# SHOPPING CART USING CRYPTOGRAPHY

## Synopsis:

An online shopping using cryptography system that permits a customer to submit online orders for items and/or services from a store to online customers. The online shopping system presents an online display of an order cutoff time and an associated delivery window for items selected by the customer. The system accepts the customer's submission of a purchase order for the item in response to a time of submission being before the order cutoff time. The online shopping system does not settle with a credit supplier of the customer until the item selected by the customer is picked from inventory but before it is delivered. Therefore, the customer can go online and make changes to the order. In addition, available service windows are presented to the customer as a function of customer selected order and service types; and further, the order picking is assigned in accordance with a picker's preference. in this web site also covered customer billing convert to cryptography method.

|  |  |  |
| --- | --- | --- |
| CHAPTER | **CONTENTS** | PAGE NO |
| 1 | **INTRODUCTION** |  |
|  | INTRODUCTION | 9 |
|  | AIMS AND OBJECTIVE | 10 |
|  | EXISTING SYSTEM | 10 |
|  | DRAWBACKS | 10 |
|  | PROPOSED SYSTEM | 11 |
| 2 | **SYSTEM STUDY** |  |
|  | HARDWARE SPECTIFICATION | 12 |
|  | SOFTWARE SPECIFICATION | 12 |
| 3 | **MODULE DESCRIPTION** |  |
|  | ITEM MODULE | 13 |
|  | ORDER MODULE | 13 |
|  | BILL MODULE | 13 |
|  | ENCRYPTION MODULE | 13 |
|  | DECRYPTION MODULE | 14 |
|  | PAYMENT MODULE | 14 |

|  |  |  |  |
| --- | --- | --- | --- |
| **4** | **SOFTWARE DESCRIPTION** |  |  |
|  | PHP PROGRAMMING | 15 |
|  | MYSQL | 30 |
| **5** | **SYSTEM DESIGN AND DEVELOPMENT** |  |
|  | INPUT DESIGN | 35 |
|  | OUTPUT DESIGN | 35 |
|  | DATABASE DESIGN | 36 |
| **6** | **TESTING AND IMPLEMENTATION** |  |
|  | UNIT TESTING | 37 |
|  | INTEGRATION TESTING | 37 |
|  | VALIDATION TESTING | 38 |
|  | SYSTEM TESTING | 38 |
| **7** | **CONCLUSION** |  |  |
|  | FUTURE WORK | 43 |
|  | BIBILOGRAPHY | 43 |
|  | SAMPLE SCREENSHOTS | 44 |
|  | SAMPLE CODING | 59 |

## SHOPPING CART USING CRYPTOGRAPHY CHAPTER-1

**INTRODUCTION**

Shopping cart is an internal business function responsible for managing documents from vendors or suppliers. Invoices represent a source document in accounting. Source documents outline specific information relating to a financial transaction. Vendors usually provide outside services such as maintenance, utilities, office supplies, or advertising to businesses. Economic resources include the raw materials and direct labor needed to produce consumer goods or services. Companies use Shopping cart processes to ensure vendors and suppliers receive payment in a timely manner

## AIM AND OBJECTIVES

* A common Shopping cart process is the three-way match system found in accounts payable functions.
* Purchase orders represent the company’s internal information that provides individuals specific authorization to purchase items from vendors or suppliers.
* Two copies of purchase orders usually exist in a business. One is sent to the vendor or supplier and one is retained in the company’s accounting department.
* This document comes into play later on during the Shopping cart process.
* In this project the customer will see the products in online and they will confirm to buy the products in online.
* The admin will view all the customer details and their ordered products details and they will send the details by encryption format.
* The customer will see the product details using decryption format.
* Company may scan this information into a server database.

## EXISTING SYSTEM

Currently the company is maintaining their data in clipper 4.0 a data base management system tool. Sales order, quotation generation, stock maintenance is running as separate modules and the data are not integrated with each other. Billing is done with printing sheets using MS - word readymade alignment. As the System is totally maintained in clipper, various problems were come across by the data entry user as well as by the manager in case of report generation.

## DRAWBACKS

* + At a time only two tables can be linked.
  + Number of table opened by clipper is only 8.
  + Data concurrency is not possible.
  + As all the data are stored in physical format data security is very poor.
  + Maintaining multi user is very difficult and in the case it is totally impossible.
  + Unauthorized users can enter into the System.
  + Presentation of data can be done only in character mode.
  + Data integrity is very tedious to be maintained in clipper.

## PROPOSED SYSTEM

Computerization of Shopping cart for business is parts are a project having the functionality of sales products. Goods are also purchase from local manufacturers. The proposed system will start the process through online. Every customer of this entire world can see the products details and this will increase the profit details in a growth way of company

## ADVANTAGES

* Sends receipt to customer by encrypting.
* Provides security to users.
* User is provided with various options in products.
* MD5 algorithm implementation makes the system more secure
* Time saving for users.

## CHAPTER 2

**SYSTEM STUDY**

## Hardware specification:

Processor Type : Pentium IV Speed : 2.4GHZ

Ram : 2GB RAM

Hard disk : 120 GB HD

## Software specification:

Operating System : windows 7.

Front End : PHP

Back End : MY SQL

## CHAPTER 3

**MODULE DESCRIPTION**

## Item module:

In this module the details of every product will display through online. Every viewer will visit this website and order by using this module. Admin can also add the product details to their website. The displayed item details will store the item table in admin side database.

## Order module:

This module will include about the customer details and their order details about the product. Customer can add the product item to their cart. They will mention the quantity of product and the delivery date product details mentioned here. Ordered item details save to admin or company database.

## Bill module:

In this module details about the customer ordered item bill details will show by the admin. The ordered item price will pay by the customer either cash payment or card payment. This procedure will store by the admin side. The cash payment procedure will display t customer side.

## Encryption module:

This module to encrypt the data for customer ordered item details. Admin view the ordered item it is ordered by the customer. The product details will send by encryption format by the admin. Encryption format will store to database. Admin will send the details to the customer in an encryption format.

## Decryption module:

This module process is customer will view the details about in an encryption way. Using secret key customer will decrypt the data and will view the details. By cryptographic method every data will secure. Details of data will store in database.

## Payment module:

Payment module will say about the cash payment details by the customer. Customer will pay the amount either cash payment or card payment by this procedure will tell by this module. If he pay the cash or card payment details of amount will store by the admin side.

## CHAPTER 4

**SOFTWRAE DESCRIPTION** **PHP PROGRAMMING**

**PHP** is a [scripting](http://en.wikipedia.org/wiki/Scripting_programming_language) language designed to fill the gap between [SSI](http://en.wikipedia.org/wiki/Server_Side_Includes) (Server Side Includes) and [Perl](http://en.wikibooks.org/wiki/Perl), intended for the Web environment. Its principal application is the implementation of Web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C- like syntax, and its simplicity. The newest version of PHP is 5.5 and it is heavily recommended to always use the newest version for better security, performance and of course features.

If you've been to a website that prompts you to login, you've probably encountered a server-side scripting language. Due to its market saturation, this means you've probably come across PHP. [PHP](http://en.wikipedia.org/wiki/PHP) was designed by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) to display his resume online and to collect data from his visitors.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "**P**HP: **H**ypertext **P**reprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

Before you embark on the wonderful journey of Server Side Processing, it is recommended that you have a basic understanding of the [HyperText Markup Language (HTML)](http://en.wikibooks.org/wiki/HTML). But [PHP](http://en.wikipedia.org/wiki/PHP) can also be used to build [GUI](http://en.wikipedia.org/wiki/GUI)-driven applications for example by using [PHP-GTK](http://en.wikipedia.org/wiki/PHP-GTK).

**PHP** is a [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](http://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](http://en.wikipedia.org/wiki/General-purpose_programming_language). As of January 2013, PHP was installed on more than 240 million [websites](http://en.wikipedia.org/wiki/Website) (39% of those sampled) and 2.1 million [web servers](http://en.wikipedia.org/wiki/Web_server). Originally created by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994, the [reference implementation](http://en.wikipedia.org/wiki/Reference_implementation) of PHP (powered by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine)) is

now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Preprocessor*, which is a [recursive](http://en.wikipedia.org/wiki/Recursive_acronym) [backronym](http://en.wikipedia.org/wiki/Backronym).

PHP code can be simply mixed with [HTML](http://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [templating engines](http://en.wikipedia.org/wiki/Web_template_system) and [web frameworks](http://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29), which is usually implemented as a web server's native [module](http://en.wikipedia.org/wiki/Plugin_%28computing%29) or a [Common](http://en.wikipedia.org/wiki/Common_Gateway_Interface) Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) (CLI) capability and can be used in [standalone](http://en.wikipedia.org/wiki/Computer_software) [graphical applications](http://en.wikipedia.org/wiki/Graphical_user_interface).

The canonical PHP interpreter, powered by the Zend Engine, is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Computing_platform), free of charge.

Despite its popularity, no written [specification](http://en.wikipedia.org/wiki/Formal_specification) or standard existed for the PHP language until 2014, leaving the canonical PHP interpreter as a [de facto](http://en.wikipedia.org/wiki/De_facto) standard. Since 2014, there is ongoing work on creating a formal PHP specification.

PHP development began in 1994 when [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) wrote a series of [Common](http://en.wikipedia.org/wiki/Common_Gateway_Interface) Gateway Interface (CGI) binaries in C, which he used to maintain his [personal homepage](http://en.wikipedia.org/wiki/Personal_homepage). He extended them to add the ability to work with [web forms](http://en.wikipedia.org/wiki/Web_form) and to communicate with [databases](http://en.wikipedia.org/wiki/Database), and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could be used to build simple, dynamic [web applications](http://en.wikipedia.org/wiki/Web_application). Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" publicly to accelerate [bug](http://en.wikipedia.org/wiki/Software_bug) location and improve the code, on the [Usenet](http://en.wikipedia.org/wiki/Usenet) discussion group *comp.infosystems.*[*www.authoring.cgi*](http://www.authoring.cgi/)on June 8, 1995. This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The [syntax](http://en.wikipedia.org/wiki/Syntax) resembled that of Perl but was simpler, more limited and less consistent[.](http://en.wikipedia.org/wiki/PHP)

Early PHP was not intended to be a new programming language, and grew organically, with Lerdorf noting in retrospect: "I don’t know how to stop it, there was never any intent to write a programming language […] I have absolutely no idea how to write a programming

language, I just kept adding the next logical step on the way." A development team began to form and, after months of work and [beta](http://en.wikipedia.org/wiki/Beta_development_stage) testing, officially released PHP/FI 2 in November 1997. One criticism of PHP is that it was not originally designed, but instead it was developed organically; among other things, this has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower- level libraries which PHP was "wrapping", while in some very early versions of PHP the length of the function names was used internally as a [hash function](http://en.wikipedia.org/wiki/Hash_function), so names were chosen to improve

the distribution of hash values.

[Zeev Suraski](http://en.wikipedia.org/wiki/Zeev_Suraski) and [Andi Gutmans](http://en.wikipedia.org/wiki/Andi_Gutmans) rewrote the [parser](http://en.wikipedia.org/wiki/Parser) in 1997 and formed the base of PHP 3, changing the language's name to the [recursive acronym](http://en.wikipedia.org/wiki/Recursive_acronym) *PHP: Hypertext Preprocessor*. Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new [rewrite](http://en.wikipedia.org/wiki/Rewrite_%28programming%29) of PHP's core, producing the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine) in 1999. They also founded [Zend Technologies](http://en.wikipedia.org/wiki/Zend_Technologies) in [Ramat Gan](http://en.wikipedia.org/wiki/Ramat_Gan), Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released. As of August 2008 this branch reached version 4.4.9. PHP 4 is no longer under development nor will any security updates be released.

On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. In 2008 PHP 5 became the only stable version under development. [Late static binding](http://en.wikipedia.org/wiki/Late_static_binding) had been missing from PHP and was added in version 5.3.

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

Over time, PHP interpreters became available on most existing [32-bit](http://en.wikipedia.org/wiki/32-bit) and [64-bit](http://en.wikipedia.org/wiki/64-bit) operating systems, either by building them from the PHP source code, or by using pre-built binaries. For the PHP versions 5.3 and 5.4, the only available [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) binary distributions were 32-bit [x86](http://en.wikipedia.org/wiki/X86) builds, requiring Windows 32-bit compatibility mode while using [Internet Information Services](http://en.wikipedia.org/wiki/Internet_Information_Services) (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64- bit [x86-64](http://en.wikipedia.org/wiki/X86-64) builds available for Microsoft Windows.

## PHP 6 and Unicode

PHP received mixed reviews due to lacking native [Unicode](http://en.wikipedia.org/wiki/Unicode) support at the core language level. In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the [International Components for Unicode](http://en.wikipedia.org/wiki/International_Components_for_Unicode) (ICU) library, and representing text strings as [UTF-16](http://en.wikipedia.org/wiki/UTF-16) internally. Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.

However, a shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF-16, which is rarely used in a web context, led to delays in the project. As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features back-ported from PHP 6, notably namespaces. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6, such as traits and closure re- binding. Initial hopes were that a new plan would be formed for Unicode integration, but as of 2014 none has been adopted.

### PHP 7

As of 2014, work is underway on a new major PHP version named PHP 7. There was some dispute as to whether the next major version of PHP was to be called PHP 6 or PHP 7. While the PHP 6 unicode experiment had never been released, a number of articles and book titles referenced the old PHP 6 name, which might have caused confusion if a new release were to reuse the PHP 6 name. After a vote, the name PHP 7 was chosen.

PHP 7 gets its foundations from an experimental PHP [branch](http://en.wikipedia.org/wiki/Branching_%28revision_control%29) that was originally named *phpng* (*PHP next generation*), which aims at optimizing PHP performance by refactoring the Zend Engine while retaining near-complete language compatibility. As of 14 July 2014, [WordPress](http://en.wikipedia.org/wiki/WordPress)-based benchmarks, which serve as the main benchmark suite for phpng project, show an almost 100% increase in performance. Changes from phpng are also expected to make it easier to improve performance in the future, as more compact data structures and other changes are seen as better suited for a successful migration to a [just-in-time](http://en.wikipedia.org/wiki/Just-in-time_compilation) (JIT) compiler. Because of

the significant changes, this reworked Zend Engine will be called *Zend Engine 3*, succeeding the Zend Engine 2 used in PHP 5.

In terms of new language features, PHP 7 will add features such as return type declarations[,](http://en.wikipedia.org/wiki/PHP) which will complement its existing parameter type declarations. PHP 7 will also contain an improved variable syntax which is internally consistent and complete, resolving a long-standing issue in PHP, what will allow use of ->, [], (), {}, and :: operators with arbitrary meaningful left-hand-side expressions.

### Syntax

The following [Hello world program](http://en.wikipedia.org/wiki/Hello_world_program) is written in PHP code embedded in an [HTML](http://en.wikipedia.org/wiki/HTML) document:

<!DOCTYPE html>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php echo '<p>Hello World</p>'; ?>

</body>

</html>

However, as PHP does not need to be embedded in HTML or used with a web server, the simplest version of a Hello World program can be written like this, with the closing tag omitted as preferred in files containing pure PHP code (prior to PHP 5.4.0, this short syntax for echo() only works with the short\_open\_tag configuration setting enabled, while for PHP 5.4.0 and later it is always available):

<?= 'Hello world';

The PHP interpreter only executes PHP code within its [delimiters](http://en.wikipedia.org/wiki/Delimiter). Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to [control structures](http://en.wikipedia.org/wiki/Control_structure) described in PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <? or <?= (which is used to echo back a [string](http://en.wikipedia.org/wiki/String_%28computer_science%29) or [variable](http://en.wikipedia.org/wiki/Variable_%28programming%29)) and ?> as well as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages)-style short forms <% or <%= and %>. Short delimiters make script files less portable, since support for them can be disabled in the local PHP configuration, and they are therefore

discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, <?php and ?>, in [XHTML](http://en.wikipedia.org/wiki/XHTML) and other [XML](http://en.wikipedia.org/wiki/XML) documents, creates correctly formed XML "processing instructions". This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a [dollar symbol](http://en.wikipedia.org/wiki/Dollar_sign), and a [type](http://en.wikipedia.org/wiki/Primitive_type) does not need to be specified in advance. PHP 5 introduced *type hinting* that allows functions to force their parameters to be objects of a specific class, arrays, interfaces or [callback functions](http://en.wikipedia.org/wiki/Callback_function). However, type hints can not be used with scalar types such as integer or string.

Unlike function and class names, variable names are case sensitive. Both double-quoted ("") and [heredoc](http://en.wikipedia.org/wiki/Heredoc) strings provide the ability to interpolate a variable's value into the string. PHP treats [newlines](http://en.wikipedia.org/wiki/Newline) as [whitespace](http://en.wikipedia.org/wiki/Whitespace_character) in the manner of a [free-form language](http://en.wikipedia.org/wiki/Free-form_language), and statements are terminated by a semicolon. PHP has three types of [comment syntax](http://en.wikipedia.org/wiki/Comparison_of_programming_languages_%28syntax%29): /\* \*/ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text, *e.g.*, to a web browser.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. if conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, C#, Java and Perl.

### Data types

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit [signed](http://en.wikipedia.org/wiki/Signed_number_representations) [integer](http://en.wikipedia.org/wiki/Integer_%28computer_science%29) equivalent to the [C-language long type](http://en.wikipedia.org/wiki/C_variable_types_and_declarations). Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), [octal](http://en.wikipedia.org/wiki/Octal), [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal), and [binary](http://en.wikipedia.org/wiki/Binary_code) notations.

[Floating point](http://en.wikipedia.org/wiki/Floating_point) numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of [scientific notation](http://en.wikipedia.org/wiki/Scientific_notation). PHP has a native [Boolean](http://en.wikipedia.org/wiki/Boolean_datatype) type that is similar to the native Boolean types in [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in [Perl](http://en.wikipedia.org/wiki/Perl) and C++.

The null data type represents a variable that has no value; *NULL* is the only allowed value for this data type.

Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources.

Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in [hashes](http://en.wikipedia.org/wiki/Hash_table) with both keys and values, and the two can be intermingled. PHP also supports [strings](http://en.wikipedia.org/wiki/String_%28computing%29), which can be used with single quotes, double quotes, nowdoc or [heredoc](http://en.wikipedia.org/wiki/Heredoc) syntax.

The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

## Functions

PHP has hundreds of functions provided by the core language functionality and thousands more available via various extensions; these functions are well documented in the online PHP documentation. However, the built-in library has a wide variety of naming conventions and associated inconsistencies, as described under [history](http://en.wikipedia.org/wiki/PHP) above.

Additional functions can be defined by the developer:

function myAge($birthYear) // defines a function, this one is named "myAge"

{

$yearsOld = date('Y') - $birthYear; // calculates the age

return $yearsOld . ' year' . ($yearsOld != 1 ? 's' : ''); // returns the age in a descriptive form

}

echo 'I am currently ' . myAge(1981) . ' old.'; // outputs the text concatenated

// with the return value of myAge()

// As the result of this syntax, myAge() is called.

// In 2014, the output of this sample program will be 'I am currently 33 years old.'

In PHP, normal functions are not [first-class](http://en.wikipedia.org/wiki/First-class_function) and can only be referenced by their name directly, or dynamically by a variable containing the name of the function (referred to as "variable functions"). User-defined functions can be created at any time without being [prototyped](http://en.wikipedia.org/wiki/Function_prototype). Functions can be defined inside code blocks, permitting a [run-time decision](http://en.wikipedia.org/wiki/Dynamic_dispatch) as to whether or not a function should be defined. Function calls must use parentheses, with the

exception of zero-argument class [constructor](http://en.wikipedia.org/wiki/Constructor_%28computer_science%29) functions called with the PHP new operator, where parentheses are optional.

Until PHP 5.3, support for true [anonymous functions](http://en.wikipedia.org/wiki/Anonymous_functions) or [closures](http://en.wikipedia.org/wiki/Closure_%28computer_science%29) did not exist in PHP. While create\_function() exists since PHP 4.0.1, it is merely a thin wrapper around eval() that allows normal PHP functions to be created during program execution[.](http://en.wikipedia.org/wiki/PHP) Also, support for variable functions allows normal PHP functions to be used, for example, as [callbacks](http://en.wikipedia.org/wiki/Callback_function) or within [function tables](http://en.wikipedia.org/wiki/Function_table)[.](http://en.wikipedia.org/wiki/PHP) PHP 5.3 added support for closures, which are true anonymous, first-class functions, whose syntax can be seen in the following example:

function getAdder($x)

{

return function($y) use ($x)

{

return $x + $y;

};

}

$adder = getAdder(8);

echo $adder(2); // prints "10"

In the example above, getAdder() function creates a closure using passed argument $x (the keyword use imports a variable from the lexical context), which takes an additional argument $y, and returns the created closure to the caller. Such a function is a first-class object, meaning that it can be stored in a variable, passed as a parameter to other functions, etc.

The [goto](http://en.wikipedia.org/wiki/Goto) flow control statement is used as in the following example: function lock()

{

$file = fopen('file.txt', 'r+');

retry:

if (!flock($file, LOCK\_EX | LOCK\_NB)) goto retry;

fwrite($file, 'Success!'); fclose($file);

}

When flock() is called, PHP opens a file and tries to lock it. The target label retry: defines the point to which execution should return if flock() is unsuccessful and goto retry; is called. The goto statement is restricted and requires that the target label be in the same file and context. The goto statement has been supported since PHP 5.3.

## Objects

Basic [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) functionality was added in PHP 3 and improved in PHP 4. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like [value types](http://en.wikipedia.org/wiki/Value_type)[.](http://en.wikipedia.org/wiki/PHP) The drawback of this method was that the whole object was copied when a variable was assigned or passed as a parameter to a method. In the new approach, objects are referenced by [handle](http://en.wikipedia.org/wiki/Handle_%28computing%29), and not by value.

PHP 5 introduced private and protected [member variables](http://en.wikipedia.org/wiki/Member_variable) and methods, along with [abstract classes](http://en.wikipedia.org/wiki/Abstract_type), [final classes](http://en.wikipedia.org/wiki/Final_type), [abstract methods](http://en.wikipedia.org/wiki/Abstract_method), and [final methods](http://en.wikipedia.org/wiki/Final_method). It also introduced a standard way of declaring [constructors](http://en.wikipedia.org/wiki/Constructor_%28computer_science%29) and [destructors](http://en.wikipedia.org/wiki/Destructor_%28computer_science%29), similar to that of other object-oriented languages such as [C++](http://en.wikipedia.org/wiki/C%2B%2B), and a standard [exception handling](http://en.wikipedia.org/wiki/Exception_handling) model. Furthermore, PHP 5 added [interfaces](http://en.wikipedia.org/wiki/Interface_%28computing%29) and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. [Objects](http://en.wikipedia.org/wiki/Object_%28computer_science%29) implementing ArrayAccess can be used with [array](http://en.wikipedia.org/wiki/Array_data_type) syntax and objects implementing [Iterator](http://en.wikipedia.org/wiki/Iterator) or [IteratorAggregate](http://en.wikipedia.org/wiki/IteratorAggregate) can be used with the foreach [language construct](http://en.wikipedia.org/wiki/Language_construct). There is no [virtual table](http://en.wikipedia.org/wiki/Virtual_table) feature in the engine, so [static variables](http://en.wikipedia.org/wiki/Static_variable) are bound with a name instead of a reference at compile time.

If the developer creates a copy of an object using the reserved word clone, the Zend engine will check whether a clone() method has been defined. If not, it will call a default

clone() which will copy the object's properties. If a clone() method is defined, then it will be responsible for setting the necessary properties in the created object. For convenience, the engine will supply a function that imports the properties of the source object, so the programmer can start with a by-value [replica](http://en.wiktionary.org/wiki/replica) of the source object and only override properties that need to be changed.

The following is a basic example of [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) in PHP: class Person

{

public $firstName; public $lastName;

public function construct($firstName, $lastName = '') { // optional second argument

$this->firstName = $firstName;

$this->lastName = $lastName;

}

public function greet() {

return 'Hello, my name is ' . $this->firstName .

(($this->lastName != '') ? (' ' . $this->lastName) : '') . '.';

}

public static function staticGreet($firstName, $lastName) { return 'Hello, my name is ' . $firstName . ' ' . $lastName . '.';

}

}

$he = new Person('John', 'Smith');

$she = new Person('Sally', 'Davis');

$other = new Person('iAmine');

echo $he->greet(); // prints "Hello, my name is John Smith." echo '<br />';

echo $she->greet(); // prints "Hello, my name is Sally Davis." echo '<br />';

echo $other->greet(); // prints "Hello, my name is iAmine." echo '<br />';

echo Person::staticGreet('Jane', 'Doe'); // prints "Hello, my name is Jane Doe."

The [visibility](http://en.wikipedia.org/wiki/Visibility_%28computer_science%29) of PHP properties and methods is defined using the [keywords](http://en.wikipedia.org/wiki/Keyword_%28computer_programming%29) public, private, and protected. The default is public, if only [var](http://en.wikipedia.org/wiki/Variable_%28programming%29) is used; var is a synonym for public. Items declared public can be accessed everywhere. protected limits access to [inherited classes](http://en.wikipedia.org/wiki/Inherited_class) (and to the class that defines the item). private limits visibility only to the class that defines the item. Objects of the same type have access to each other's private and protected members even though they are not the same instance. PHP's member visibility features have sometimes been described as "highly useful." However, they have also sometimes been described as "at best irrelevant and at worst positively harmful."

## Implementations

The original, only complete and most widely used PHP implementation is powered by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine) and known simply as PHP. To disambiguate it from other implementations, it is sometimes unofficially referred to as "Zend PHP". The Zend Engine [compiles](http://en.wikipedia.org/wiki/Compiler) PHP [source code](http://en.wikipedia.org/wiki/Source_code) on-the-fly into an internal format that it can execute, thus it works as an [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29). It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP itself. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

PHP's single-request-per-script-execution model, and the fact the Zend Engine is an interpreter, lead to inefficiency. As a result, various products have been developed to help improve PHP performance. In order to speed up execution time and not have to compile the PHP source code every time the web page is accessed, PHP scripts can also be deployed in the PHP engine's internal format by using an [opcode](http://en.wikipedia.org/wiki/Opcode) cache, which works by [caching](http://en.wikipedia.org/wiki/Cache_%28computing%29) the compiled form of a PHP script (opcodes) in [shared memory](http://en.wikipedia.org/wiki/Shared_memory) to avoid the overhead of [parsing](http://en.wikipedia.org/wiki/Parsing) and [compiling](http://en.wikipedia.org/wiki/Compiling) the code every time the script runs. An opcode cache, [Zend Opcache](http://en.wikipedia.org/wiki/Zend_Opcache), is built into PHP since version

5.5. Another example of a widely used opcode cache is the [Alternative PHP Cache](http://en.wikipedia.org/wiki/Alternative_PHP_Cache) (APC), which is available as a [PECL](http://en.wikipedia.org/wiki/PHP_Extension_Community_Library) extension.

While Zend PHP is still the most popular implementation, several other implementations have been developed. Some of these are [compilers](http://en.wikipedia.org/wiki/Compiler) or support [JIT compilation](http://en.wikipedia.org/wiki/JIT_compilation), and hence offer performance benefits over Zend PHP at the expense of lacking full PHP compatibility. Alternative implementations include the following:

* [HipHop Virtual Machine](http://en.wikipedia.org/wiki/HipHop_Virtual_Machine) (HHVM) – developed at Facebook and available as open source, it converts PHP code into a high-level bytecode (commonly known as an [intermediate language](http://en.wikipedia.org/wiki/Intermediate_language)), which is then translated into x86-64 machine code dynamically at runtime by a [just-in-time](http://en.wikipedia.org/wiki/Just-in-time_compiler) (JIT) compiler, resulting in up to 6× performance improvements.
* [Parrot](http://en.wikipedia.org/wiki/Parrot_virtual_machine) – a virtual machine designed to run dynamic languages efficiently; Pipp transforms the PHP source code into the [Parrot intermediate representation](http://en.wikipedia.org/wiki/Parrot_intermediate_representation), which is then translated into the Parrot's bytecode and executed by the virtual machine.
* [Phalanger](http://en.wikipedia.org/wiki/Phalanger_%28compiler%29) – compiles PHP into [Common Intermediate Language](http://en.wikipedia.org/wiki/Common_Intermediate_Language) (CIL) bytecode
* [HipHop](http://en.wikipedia.org/wiki/HipHop_for_PHP) – developed at Facebook and available as open source, it transforms the PHP scripts into [C++](http://en.wikipedia.org/wiki/C%2B%2B) code and then compiles the resulting code, reducing the server load up to 50%. In early 2013, Facebook deprecated it in favor of HHVM due to multiple reasons, including deployment difficulties and lack of support for the whole PHP language, including the create\_function() and eval() constructs.

## Licensing

PHP is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License), which stipulates that:

Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from [group@php.net.](mailto:group@php.net) You may indicate that your software works in conjunction with PHP by saying "[Foo](http://en.wikipedia.org/wiki/Foo) for PHP" instead of calling it "PHP Foo" or "phpfoo".

This restriction on use of the name *PHP* makes the PHP License incompatible with the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL), while the Zend License is incompatible due to an advertising clause similar to that of the original license of BSD.

## Development and community

PHP includes various [free and open-source libraries](http://en.wikipedia.org/wiki/List_of_PHP_libraries) in its source distribution, or uses them in resulting PHP binary builds. PHP is fundamentally an [Internet](http://en.wikipedia.org/wiki/Internet)-aware system with built- in modules for accessing [File Transfer Protocol](http://en.wikipedia.org/wiki/File_Transfer_Protocol) (FTP) servers and many database servers, including [PostgreSQL](http://en.wikipedia.org/wiki/PostgreSQL), [MySQL](http://en.wikipedia.org/wiki/MySQL), [Microsoft SQL Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server) and [SQLite](http://en.wikipedia.org/wiki/SQLite) (which is an embedded

database), [LDAP](http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol) servers, and others. Numerous functions familiar to C programmers, such as those in the [stdio](http://en.wikipedia.org/wiki/Stdio.h) family, are available in standard PHP builds.

PHP allows developers to write [extensions](http://en.wikipedia.org/wiki/Software_extension) in [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) to add functionality to the PHP language. PHP extensions can be compiled statically into PHP or loaded dynamically at runtime. Numerous extensions have been written to add support for the [Windows API](http://en.wikipedia.org/wiki/Windows_API), process management on [Unix-like](http://en.wikipedia.org/wiki/Unix-like) [operating systems](http://en.wikipedia.org/wiki/Operating_system), multibyte strings ([Unicode](http://en.wikipedia.org/wiki/Unicode)), [cURL](http://en.wikipedia.org/wiki/CURL), and several popular [compression formats](http://en.wikipedia.org/wiki/Archive_format). Other PHP features made available through extensions include integration with [IRC](http://en.wikipedia.org/wiki/Internet_Relay_Chat), dynamic generation of images and [Adobe Flash](http://en.wikipedia.org/wiki/Adobe_Flash) content, *PHP Data Objects* (PDO) as an abstraction layer used for accessing databases, and even [speech synthesis](http://en.wikipedia.org/wiki/Speech_synthesis). Some of the language's core functions, such as those dealing with strings and arrays, are also implemented as extensions. The [PHP Extension Community Library](http://en.wikipedia.org/wiki/PHP_Extension_Community_Library) (PECL) project is a repository for extensions to the PHP language.

Some other projects, such as *Zephir*, provide the ability for PHP extensions to be created in a high-level language and compiled into native PHP extensions. Such an approach, instead of writing PHP extensions directly in C, simplifies the development of extensions and reduces the time required for programming and testing.

[Zend Technologies](http://en.wikipedia.org/wiki/Zend_Technologies) provides a [certification](http://en.wikipedia.org/wiki/Zend_Certified_Engineer) exam for programmers to become certified PHP developers.

## Installation and configuration

There are two primary ways for adding support for PHP to a web server – as a native web server module, or as a CGI executable. PHP has a direct module interface called [Server](http://en.wikipedia.org/wiki/Server_Application_Programming_Interface) Application Programming Interface (SAPI), which is supported by many web servers including [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), [Microsoft IIS](http://en.wikipedia.org/wiki/Microsoft_Internet_Information_Server), [Netscape](http://en.wikipedia.org/wiki/Netscape) (now defunct) and [iPlanet](http://en.wikipedia.org/wiki/IPlanet). Some other web servers, such as OmniHTTPd, support the [Internet Server Application Programming Interface](http://en.wikipedia.org/wiki/Internet_Server_Application_Programming_Interface) (ISAPI), which is a [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s web server module interface. If PHP has no module support for a web server, it can always be used as a [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) or [FastCGI](http://en.wikipedia.org/wiki/FastCGI) processor; in that case, the web server is configured to use PHP's CGI executable to process all requests to PHP files.

PHP-FPM (FastCGI Process Manager) is an alternative FastCGI implementation for PHP, bundled with the official PHP distribution since version 5.3.3. When compared to the older

FastCGI implementation, it contains some additional features, mostly useful for heavily loaded web servers.

When using PHP for command-line scripting, a PHP [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) (CLI) executable is needed. PHP supports a CLI SAPI as of PHP 4.3.0. The main focus of this SAPI is developing [shell](http://en.wikipedia.org/wiki/Command_line_interface) applications using PHP. There are quite a few differences between the CLI SAPI and other SAPIs, although they do share many of the same behaviors.

PHP can also be used for writing desktop [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) applications, by using the [PHP-GTK](http://en.wikipedia.org/wiki/PHP-GTK) extension. PHP-GTK is not included in the official PHP distribution, and as an extension it can be used only with PHP versions 5.1.0 and newer. The most common way of installing PHP-GTK is compiling it from the source code.

When PHP is installed and used in [cloud](http://en.wikipedia.org/wiki/Cloud_computing) environments, [software development kits](http://en.wikipedia.org/wiki/Software_development_kit) (SDKs) are provided for using cloud-specific features. For example:

* [Amazon Web Services](http://en.wikipedia.org/wiki/Amazon_Web_Services) provides the AWS SDK for PHP
* [Windows Azure](http://en.wikipedia.org/wiki/Windows_Azure) can be used with the Windows Azure SDK for PHP[.](http://en.wikipedia.org/wiki/PHP)

Numerous configuration options are supported, affecting both core PHP features and extensions. Configuration file php.ini is searched for in different locations, depending on the way PHP is used. The configuration file is split into various sections, while some of the configuration options can be also set within the web server configuration[.](http://en.wikipedia.org/wiki/PHP)

### Use

A broad overview of

the LAMP software

bundle, displayed here together with [Squid](http://en.wikipedia.org/wiki/Squid_%28software%29).

PHP is a general-purpose scripting language that is especially suited to [server-side](http://en.wikipedia.org/wiki/Server-side_scripting) [web](http://en.wikipedia.org/wiki/Web_development) development, in which case PHP generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server). Any PHP code in a requested file is [executed](http://en.wikipedia.org/wiki/Execution_%28computing%29) by the PHP runtime, usually to create [dynamic web page](http://en.wikipedia.org/wiki/Dynamic_web_page) content or dynamic

images used on websites or elsewhere. It can also be used for [command-line](http://en.wikipedia.org/wiki/Command-line) scripting and [client-](http://en.wikipedia.org/wiki/Client-side) side [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) applications. PHP can be deployed on most web servers, many [operating systems](http://en.wikipedia.org/wiki/Operating_system) and [platforms](http://en.wikipedia.org/wiki/Computing_platform), and can be used with many [relational database](http://en.wikipedia.org/wiki/Relational_database_management_system) management systems (RDBMS). Most [web hosting](http://en.wikipedia.org/wiki/Web_hosting) providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP acts primarily as a [filter](http://en.wikipedia.org/wiki/Filter_%28software%29), taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data. Most commonly the output will be HTML, although it could be [JSON](http://en.wikipedia.org/wiki/JSON), [XML](http://en.wikipedia.org/wiki/XML) or [binary data](http://en.wikipedia.org/wiki/Binary_data) such as image or audio formats. Since PHP 4, the PHP [parser](http://en.wikipedia.org/wiki/Parser) [compiles](http://en.wikipedia.org/wiki/Compiler) input to produce [bytecode](http://en.wikipedia.org/wiki/Bytecode) for processing by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine), giving improved performance over its [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29) predecessor.

Originally designed to create dynamic [web pages](http://en.wikipedia.org/wiki/Web_page), PHP now focuses mainly on [server-](http://en.wikipedia.org/wiki/Server-side_scripting) side scripting,[[126]](http://en.wikipedia.org/wiki/PHP) and it is similar to other server-side scripting languages that provide dynamic content from a web server to a [client](http://en.wikipedia.org/wiki/Client_%28computing%29), such as [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s [ASP.NET](http://en.wikipedia.org/wiki/ASP.NET), [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems)' [JavaServer Pages](http://en.wikipedia.org/wiki/JavaServer_Pages), and [mod\_perl](http://en.wikipedia.org/wiki/Mod_perl). PHP has also attracted the development of many [software](http://en.wikipedia.org/wiki/Software_framework) frameworks that provide building blocks and a design structure to promote [rapid application](http://en.wikipedia.org/wiki/Rapid_application_development) development (RAD). Some of these include [PRADO](http://en.wikipedia.org/wiki/PRADO_%28framework%29), [CakePHP](http://en.wikipedia.org/wiki/CakePHP), [Symfony](http://en.wikipedia.org/wiki/Symfony), [CodeIgniter](http://en.wikipedia.org/wiki/CodeIgniter), [Laravel](http://en.wikipedia.org/wiki/Laravel), [Yii Framework](http://en.wikipedia.org/wiki/Yii_Framework), and [Zend Framework](http://en.wikipedia.org/wiki/Zend_Framework), offering features similar to other [web application](http://en.wikipedia.org/wiki/List_of_web_application_frameworks) frameworks.

The [LAMP architecture](http://en.wikipedia.org/wiki/LAMP_architecture) has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the *P* in this bundle alongside [Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server) and [MySQL](http://en.wikipedia.org/wiki/MySQL), although the *P* may also refer to [Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29), [Perl](http://en.wikipedia.org/wiki/Perl), or some mix of the three. Similar packages, [WAMP](http://en.wikipedia.org/wiki/WAMP_%28software_bundle%29) and [MAMP](http://en.wikipedia.org/wiki/MAMP), are also available for [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) and [OS X](http://en.wikipedia.org/wiki/OS_X), with the first letter standing for the respective operating system. Although both PHP and Apache are provided as part of the Mac OS X base install, users of these packages seek a simpler installation mechanism that can be more easily kept up to date.

As of April 2007, over 20 million Internet domains had web services hosted on servers with PHP installed and mod\_php was recorded as the most popular [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server) module. As of October 2010, PHP was used as the server-side programming language on 75% of all websites whose server-side programming language was known (as of February 2014, the percentage had reached 82%), and PHP was the most-used open source software within

enterprises. [Web content management systems](http://en.wikipedia.org/wiki/Web_content_management_system) written in PHP include [MediaWiki](http://en.wikipedia.org/wiki/MediaWiki), [Joomla](http://en.wikipedia.org/wiki/Joomla), [eZ](http://en.wikipedia.org/wiki/EZ_Publish) Publish, [SilverStripe](http://en.wikipedia.org/wiki/SilverStripe), [WordPress](http://en.wikipedia.org/wiki/WordPress), [Drupal](http://en.wikipedia.org/wiki/Drupal), [Moodle](http://en.wikipedia.org/wiki/Moodle), the user-facing portion of [Facebook](http://en.wikipedia.org/wiki/Facebook), and [Digg](http://en.wikipedia.org/wiki/Digg).

For specific and more advanced usage scenarios, PHP offers a well defined and documented way for writing custom extensions in [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) or [C++](http://en.wikipedia.org/wiki/C%2B%2B). Besides extending the language itself in form of additional [libraries](http://en.wikipedia.org/wiki/Library_%28computing%29), extensions are providing a way for improving execution speed where it is critical and there is room for improvements by using a true [compiled language](http://en.wikipedia.org/wiki/Compiled_language). PHP also offers well defined ways for embedding itself into other software projects. That way PHP can be easily used as an internal [scripting language](http://en.wikipedia.org/wiki/Scripting_language) for another project, also providing tight interfacing with the project's specific internal [data structures](http://en.wikipedia.org/wiki/Data_structure).

PHP received mixed reviews due to lacking support for [multithreading](http://en.wikipedia.org/wiki/Multithreading_%28software%29) at the core language level, though using threads is made possible by the "pthreads" [PECL](http://en.wikipedia.org/wiki/PHP_Extension_Community_Library) extension.

# MYSQL

**MySQL** is (as of July 2013) the world's second most widely used [relational database](http://en.wikipedia.org/wiki/Relational_database_management_system) management system (RDBMS) and most widely used open-source RDBMS. It is named after co- founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, my. The [SQL](http://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query](http://en.wikipedia.org/wiki/Structured_Query_Language) Language.

The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB)**,** now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29)." [Free-](http://en.wikipedia.org/wiki/Free_software) software-open source projects that require a full-featured database management system often use MySQL.

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: [TYPO3](http://en.wikipedia.org/wiki/TYPO3), [MODx](http://en.wikipedia.org/wiki/MODx), [Joomla](http://en.wikipedia.org/wiki/Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal) and other software. MySQL is also used in many high-profile, large-scale

[websites](http://en.wikipedia.org/wiki/Website), including [Google](http://en.wikipedia.org/wiki/Google)[[13][14]](http://en.wikipedia.org/wiki/MySQL) (though not for searches), [Face book](http://en.wikipedia.org/wiki/Facebook), [Twitter](http://en.wikipedia.org/wiki/Twitter), [Flickr](http://en.wikipedia.org/wiki/Flickr), and [YouTube](http://en.wikipedia.org/wiki/YouTube).

## Interfaces

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, [MySQL](http://en.wikipedia.org/wiki/MySQL_Workbench) Workbench is actively developed by Oracle, and is freely available for use.

## Graphical

The official [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, [MySQL GUI Tools](http://en.wikipedia.org/wiki/MySQL_GUI_Tools). Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular [free and open source](http://en.wikipedia.org/wiki/Free_and_open_source_software) *Community Edition* which may be downloaded from the MySQL website, and the proprietary *Standard Edition* which extends and improves the feature set of the Community Edition.

Third-party proprietary and free graphical administration applications (or "front ends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known front ends, in alphabetical order, are:

* [Adminer](http://en.wikipedia.org/wiki/Adminer) – a free MySQL front end written in one [PHP](http://en.wikipedia.org/wiki/PHP) script, capable of managing multiple databases, with many [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) skins available
* [Chive](http://en.wikipedia.org/wiki/Chive) a free, open source, web-based database management tool designed as an alternative to phpMyAdmin
* [Database Workbench](http://en.wikipedia.org/wiki/Database_Workbench) – a software application for development and administration of multiple relational databases including MySQL, with interoperationality between different database systems
* [DBEdit](http://en.wikipedia.org/wiki/DBEdit) – a free front end for MySQL and other databases
* [HeidiSQL](http://en.wikipedia.org/wiki/HeidiSQL) – a full featured free front end that runs on [Windows](http://en.wikipedia.org/wiki/Windows), and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialised GUI features for date/time fields and enumerated multiple- value fields.
* [LibreOffice Base](http://en.wikipedia.org/wiki/LibreOffice) – LibreOffice Base allows the creation and management of databases, preparation of forms and reports that provide end users easy access to data. Like [Microsoft](http://en.wikipedia.org/wiki/Microsoft_Access) Access, it can be used as a front-end for various database systems, including Access databases (JET), ODBC data sources, and MySQL or [PostgreSQL](http://en.wikipedia.org/wiki/PostgreSQL).
* [Navicat](http://en.wikipedia.org/wiki/Navicat) – a series of proprietary graphical database management applications, developed for Windows, Macintosh and Linux
* [OpenOffice.org](http://en.wikipedia.org/wiki/OpenOffice.org) – freely available [OpenOffice.org Base](http://en.wikipedia.org/wiki/OpenOffice.org_Base) can manage MySQL databases if the entire suite is installed
* [phpMyAdmin](http://en.wikipedia.org/wiki/PhpMyAdmin) – a free Web-based front-end, widely installed by [web hosting services](http://en.wikipedia.org/wiki/Web_hosting_service) since it is developed in PHP and included in the LAMP stack, and [MAMP](http://en.wikipedia.org/wiki/MAMP), [XAMPP](http://en.wikipedia.org/wiki/XAMPP) and [WAMP](http://en.wikipedia.org/wiki/WAMP_%28software_bundle%29) software bundle installers
* [SQLBuddy](http://en.wikipedia.org/wiki/SQLBuddy) – a free Web-based front end, developed in PHP
* [SQLyog](http://en.wikipedia.org/wiki/SQLyog) – proprietary, but there is also a free 'community' edition available
* [Toad for MySQL](http://en.wikipedia.org/wiki/TOAD_%28software%29) – a free development and administration front end for MySQL from [Dell](http://en.wikipedia.org/wiki/Dell_Software) Software
* [Webmin](http://en.wikipedia.org/wiki/Webmin) – a free Web-based management utility and a MySQL front end, developed in Perl with some parts written in Java

Other available proprietary MySQL front ends include [dbForge Studio for MySQL](http://en.wikipedia.org/w/index.php?title=DbForge_Studio_for_MySQL&action=edit&redlink=1), DBStudio, [Epictetus](http://en.wikipedia.org/wiki/Epictetus_Database_Client), [Microsoft Access](http://en.wikipedia.org/wiki/Microsoft_Access), [Oracle SQL Developer](http://en.wikipedia.org/wiki/Oracle_SQL_Developer), SchemaBank, [SQLPro SQL](http://en.wikipedia.org/wiki/SQLPro_SQL_Client) Client, [Toad Data Modeler](http://en.wikipedia.org/wiki/Toad_Data_Modeler) and [DaDaBIK](http://en.wikipedia.org/wiki/DaDaBIK).

### Command line

MySQL ships with many [command line](http://en.wikipedia.org/wiki/Command_line) tools, from which the main interface is 'mysql' client. Third parties have also developed tools to manage MySQL servers.

* MySQL Utilities – a set of utilities designed to perform common maintenance and administrative tasks. Originally included as part of the MySQL Workbench, the utilities are now a stand-alone download available from Oracle.
* Percona Toolkit – a cross-platform toolkit for MySQL, developed in [Perl](http://en.wikipedia.org/wiki/Perl).[[31]](http://en.wikipedia.org/wiki/MySQL) Percona Toolkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Percona Toolkit is included with several [Linux](http://en.wikipedia.org/wiki/Linux) distributions such as [CentOS](http://en.wikipedia.org/wiki/CentOS) and [Debian](http://en.wikipedia.org/wiki/Debian), and packages are available for [Fedora](http://en.wikipedia.org/wiki/Fedora_%28operating_system%29) and [Ubuntu](http://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29) as well. Percona Toolkit was originally developed as Maatkit, but as of late 2011, Maatkit is no longer developed. **Programming**

MySQL works on many [system platforms](http://en.wikipedia.org/wiki/System_platform), including [AIX](http://en.wikipedia.org/wiki/AIX_operating_system), [BSDi](http://en.wikipedia.org/wiki/BSD/OS), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), [HP-UX](http://en.wikipedia.org/wiki/HP-UX), [eComStation](http://en.wikipedia.org/wiki/EComStation), [i5/OS](http://en.wikipedia.org/wiki/IBM_i5/OS), [IRIX](http://en.wikipedia.org/wiki/IRIX), [Linux](http://en.wikipedia.org/wiki/Linux), [OS X](http://en.wikipedia.org/wiki/OS_X), [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [NetBSD](http://en.wikipedia.org/wiki/NetBSD), [Novell NetWare](http://en.wikipedia.org/wiki/Novell_NetWare), [OpenBSD](http://en.wikipedia.org/wiki/OpenBSD), [OpenSolaris](http://en.wikipedia.org/wiki/OpenSolaris), [OS/2](http://en.wikipedia.org/wiki/OS/2) Warp, [QNX](http://en.wikipedia.org/wiki/QNX), [Oracle Solaris](http://en.wikipedia.org/wiki/Solaris_%28operating_system%29), [Symbian](http://en.wikipedia.org/wiki/Symbian), [SunOS](http://en.wikipedia.org/wiki/SunOS), [SCO OpenServer](http://en.wikipedia.org/wiki/SCO_OpenServer), SCO [UnixWare](http://en.wikipedia.org/wiki/UnixWare), [Sanos](http://en.wikipedia.org/wiki/Sanos) and [Tru64](http://en.wikipedia.org/wiki/Tru64). A port of MySQL to [OpenVMS](http://en.wikipedia.org/wiki/OpenVMS) also exists.

MySQL is written in [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [yacc](http://en.wikipedia.org/wiki/Yacc), but it uses a home- brewed [lexical analyzer](http://en.wikipedia.org/wiki/Lexical_analysis). Many [programming languages](http://en.wikipedia.org/wiki/Programming_language) with language-specific [APIs](http://en.wikipedia.org/wiki/Application_programming_interface) include [libraries](http://en.wikipedia.org/wiki/Library_%28computing%29) for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's [Visual Studio](http://en.wikipedia.org/wiki/Visual_Studio) (languages such as [C#](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29) and [VB](http://en.wikipedia.org/wiki/Visual_Basic) are most commonly used) and the JDBC driver for Java. In addition, an [ODBC](http://en.wikipedia.org/wiki/ODBC) interface called [MyODBC](http://en.wikipedia.org/wiki/MyODBC) allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages) or [ColdFusion](http://en.wikipedia.org/wiki/Adobe_ColdFusion). The [HTSQL](http://en.wikipedia.org/wiki/HTSQL) – [URL](http://en.wikipedia.org/wiki/Uniform_resource_locator)-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

## Features

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary [Enterprise Server](http://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plug-in, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

* A broad subset of ANSI SQL 99, as well as extensions
* Cross-platform support
* [Stored procedures](http://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](http://en.wikipedia.org/wiki/SQL/PSM)
* [Triggers](http://en.wikipedia.org/wiki/Database_trigger)
* [Cursors](http://en.wikipedia.org/wiki/Cursor_%28databases%29)
* Updatable [views](http://en.wikipedia.org/wiki/View_%28SQL%29)
* [Online DDL](http://en.wikipedia.org/wiki/Data_Definition_Language) when using the InnoDB Storage Engine.
* [Information schema](http://en.wikipedia.org/wiki/Information_schema)
* Performance Schema
* A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
* [X/Open XA](http://en.wikipedia.org/wiki/X/Open_XA) [distributed transaction processing](http://en.wikipedia.org/wiki/Distributed_transaction_processing) (DTP) support; [two phase commit](http://en.wikipedia.org/wiki/Two-phase-commit_protocol) as part of this, using the default [InnoDB](http://en.wikipedia.org/wiki/InnoDB) storage engine
* Transactions with [save points](http://en.wikipedia.org/wiki/Savepoint) when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
* [ACID](http://en.wikipedia.org/wiki/Atomicity%2C_consistency%2C_isolation%2C_durability) compliance when using InnoDB and NDB Cluster Storage Engines
* [SSL](http://en.wikipedia.org/wiki/Secure_Sockets_Layer) support
* Query [caching](http://en.wikipedia.org/wiki/Cache_%28computing%29)
* Sub-[SELECTs](http://en.wikipedia.org/wiki/Select_%28SQL%29) (i.e. nested SELECTs)
* Built-in [Replication](http://en.wikipedia.org/wiki/Database_replication) support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master. [Multi-master replication](http://en.wikipedia.org/wiki/Multi-master_replication) is provided in [MySQL](http://en.wikipedia.org/wiki/MySQL_Cluster) Cluster, and multi-master support can be added to unclustered configurations using Galera Cluster.
* Full-text [indexing](http://en.wikipedia.org/wiki/Index_%28database%29) and searching
* Embedded database library
* [Unicode](http://en.wikipedia.org/wiki/Unicode) support
* Partitioned tables with pruning of partitions in optimizer
* [Shared-nothing](http://en.wikipedia.org/wiki/Shared-nothing) clustering through [MySQL Cluster](http://en.wikipedia.org/wiki/MySQL_Cluster)
* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
* Native storage engines InnoDB, MyISAM, Merge, Memory (heap), [Federated](http://en.wikipedia.org/wiki/MySQL_Federated), Archive, [CSV](http://en.wikipedia.org/wiki/Comma-separated_values), Black hole, NDB Cluster.
* Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

The developers release minor updates of the MySQL Server approximately every two months. The sources can be obtained from MySQL's website or from MySQL's [Bazaar](http://en.wikipedia.org/wiki/Bazaar_%28software%29) repository, both under the GPL license.

## CHAPTER 5

**SYSTEM DESIGN AND DEVELOPMENT**

## INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer- based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system.

In the project, the input design is made in various web forms with various methods.For example, in the user creation form, the empty username and password is not allowed. The username if exists in the database, the input is considered to be invalid and is not accepted. Likewise, during the login process, the username is a must and must be available in the user list in the database. Then only login is allowed.

Input forms are User registration Login

view product

# OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

In the project, the user details, search, are the web forms in which the output is

available.

Outputs are Add cart reports

## DATABASE DESIGN

The database design is a must for any application developed especially more for the data store projects. Since the chatting method involves storing the message in the table and produced to the sender and receiver, proper handling of the table is a must. In the project, login table is designed to be unique in accepting the username and the length of the username and password should be greater than zero. The different users view the data in different format according to the privileges given.

## CHAPTER 6 TESTING

**INTRODUCTION:**

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depends on the system specifications and the time frame given for the design.

A newly designed system should have all the subsystems working together, but in reality each subsystems work independently. During this phase, all the subsystems are gathered into one pool and tested to determine whether it meets the user requirements.

Testing is done at two level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

## TYPES OF TESTING

**Unit Testing**

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values.

Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

## Integration Testing

In integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

## Validation Testing

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

## System Testing

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

## SYSTEM DEVELOPEMENT

A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. A number of system development life cycle (SDLC) models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next.

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping.

### FILE DESIGN

A ﬁle system provides the machinery to support the project tasks. At the highest level a ﬁle system is a way to organize, store, retrieve, and manage information on a permanent storage medium such as a disk. File systems manage permanent storage and form an integral part of all operating systems. There are many different approaches to the task of managing permanent storage. At one end of the spectrum are simple ﬁle systems that impose enough restrictions to inconvenience users and make using the ﬁle system difficult. In deciding what type of ﬁling system is appropriate for a particular operating system, we must weigh the needs of the problem with the other constraints of the project. The two basic abstractions of files and directories form the basis of what a ﬁle system can operate on. There are many operations that a file system can perform on ﬁles and directories. All ﬁle systems must provide some basic level of support. Beyond the most basic ﬁle system primitives lay other features, extensions, and more sophisticated operations.

The Structure of a File is given the concept of a ﬁle, a ﬁle system may impose no structure on the ﬁle, or it may enforce a considerable amount of structure on the contents of the ﬁle. An unstructured, “raw” ﬁle, often referred to as a “stream of bytes,” literally has no structure. The ﬁle system simply records the size of the ﬁle and allows programs to read the bytes in any order or fashion that they desire. If a ﬁle system chooses to enforce a formal structure on ﬁles, it usually does so in the form of records. With the concept of records, a programmer specifies the

size and format of the record, and then all I/O to that ﬁle must happen on record boundaries and be a multiple of the record length.

## IMPLEMENTATION

Implementation is the most crucial stage in achieving a successful system and giving the user’s confidence that the new system is effective and workable. Implementation of this project refers to the installation of the package in its real environment to the full satisfaction of the users and operations of the system.

Testing is done individually at the time of development using the data and verification is done the way specified in the program specification. In short, implementation constitutes all activities that are required to put an already tested and completed package into operation. The success of any information system lies in its successful implementation.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

The project execution was checked with live environment and the user requirements are satisfied. Proper implementation is essential to provide a reliable system to meet the organization requirements.

## CHAPTER 7

**CONCLUSION**

The project **“shopping cart”** is something like the original grocery shop shopping cart that is used by the customer in selecting certain products. Finally after selection the customer confirms orders for all the purchasing items and submits his/her account details with tax information at the checkout counter.

Shopping cart is used around the world in e-commerce to manage business through online. There are different kinds of software available that are useful for all in making purchase online. Through this software, one can choose the purchasing item and the software calculates the net amount for the order including packaging, moving and also taxes if applicable.

The software collects the credit card information of the customer and it provides a secure gateway for all kinds of transaction online. The shopping cart software provides a reliable platform for keeping all sensitive information. For this kind of online business, the special software must be installed on the server which host the site, or on a secure server which receives all sensitive data.

shopping cart software is its **security** as better security can attract customer by protecting their personal information. Security features include encrypting information and using a reputable processing service for credit cards.

### Future work:

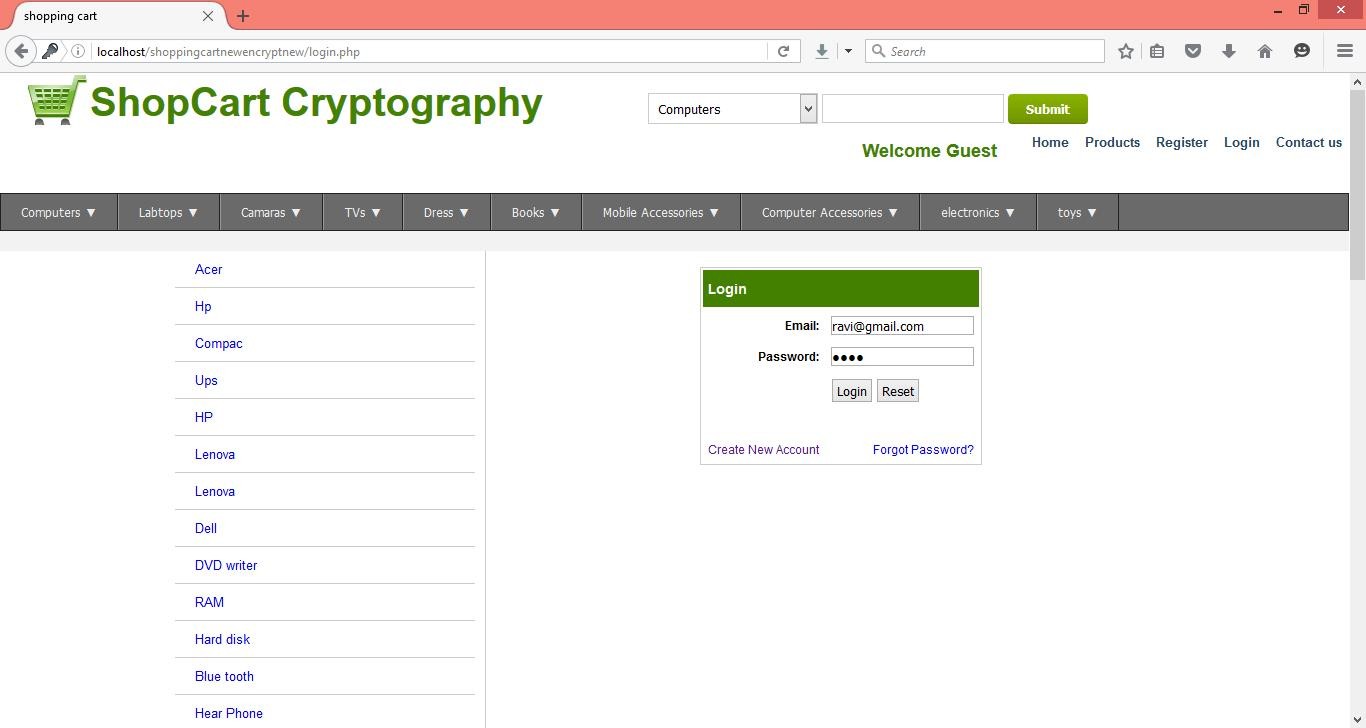
Shop online is on online application, which provides the online shopping facility available for everyone. Any type of the product will available for customer, and it can be easily purchased faster. Shop online application concentrates more on user friendly interface and promotes user to purchase faster and easier. There is a facility available for online purchase. Shop online has registration facility. in this project future work is android application development and also security maintain in cryptography method to bill.

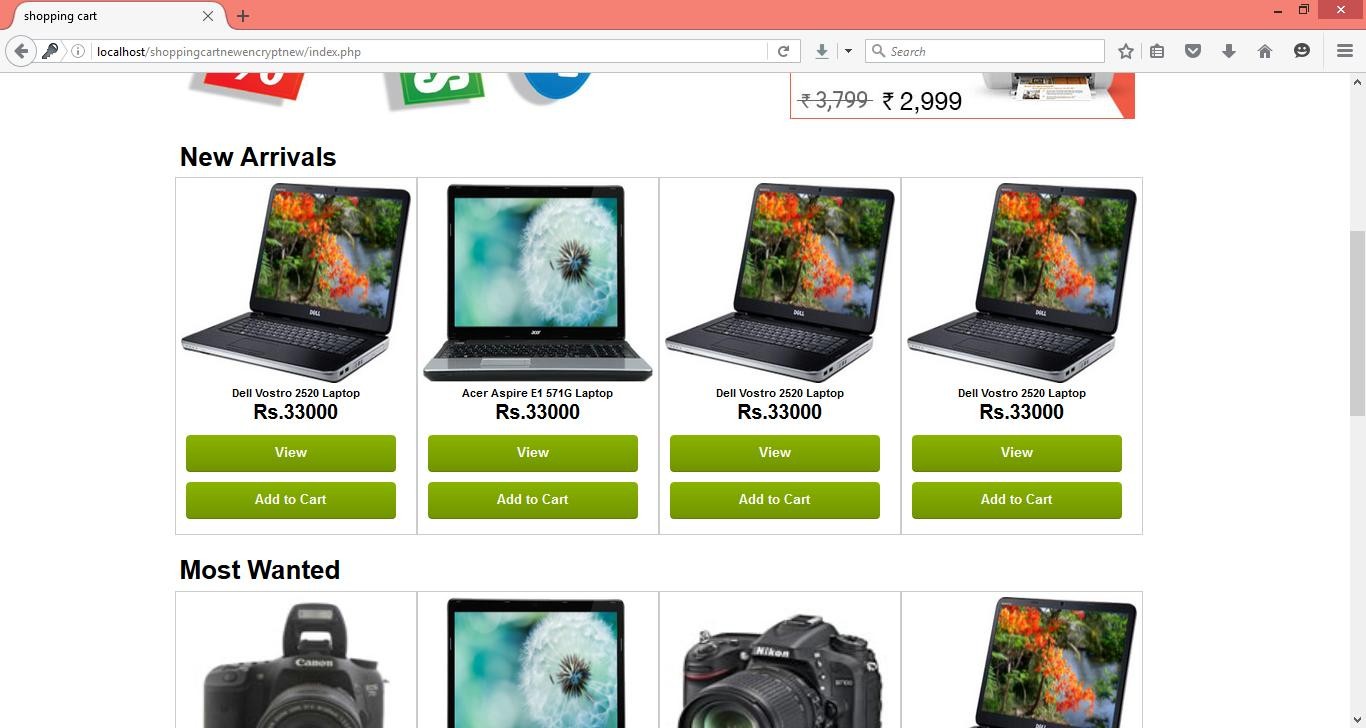
## Bibliography

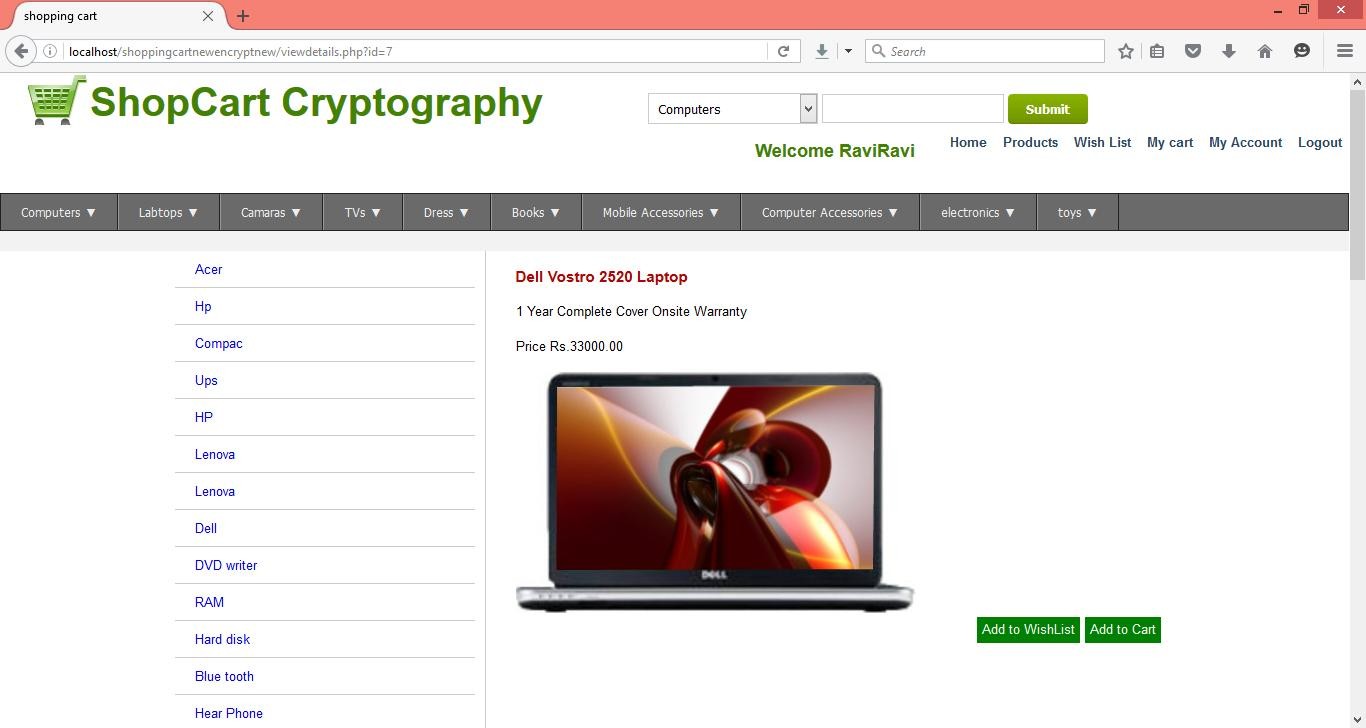
**Textual Reference**

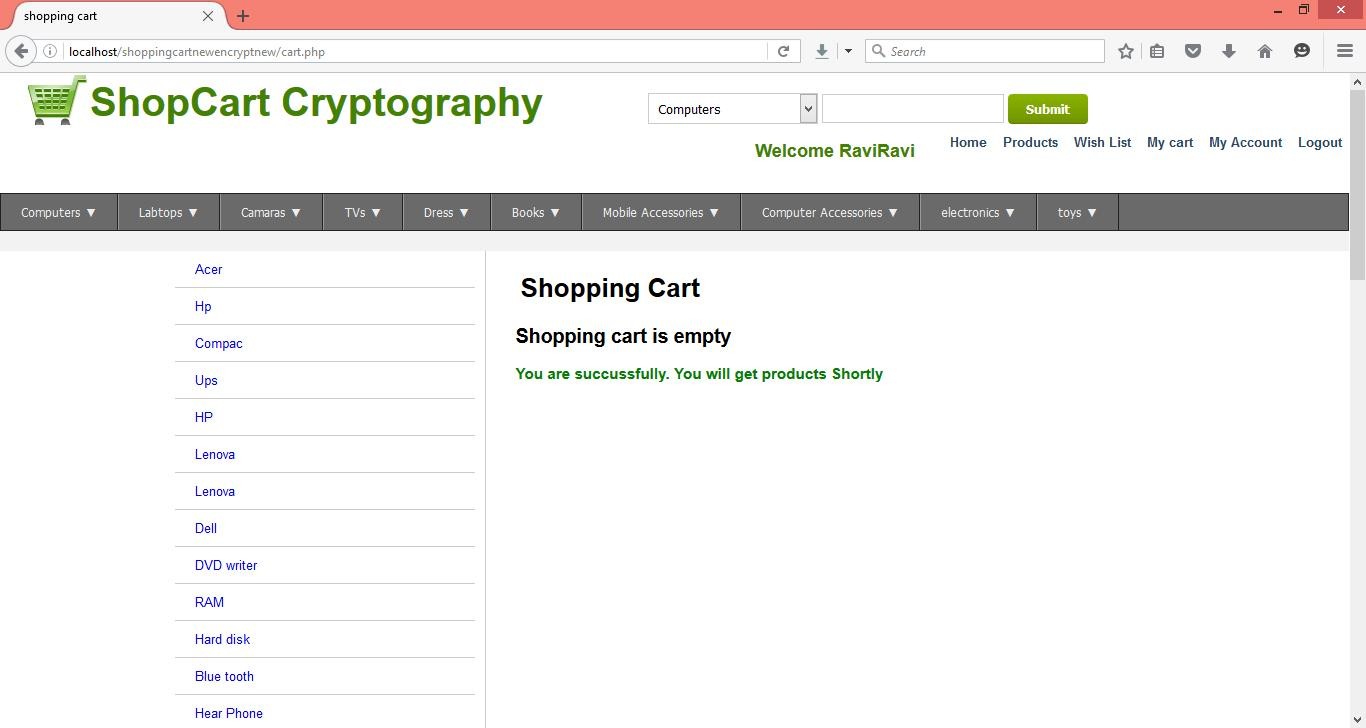
* PHP book by Vasvani (TMH publications).
* Beginning PHP5 by WROX.
* Informatics practices by Sumita Arora.
* Head First PHP & MySQL by Lynn Beighley and Michael Morrison(O’Reilly) Online Reference:
* [www.wikipedia.com](http://www.wikipedia.com/)
* [www.w3schools.com](http://www.w3schools.com/)
* <http://www.phpreferencebook.com/>

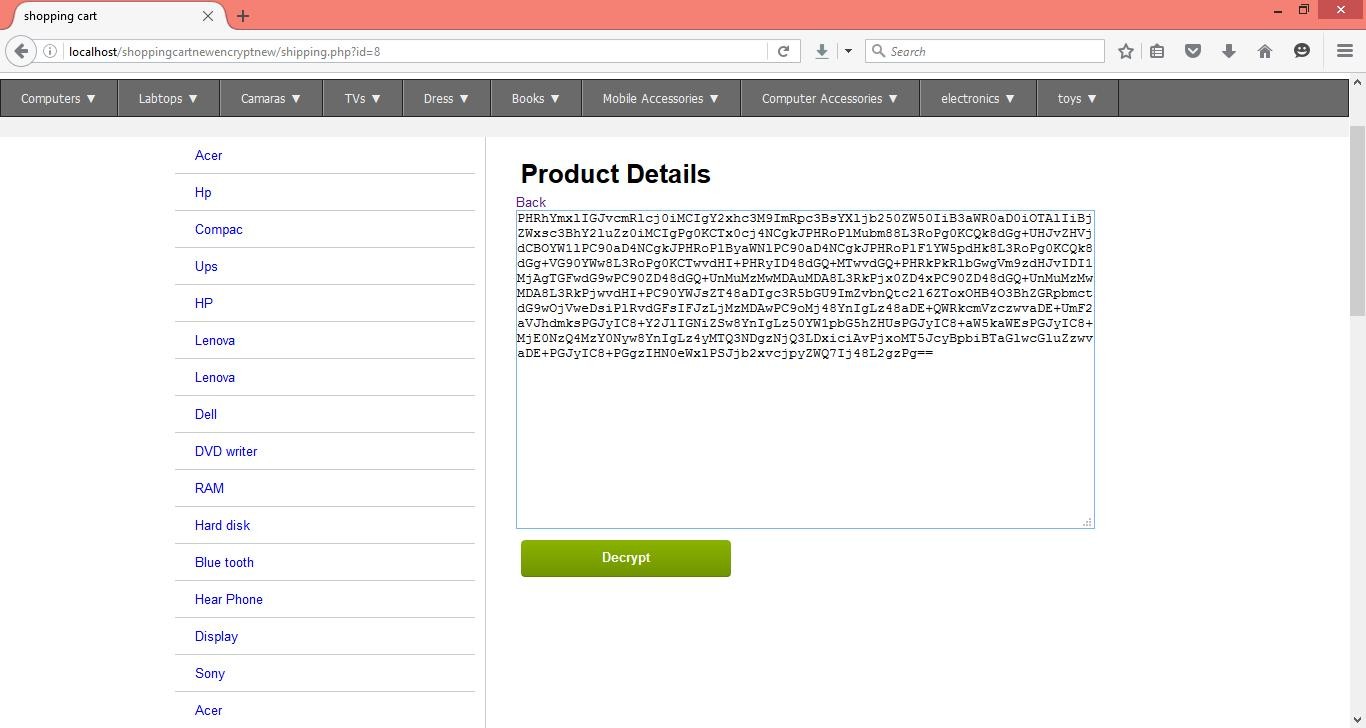
# SAMPLE SCREENSHOTS

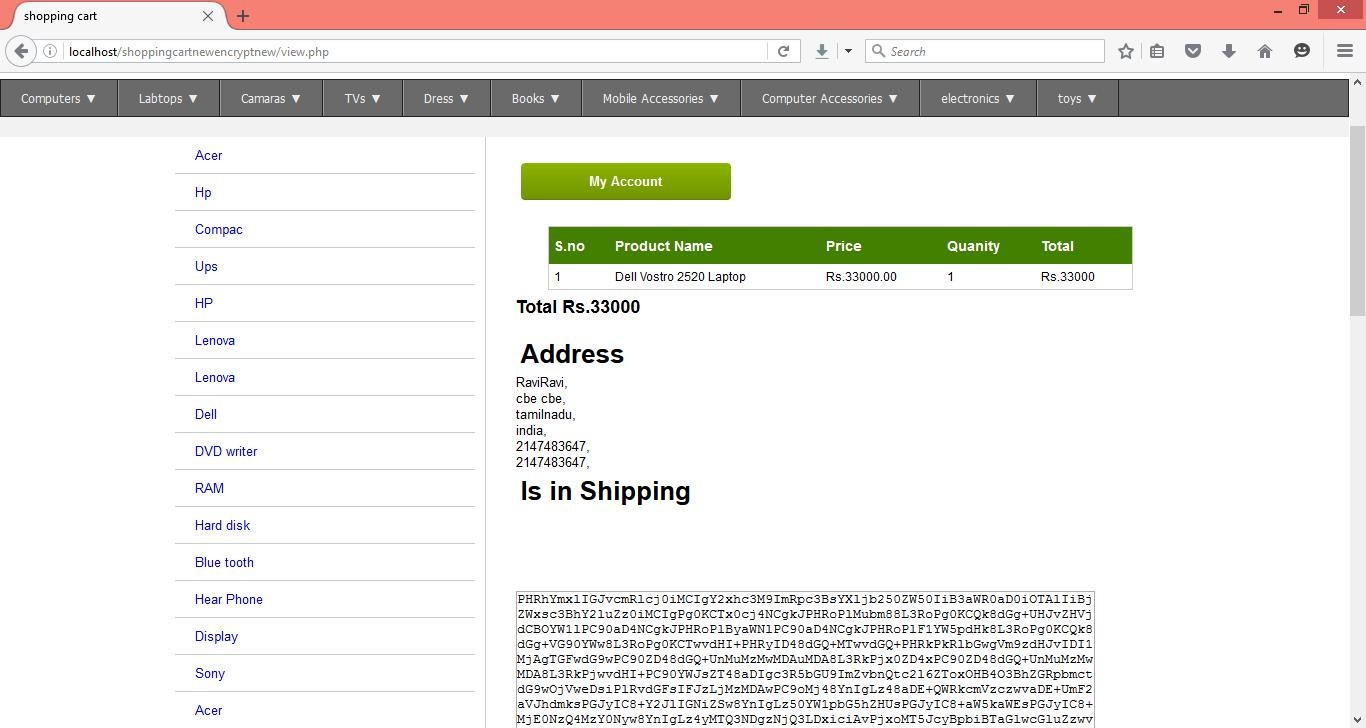


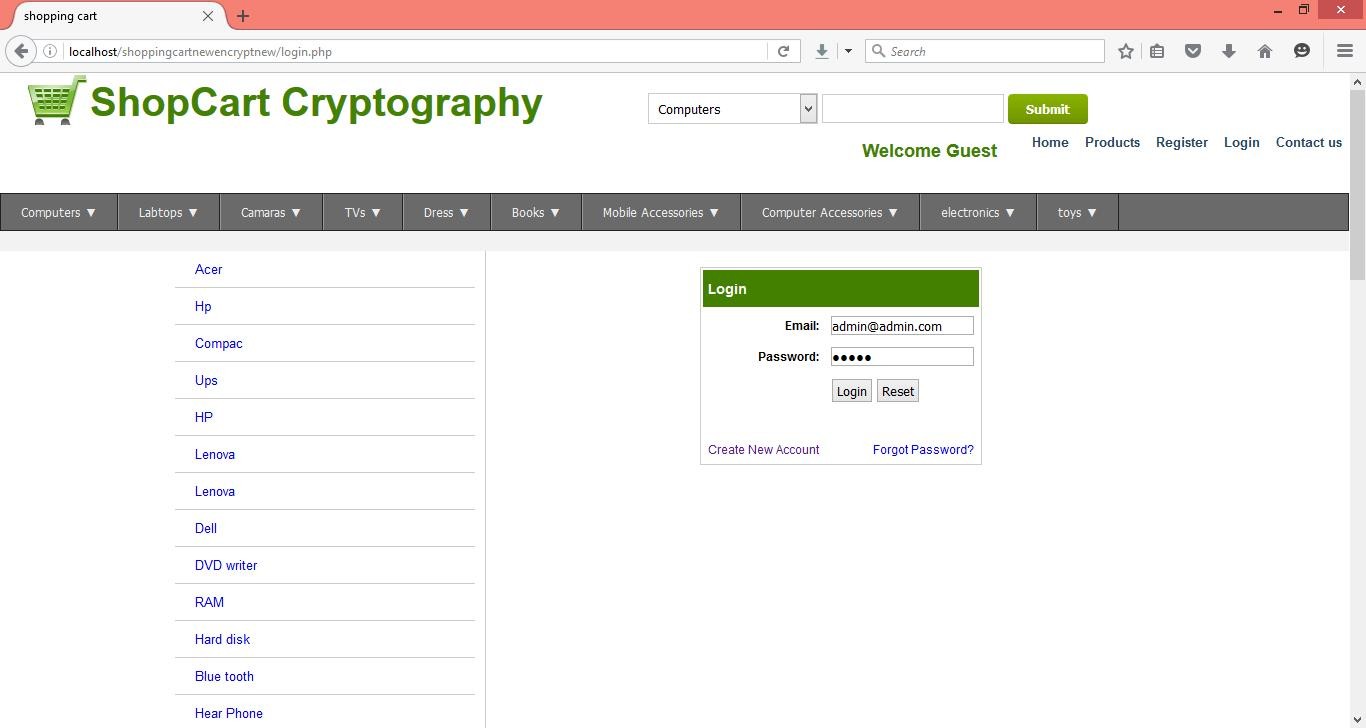


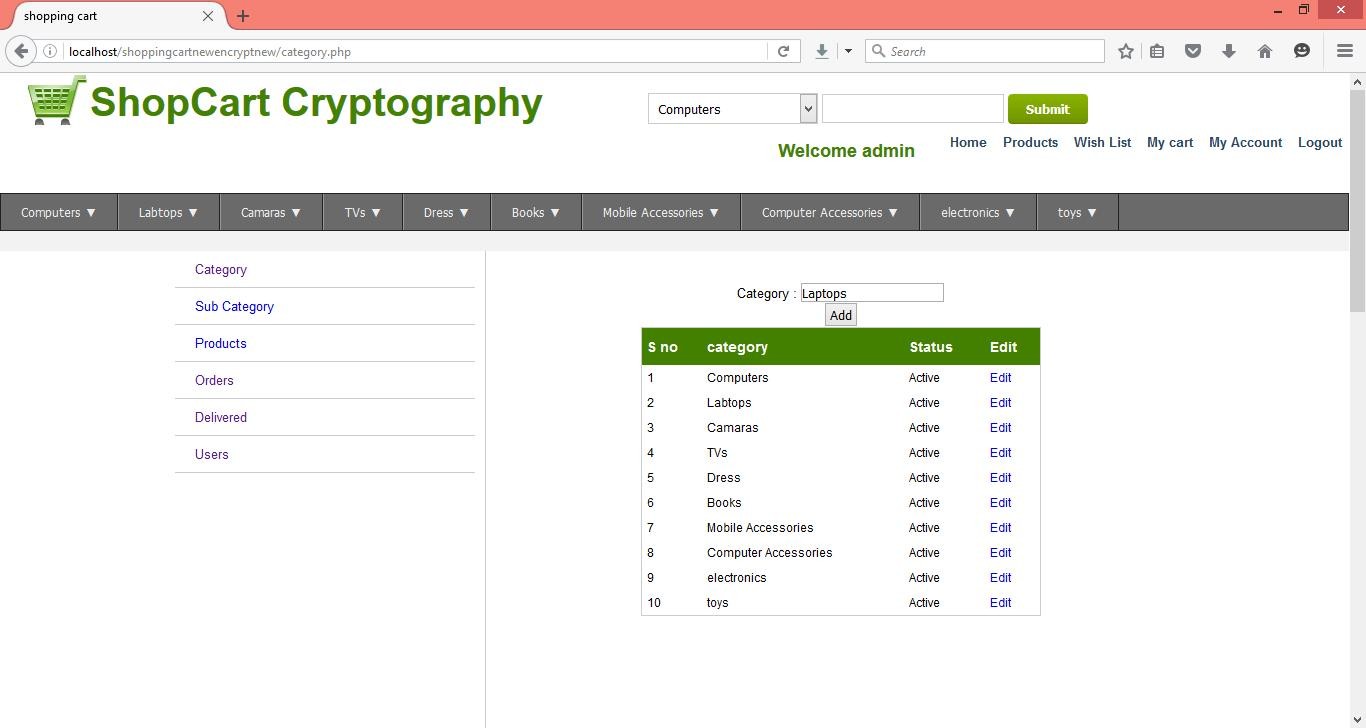


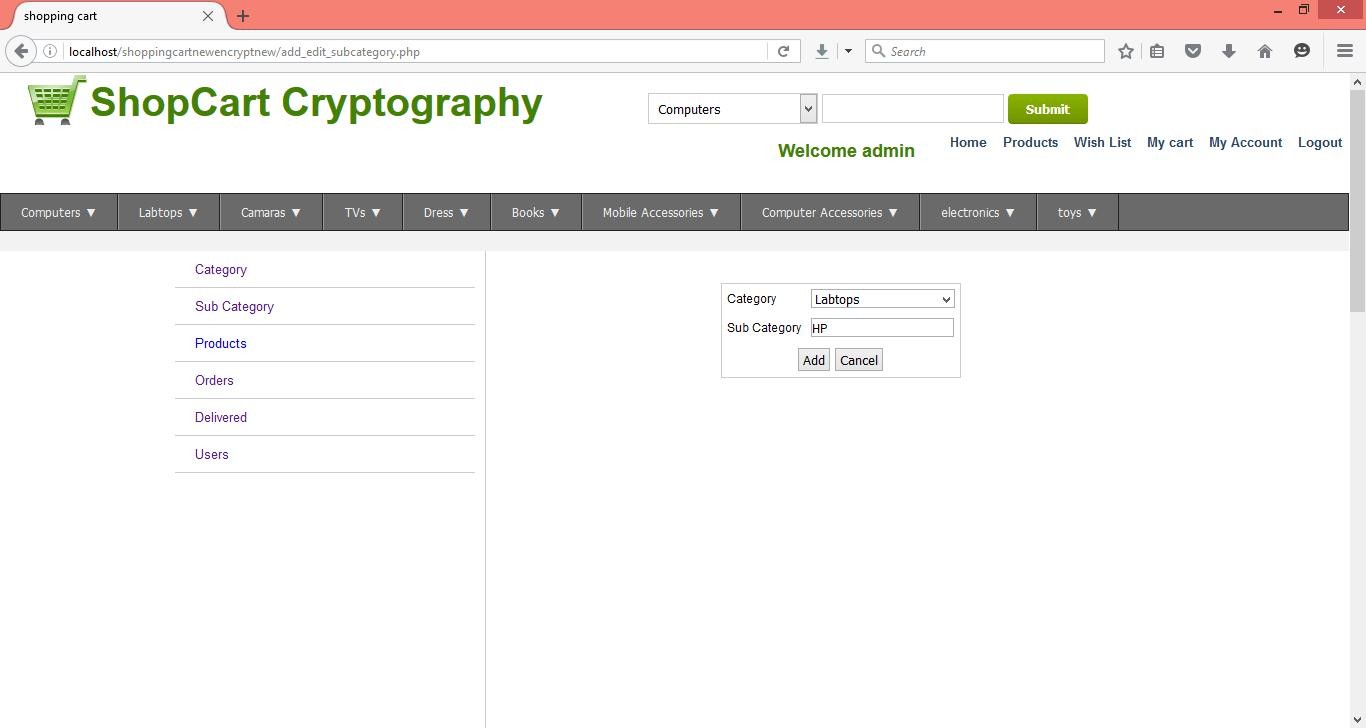


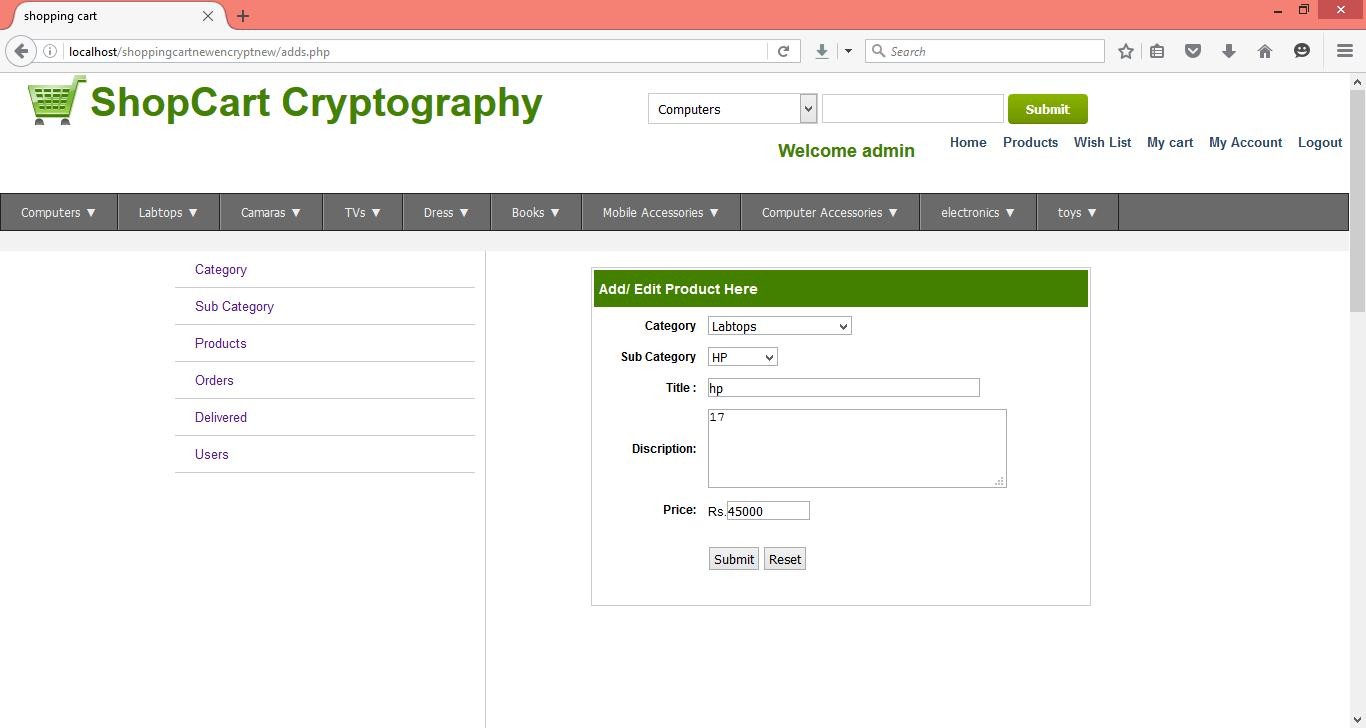










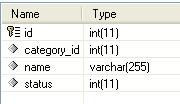


Tables

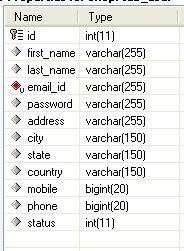
Category



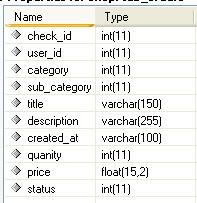
Sub category



user



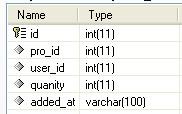
orders



Check out

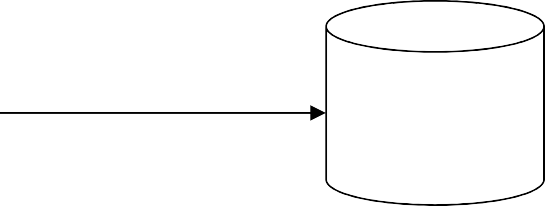
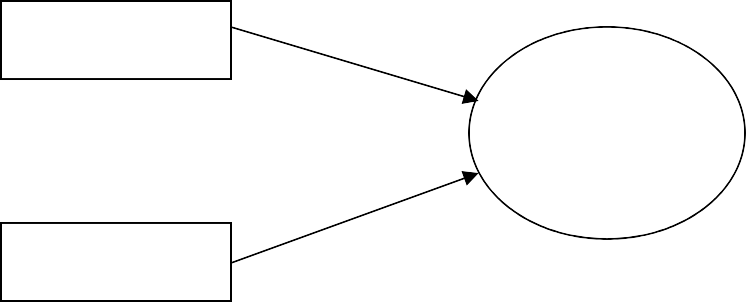


cart



Data flow diagram

mail id, name, pwd



User

Store

Shopping

card using cryptograph

DB

retrieve

Admin

mail id, pwd

Level 0

Admin

Category

Category

category name

Sub Category

Sub Category

subcategory name

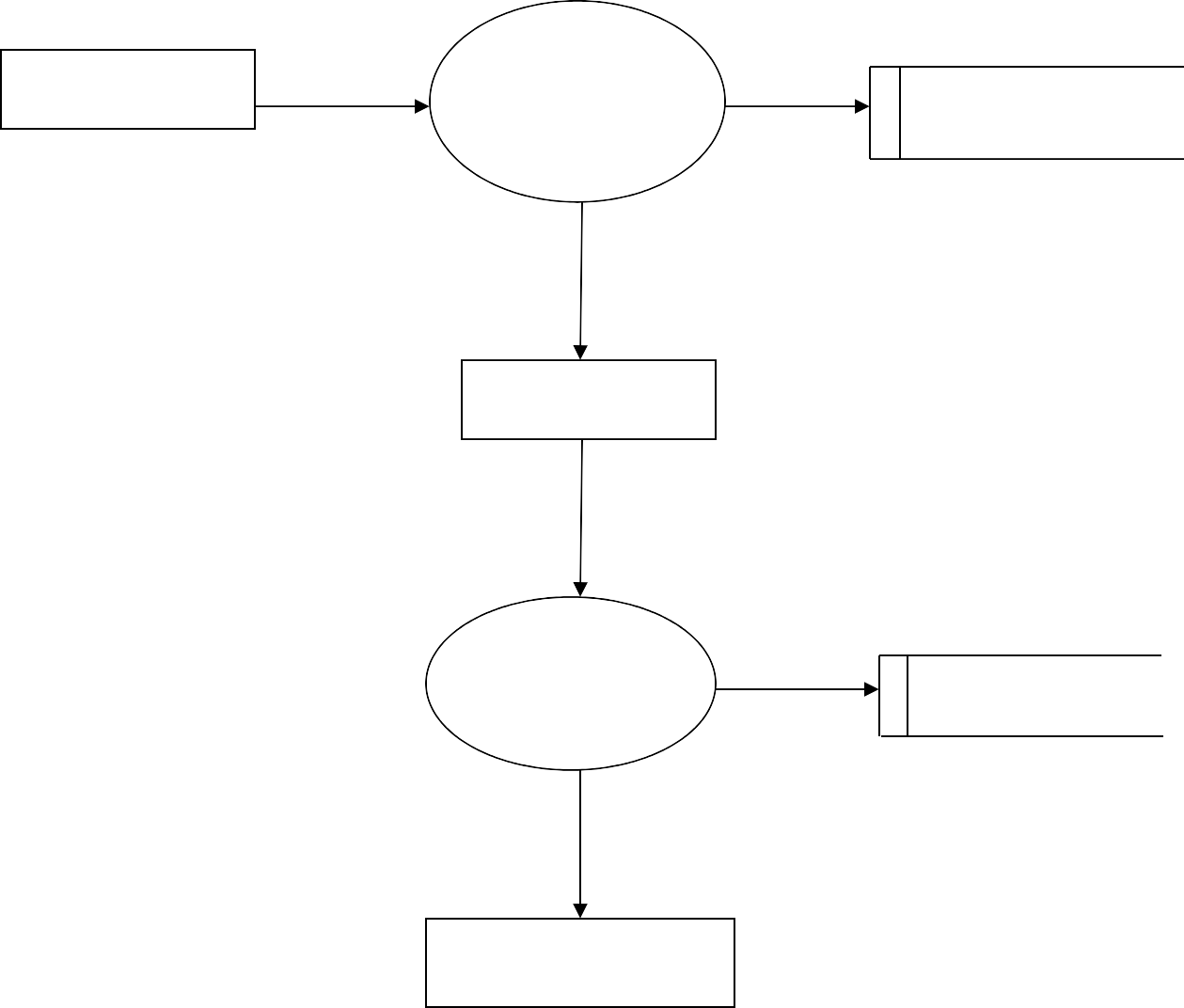
Add’s

Product

product name, price

Product

Level 1



User

Registration

User

mail id, address, pwd

Login

mail id, address, pwd

buy

product

pro id, name, price, qty

View product

<?php include\_once("config/config.php"); isAdminLoggedIn(); include\_once("common/header.php"); include\_once("common/left\_navi.php");

$query = "select \* from category ";

$result = mysql\_query($query) or die(mysql\_error()); while($row = mysql\_fetch\_assoc($result)){

$category[] = $row;

}

if(isset($\_POST['category'])){

$category\_id = $\_POST['category'];

}

else{

$category\_id = $category[0]['id'];

}

$subcategory = array();

$query = "select \* from sub\_category where category\_id='".$category\_id."'";

$result = mysql\_query($query); while($row = mysql\_fetch\_assoc($result)){

$subcategory[] = $row;

}

$i=1;?>

<div id="right\_navi">

<br /><br />

<a href="add\_edit\_subcategory.php" class="btn\_sm">Add</a>

<div align="center">

<form action="" method="post" id="subcategoryform"> Category :

<select name="category" onchange="document.getElementById('subcategoryform').submit();">

<?php

foreach($category as $cat){

if($cat['id'] == $category\_id){$selected= 'selected="selected"';} else $selected = '';

echo '<option '.$selected.' value="'.$cat['id'].'">'.$cat['category'].'</option>';

}

?>

</select>

</form>

</div>

<table border="0" cellspacing="0" class="displaycontent" width="400">

<tr>

<th>S no</th>

<th>Subcategory</th>

<th>Status</th>

<th>Edit</th>

</tr>

<?php if(count($subcategory)>0){

foreach($subcategory as $subcat){?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $subcat['name']; ?></td>

<td><?php if($subcat['status']==0) echo 'Active';else 'Deactive'; ?></td>

<td><a href='add\_edit\_subcategory.php?sub\_id=<?php echo

$subcat['id'];?>'>Edit</a></td>

</tr>

<?php }} ?>

</table>

</div>

</div>

<?php include\_once("common/footer.php")?>

<?php include\_once("config/config.php"); isUserLoggedIn(); include\_once("common/header.php"); include\_once("common/left\_navi.php");

$query = "SELECT \* FROM `tab\_checkout` where user\_id=".$\_SESSION['loggedIn'];

$result = mysql\_query($query); if(mysql\_num\_rows($result)>0){

?>

<div id="right\_navi">

<br />

<table border="0" cellspacing="0" class="displaycontent" width="600">

<tr>

<th>Sno.</th>

<th>Date</th>

<th>First Name</th>

<th>Last Name</th>

<th>Address</th>

<th>City</th>

<th>State</th>

<th>Country</th>

<th>Mobile</th>

<th>Phone</th>

<th>Status</th>

</tr>

<?php

$i=1;

while($row = mysql\_fetch\_assoc($result)){

?>

<tr>

<td><?php echo $i++; ?></td>

<td><a href="shipping.php?id=<?php echo $row['id'];?>"><?php echo

$row['created\_at']; ?></a></td>

<td><?php echo $row['first\_name']; ?></td>

<td><?php echo $row['last\_name']; ?></td>

<td><?php echo $row['address']; ?></td>

<td><?php echo $row['city']; ?></td>

<td><?php echo $row['state']; ?></td>

<td><?php echo $row['country']; ?></td>

<td><?php echo $row['mobile']; ?></td>

<td><?php echo $row['phone']; ?></td>

<td><?php if($row['status']) echo 'deliveried'; else echo 'Shipping'; ?></td>

</tr>

<?php

}

?>

</table>

<?php } else{?>

<br /><br /><br /><br /><br /><h2 align="center">No Record Found</h2>

<?php } ?>

</div>

</div>

<?php include\_once("common/footer.php");?>

<?php include\_once("config/config.php"); isAdminLoggedIn(); include\_once("common/header.php"); include\_once("common/left\_navi.php");

$query = "select \* from category ";

$result = mysql\_query($query) or die(mysql\_error()); while($row = mysql\_fetch\_assoc($result)){

$category[] = $row;

}

if(isset($\_POST['category'])){

$category\_id = $\_POST['category'];

}

else{

$category\_id = $category[0]['id'];

}

$subcategory = array();

$query = "select \* from sub\_category where category\_id='".$category\_id."'";

$result = mysql\_query($query); while($row = mysql\_fetch\_assoc($result)){

$subcategory[] = $row;

}

$i=1;?>

<div id="right\_navi">

<br /><br />

<a href="add\_edit\_subcategory.php" class="btn\_sm">Add</a>

<div align="center">

<form action="" method="post" id="subcategoryform"> Category :

<select name="category" onchange="document.getElementById('subcategoryform').submit();">

<?php

foreach($category as $cat){

if($cat['id'] == $category\_id){$selected= 'selected="selected"';} else $selected = '';

echo '<option '.$selected.' value="'.$cat['id'].'">'.$cat['category'].'</option>';

}

?>

</select>

</form>

</div>

<table border="0" cellspacing="0" class="displaycontent" width="400">

<tr>

<th>S no</th>

<th>Subcategory</th>

<th>Status</th>

<th>Edit</th>

</tr>

<?php if(count($subcategory)>0){

foreach($subcategory as $subcat){?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $subcat['name']; ?></td>

<td><?php if($subcat['status']==0) echo 'Active';else 'Deactive'; ?></td>

<td><a href='add\_edit\_subcategory.php?sub\_id=<?php echo

$subcat['id'];?>'>Edit</a></td>

</tr>

<?php }} ?>

</table>

</div>

</div>

<?php include\_once("common/footer.php")?>

<?php include\_once("config/config.php"); isUserLoggedIn(); include\_once("common/header.php"); include\_once("common/left\_navi.php");

$query = "SELECT \* FROM `tab\_checkout` where user\_id=".$\_SESSION['loggedIn'];

$result = mysql\_query($query); if(mysql\_num\_rows($result)>0){

?>

<div id="right\_navi">

<br />

<table border="0" cellspacing="0" class="displaycontent" width="600">

<tr>

<th>Sno.</th>

<th>Date</th>

<th>First Name</th>

<th>Last Name</th>

<th>Address</th>

<th>City</th>

<th>State</th>

<th>Country</th>

<th>Mobile</th>

<th>Phone</th>

<th>Status</th>

</tr>

<?php

$i=1;

while($row = mysql\_fetch\_assoc($result)){

?>

<tr>

<td><?php echo $i++; ?></td>

<td><a href="shipping.php?id=<?php echo $row['id'];?>"><?php echo

$row['created\_at']; ?></a></td>

<td><?php echo $row['first\_name']; ?></td>

<td><?php echo $row['last\_name']; ?></td>

<td><?php echo $row['address']; ?></td>

<td><?php echo $row['city']; ?></td>

<td><?php echo $row['state']; ?></td>

<td><?php echo $row['country']; ?></td>

<td><?php echo $row['mobile']; ?></td>

<td><?php echo $row['phone']; ?></td>

<td><?php if($row['status']) echo 'deliveried'; else echo 'Shipping'; ?></td>

</tr>

<?php

}

?>

</table>

<?php } else{?>

<br /><br /><br /><br /><br /><h2 align="center">No Record Found</h2>

<?php } ?>

</div>

</div>

<?php include\_once("common/footer.php");?>