A Project Report

On

”E-Commerce Site”

*Submitted by*

*Badal Chhipa*

*Shivam Chaudhary*

In partial fulfillment for BCA Final Year

*OF*

Bachelor of Computer Application (BCA)

Session 2021-2022

At



ST. WILFRED’S COLLEGE FOR GIRLS

. JAIPUR.

Department of Bachelor of Computer Application

St. Wilfred’s College for girls, Jaipur

University of Rajasthan

A

**Project Report**

On

**“shopping”**

Submitted

In Partial Fulfillment

For the Award of the Degree of

**Bachelor of Computer Applications**

**Session 2021-2022**



**Submitted to: Submitted by:**

*Mr. Shekhar Badal Chhipa*

*Mr. Rohit*  Shivam Chaudhary Assistant Professor B.C.A. final year

Computer Science Faculty

**CERTIFICATE**

This is to certify that the work which is being presented in the project entitled “E-Commerce site” submitted by, Badal and Shivam student of final year BCA during the academic year 2021-2022, in the partial fulfillment for the degree of BECHELOR OF COMPUTER APPLICATION is a record of student’s work carried out by them under my guidance and supervision.

This work has not been submitted elsewhere for the award of any other degree.

Signature of student Signature of Guide Mr. Badal Chhipa Mr. Shekhar Mr. Shivam Chaudhary Mr. Rohit

**ACKNOWLEDGEMENT**

It is mandatory for all the students to give a Project report in final year in any of the relevant Computer Science. First of all I would like to convey my sincere thanks to **Mr. Shekhar and Mr. Rohit**, Assistant Professor ,Computer Science Faculty.

I take this opportunity to express my deep gratitude toward all those people who helped me to complete this project successfully.

I would like to offer my sincere thanks to all those staffs officials for their untiring support and help at various levels.

**ABSTRACT**

“**E-Commerce site**” is a online shopping web site. We provide shop on your doorstep. To make your shopping fast we provides you fast delivery within 2 to 3 days, so you don't get late for your destination. For all IT students and professionals, it is very important to have some basic understanding about the some of the technologies such as C, C++. On this software you can learn the basic and advance concepts of C language. It also contains the logical coding part also for deeper understanding.

**Contents**

**INTRODUCTION**

1. **ANALYSIS**

2.1 SYSTEM ANALYSIS

2.2 SYSTEM SPECIFICATIONS

* + 1. **3. DESIGN APPROACH**
    2. 3.1 INTRODUCTION TO DESIGN

3.2 UML DIAGRAMS

* + 1. 3.3 DATA FLOW DIAGRAMS
    2. 3.4 E-R DIAGRAMS

**4**.  **PROJECT MODULES**

**5. IMPLEMENTATION**

* + 1. 5.1 CONCEPTS AND TECHNIQUES
    2. 5.2 TESTING
    3. 5.2.1 TEST CASES
    4. **6. OUTPUT SCREENS**

**7. CONCLUSION**

**8. FUTURE ENHANCEMENTS**

**INTRODUCTION**

**E-Commerce site** contents provides to focus on providing the web designing language contents to students. In the paper we present techniques that are pertinent to the elements of assessment process: answers submission, computerized grading, and feedback after submission.

As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization.

The Students who are using Event Management can learn php language through this project, thus facilitating effective implementation and monitoring of various activities of C language like coding, testing, debugging . All the necessary topics are provided to the students in this project so that they can easily learn the C language.

**SYSTEM ANALYSIS**

1. **Existing System**

Existing system is a manual one in which various papers, topics are provided to the student so that they can effectively, and easily learn the language. Various programs including pattern programs are also provided.

**Disadvantages:**

The following drawbacks of existing system emphasize the need for computerization:

1. A lot of topics have to be gathered.

2. A lot of correction work hence delay in giving the results.

3. A lot of tabulation work for each topic.

1. **Proposed System**

This application is used to provide knowledge about C . The students can sit at individual terminals and login to learn the language . The topics are presented to the students. This application will perform opening of various topics, providing code to the logical problems such as Fibonacci series, palindrome etc.It also contains various types of modal papers and last year papers also.

1. **Objective of the System**

The objective of the BKS NEWSis to provide better information for the users of this system for better results for their career in future and their increasing knowledge about the ongoing technologies in the market.

**SYSTEM SPECIFICATIONS**

Hardware Requirements (Minimum):-

* Pentium - IV(Processor).
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB
* Microsoft Compatible 101 or more Key Board

**Software Requirements: -**

* **Operating System :** Windows
* **Technology:** Web Based Application
* **Front-End:** HTML,JAVA SCRIPT,CSS, php
* **Back-End: MY-**SQL

**INTRODUCTION TO DESIGN**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

UML Diagrams:

Actor: A coherent set of roles that users of use cases play when interacting with the use cases.

Use case: A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

They are as follows:

* Use case Diagram
* Sequence Diagram
* Collaboration Diagram
* Activity Diagram
* State chat Diagram

**USECASE DIAGRAMS:**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor.Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

**DATA FLOW DIAGRAMS**:

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

**RULES FOR DFD:**

* Fix the scope of the system by means of context diagrams.
* Organize the DFD so that the main sequence of the actions
* Reads left to right and top to bottom.
* Identify all inputs and outputs.
* Identify and label each process internal to the system with Rounded circles.
* A process is required for all the data transformation and Transfers. Therefore, never connect a data store to a data Source or the destinations or another data store with just a Data flow arrow.
* Do not indicate hardware and ignore control information.
* Make sure the names of the processes accurately convey everything the process is done.
* There must not be unnamed process.
* Indicate external sources and destinations of the data, withSquares.
* Number each occurrence of repeated external entities.
* Identify all data flows for each process step, except simple Record retrievals.
* Label data flow on each arrow.
* Use details flow on each arrow.
* Use the details flow arrow to indicate data movements.

**DATAFLOW DIAGRAMS:**

Database:

HOW IT WORKS

CART

ABOUT US

**E-R DIAGRAMS:**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design For the database designer, the utility of the ER model is:

* it maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* it is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

ER Notation

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

* **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
* **Relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
* **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
* **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

**Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

**MODULES**

**OVERVIEW OF TECHNOLOGIES USED**

**JAVA**

The new features and upgrades included into Java changed the face of programming environment and gave a new definition to [Object Oriented Programming](https://en.wikibooks.org/wiki/Object_Oriented_Programming) (*OOP* in short). But unlike its predecessors, Java needed to be bundled with standard functionality and be independent of the host platform.

The primary goals in the creation of the Java language:

* It is simple.
* It is object-oriented.
* It is independent of the host platform.
* It contains language facilities and libraries for networking.
* It is designed to execute code from remote sources securely.

The Java language introduces some new features that did not exist in other languages like C and C++.

Object orientation[[edit](https://en.wikibooks.org/w/index.php?title=Java_Programming/Java_Overview&action=edit&section=1)]

[](https://commons.wikimedia.org/wiki/File:KP-UML-Generalization-20060325.svg)

Object oriented programming can be represented with [UML](https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/UML) diagrams.

Object orientation ("OO") refers to a method of programming and language technique. The main idea of OO is to design software around the "things" (i.e. objects) it manipulates, rather than the actions it performs.

As the hardware of the computer advanced, it brought about the need to create better software techniques to be able to create ever increasing complex applications. The intent is to make large software projects easier to manage, thus improving quality and reducing the number of failed projects. Object oriented solution is the latest software technique.

**FEASIBILITY STUDY:**

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

* 1. Technical Feasibility
  2. Economical Feasibility
  3. Operational Feasibility.

**1.Technical Feasibility**

The project entitles "Courier Service System” is technically feasibility because of the below mentioned feature. The project was developed in Java which Graphical User Interface.

It provides the high level of reliability, availability and compatibility. All these makeJava an appropriate language for this project. Thus the existing software Java is a powerful language.

1. **Economical Feasibility**

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

3. **Operational Feasibility**

In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessaries.

**Implementation:**

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving.

confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

**MAINTENANCE AND ENVIRONMENT:**

AS the number of computer based systems, grieve libraries of computer software began to expand. In house developed projects produced tones of thousand soft program source statements. Software products purchased from the outside added hundreds of thousands of new statements. A dark cloud appeared on the horizon. All of these programs, all of those source statements-had to be corrected when false were detected, modified as user requirements changed, or adapted to new hardware that was purchased. These activities were collectively called software Maintenance.

The maintenance phase focuses on change that is associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements. Four types of changes are encountered during the maintenance phase.

## SOFTWARE METHODOLOGY

The software methodology followed in this project includes the object-oriented methodology and the application system development methodologies. The description of these methodologies is given below.

**Application System Development – A Life cycle Approach**

Although there are a growing number of applications (such as decision support systems) that should be developed using an experimental process strategy such as prototyping, a significant amount of new development work continue to involve major operational applications of broad scope. The application systems are large highly structured. User task comprehension and developer task proficiency is usually high. These factors suggest a linear or iterative assurance strategy. The most common method for this stage class of problems is a system development life cycle modal in which each stage of development is well defined and has straightforward requirements for deliverables, feedback and sign off. The system development life cycle is described in detail since it continues to be an appropriate methodology for a significant part of new development work.

The basic idea of the system development life cycle is that there is a well-defined process by which an application is conceived and developed and implemented. The life cycle gives structure to a creative process. In order to manage and control the development effort, it is necessary to know what should have been done, what has been done, and what has yet to be accomplished. The phrases in the system development life cycle provide a basis for management and control because they define segments of the flow of work, which can be identified.

The phases in the life cycle for information system development are described differently by different writers, but the differences are primarily in the amount of necessity and manner of categorization. There is a general agreement on the flow of development steps and the necessity for control procedures at each stage.

The information system development cycle for an application consists of three major stages.

1) Definition.

2) Development.

3) Installation and operation.

The first stage of the process, which defines the information requirements for a feasible cost effective system. The requirements are then translated into a physical system of forms, procedures, programs etc., by the system design, computer programming and procedure development. The resulting system is test and put into operation. No system is perfect so there is always a need for maintenance changes. To complete the cycle, there should be a post audit of the system to evaluate how well it performs and how well it meets the cost and performance specifications. The stages of definition, development and installation and operation can therefore be divided into smaller steps or phrases as follows.

**TESTING**

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

**Testing Objectives:**

1. Testing is a process of executing a program with the intent of finding an error

2. A good test case is one that has a probability of finding an as yet undiscovered error

3. A successful test is one that uncovers an undiscovered error

**Testing Principles**

1. All tests should be traceable to end user requirements
2. Tests should be planned long before testing begins
3. Testing should begin on a small scale and progress towards testing in large
4. Exhaustive testing is not possible
5. To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used.

They are

White box testing.

Black box testing.

# **White-box testing:**

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

# **Block-box testing:**

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

**Unit testing:**

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors with in the boundary of the modules. These tests were carried out during the programming stage itself. All units of ViennaSQL were successfully tested.

**Integration testing :**

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

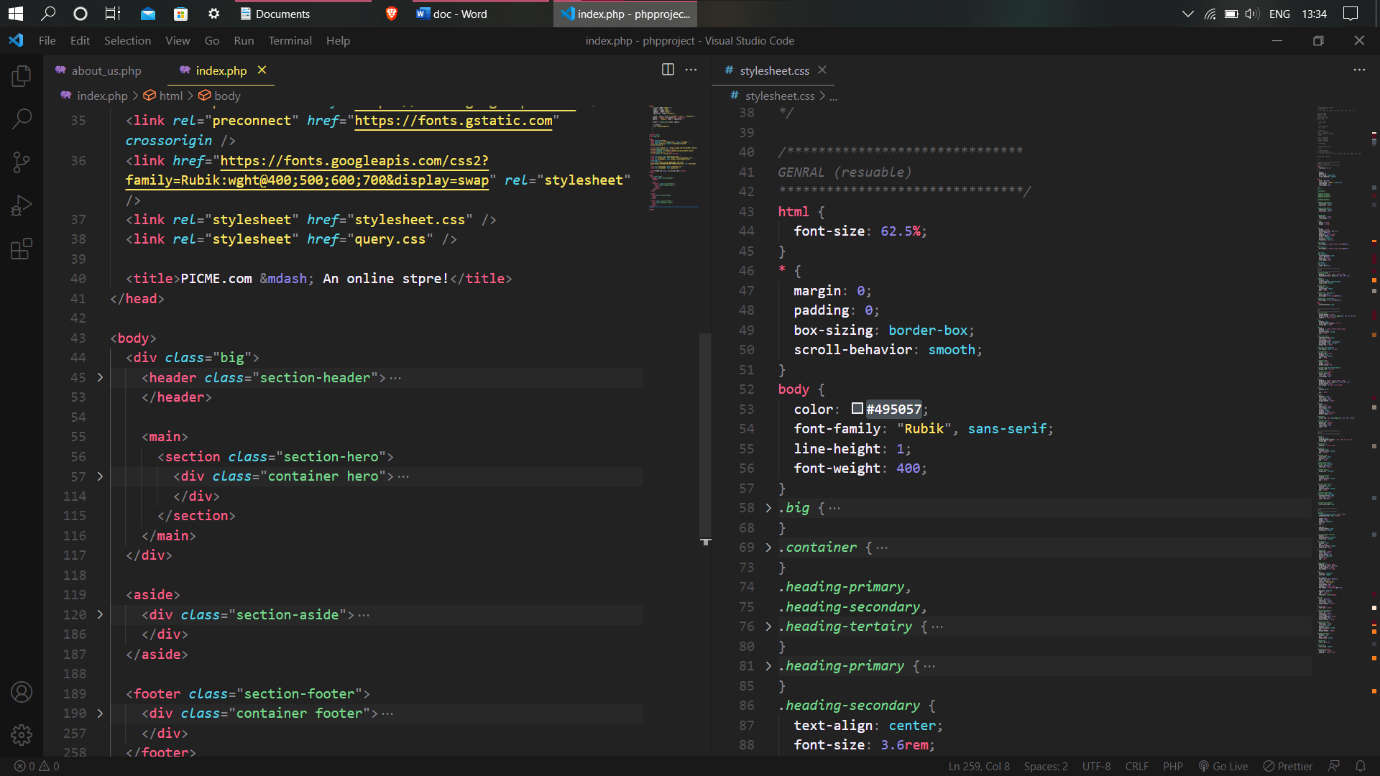
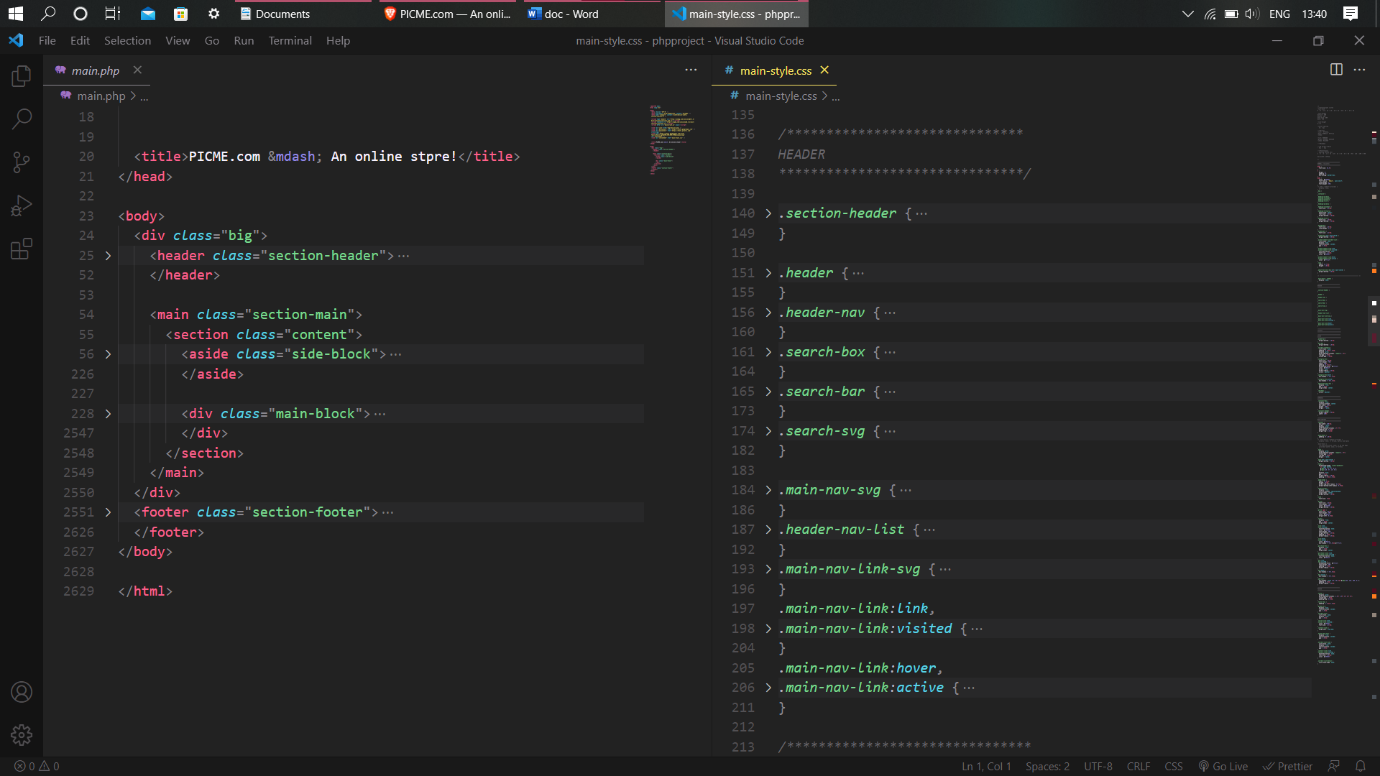
**System testing:**

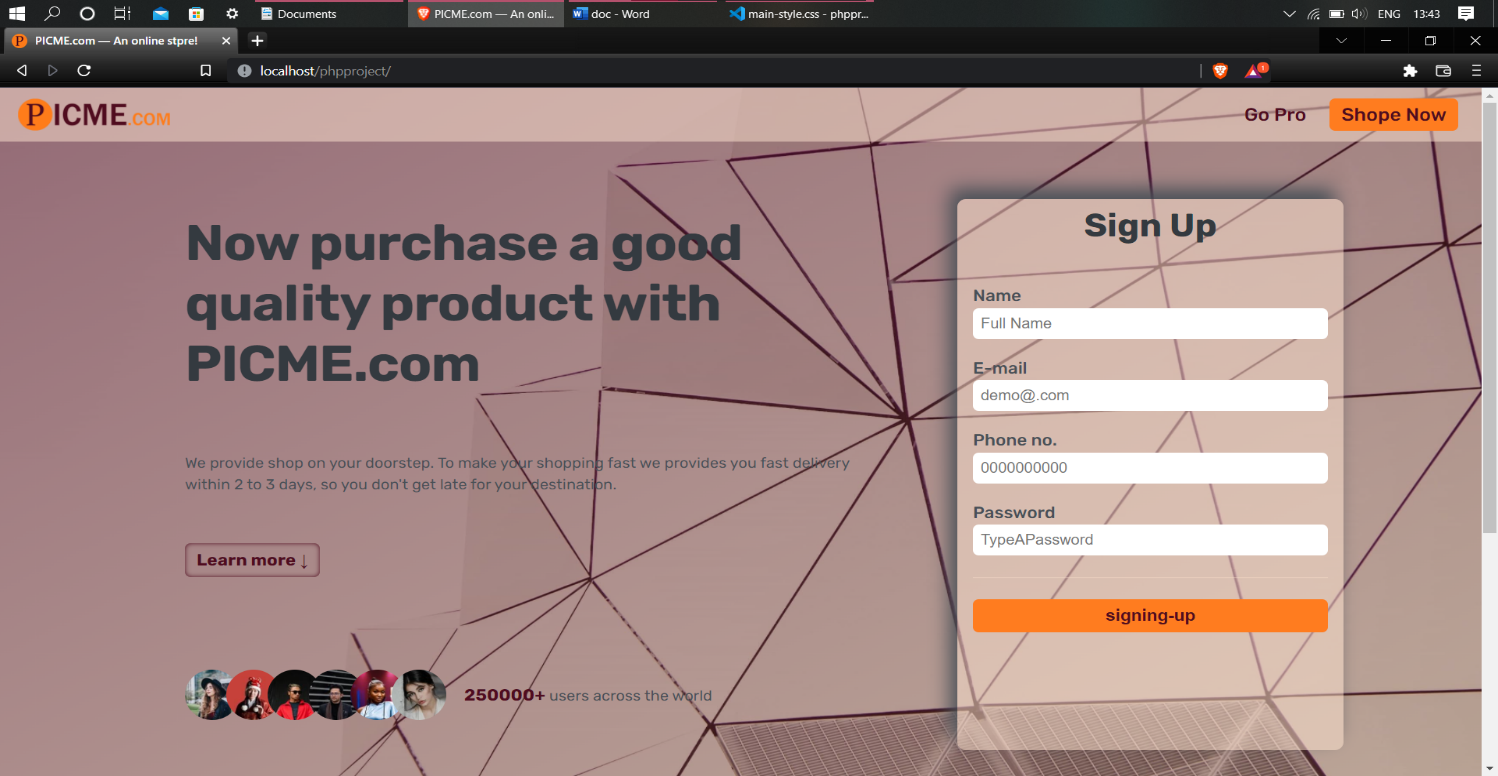
System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and it’s original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here, and specified path ODBC connection will correct or not, and giving output or not are tested here these verifications and validations are done by giving input values to the system and by comparing with expected output.

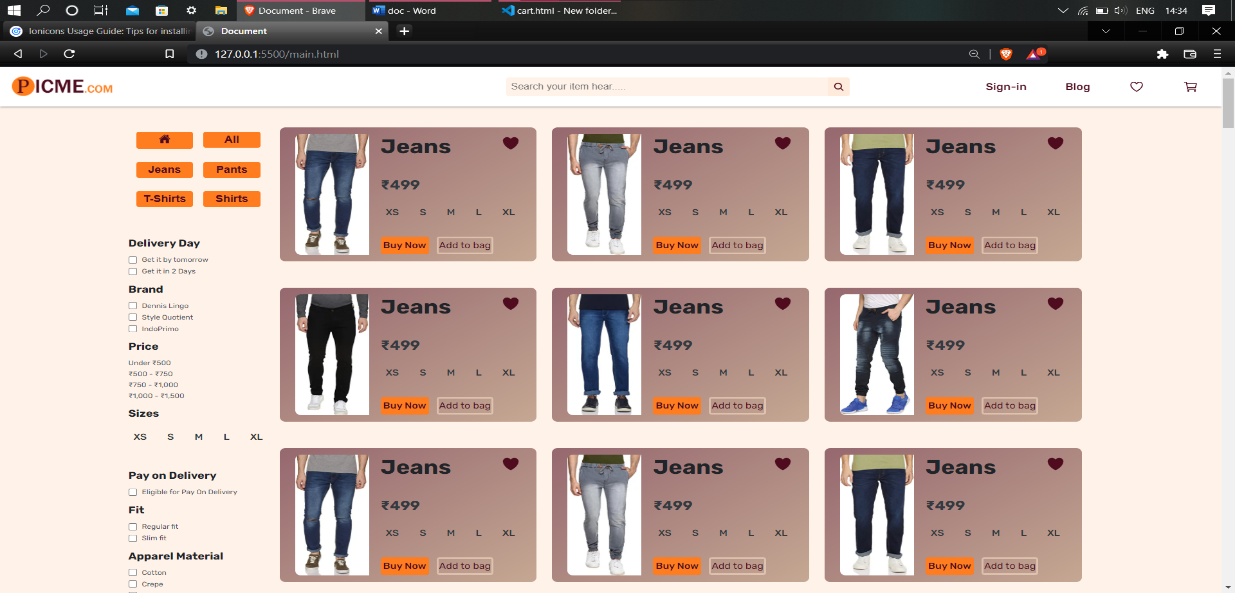
## Acceptance Testing:

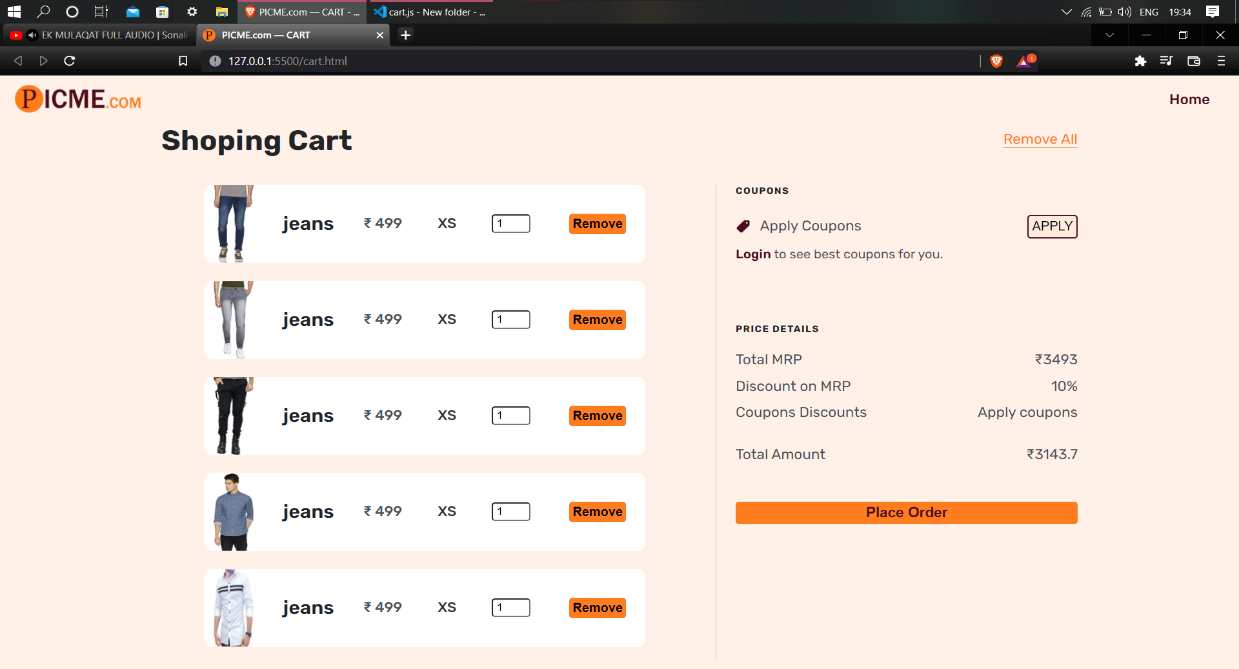
This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

**CODING**

1. **Index Page**
2. **Home Page**

**Web site look**

****

****

**CONCLUSION**

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

* + Automation of the entire system improves the efficiency
  + It provides a friendly graphical user interface which proves to be better when compared to the existing system.
  + It gives appropriate access to the authorized users depending on their permissions.
  + It effectively overcomes the delay in communications.
  + Updating of information becomes so easier.
  + System security, data security and reliability are the striking features.
  + The System has adequate scope for modification in future if it is necessary.

**FUTURE ENHANCEMENTS**

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the different scheduled examinations information that are currently issued.

Well I and my team members have worked hard in order to present an improved website better than the existing one’s regarding the information about the various activities. Still, we found out that the project can be done in a better way. Primarily, when we request information about a particular schedules it just shows the exam date and platform. So, after getting the information we can get access to the online exam.

The enhancement that we can add the searching option. We can directly search to the particular student details from this site.