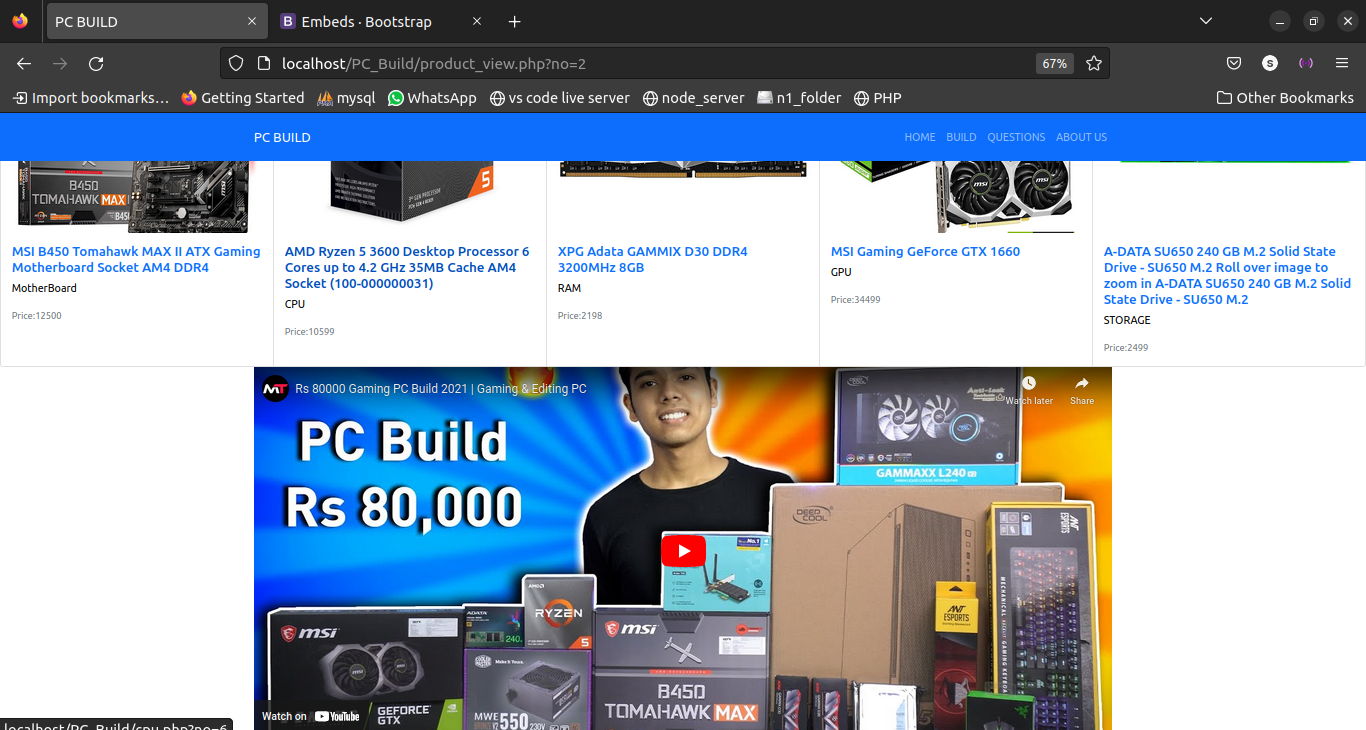
Fig 5.1 HOME Page.

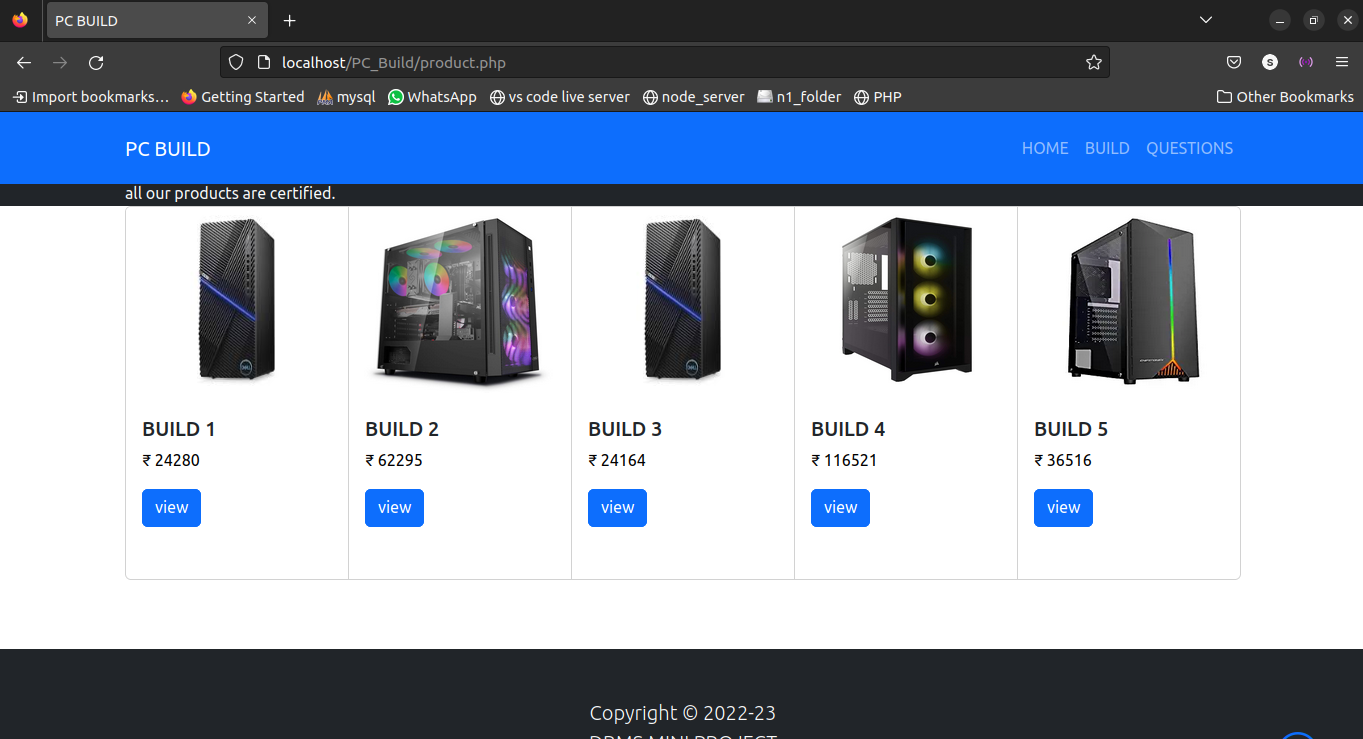
Fig 5.2 Build Page.

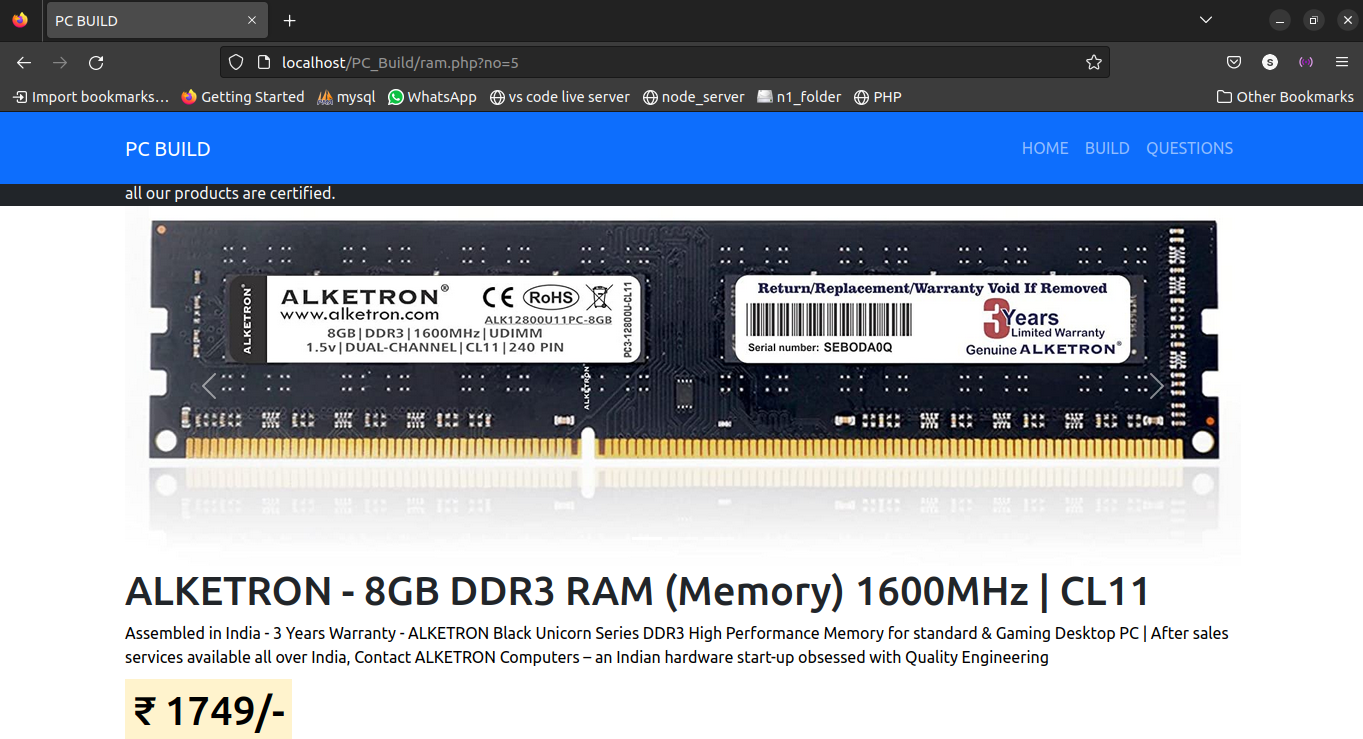
Fig 5.3 Product Details Page.

Fig 5.4 User Guide Page.

Fig 5.5 Build referance video Page.

Fig 5.6 Build referance video Page2 .

Fig 5.7 All Products Page .

Fig 5.8 Ram Page .

**Chapter 6**

**conclusion and future scope**

**conclusion**

Our conclusion to whether building a PC is easy or not is a mixed opinion. I do believe that it is hard to build a PC because a lot of research have said that people should look up books or documents on the web to help them build a computer. Our Project helps them reduce all the effort and time to Build a PC. Our project is the answer for people who want to build a computer with ease otherwise if not, you'll find it hard.

**Future ENHANCEMENT**

Currently, our project only explains about how to build a desktop PC. Our future goal is to make people able to build laptop PC’s with less effort and price.

**REFERENCES**

**TEXTBOOKS**

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson
2. Practical PHP 7, MySQL 8, and MariaDB Website Databases\_ A Simplified Approach to Developing Database-Driven Websites

**LINKS**

1. [mariadb.com](https://mariadb.com/)
2. www.php.net
3. [www.w3schools.com](http://www.w3schools.com/)
4. www.stackoverflow.com