INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Project eMart is a web application which is an ecommerce website. It is designed to help users shop online at home with easy attractive interface. Through this project I want users to provide a service where users don't have to do hard work for searching products around the market, when full market can be available to the users at their home. In this project users will get a fast and easy way to buy and sell products and keep a record of it. I chose this project to solve problems of people who find it difficult to shop outside or who have less access of the market. When anything or everything can be available to you by just sitting at home then why search outside.

1.2 ABOUT COMPANY

Since 2001, 4qt.com has been developing custom software, web, database and mobile application development delivering world-class services for established businesses and start-up companies. A trusted business partner and consultant to our customers, **4qt.com** leverages a powerful blend of the best industry-proven practices, standards based technologies, market understanding and extensive hands-on experience to meet our clients toughest challenges.

4qt.com is committed to creating reliable and cost effective software solutions to assist customers in reaching their business goals. Our solutions are driven by expertise in technology and an understanding of emerging business trends. Our custom made designs can help you enhance your corporate identity and brand. Our custom designed desktop or web software can help you with increased organizational productivity through automation of your business processes.

4th Quarter Technologies is a great place to work where you enjoy friendly work environment and great team of people.

1.3 REPORT ORGANISATION

The report is organized in the following way:

Chapter 1 starts with a brief introduction of the application and information about the company .

Chapter 2 focuses on analyzing the system by pointing out limitations of the existing system and how the proposed system is intended to solve the above said limitations. It also takes into account the feasibility study of the proposed system.

Chapter 3 deals with system specification i.e. the hardware and the software requirements for building and deploying the application.

Chapter 4 describes the software with its features that is used to build this web application. The front end and backend software with their general features are described in this section.

Chapter 5 gives details about system modules, use case diagrams and data flow diagrams.

Chapter 6 provides code used in creating the application. Code is written using PHP and designing is done using html5, css3, javascript and jquery.

Chapter 7 includes about test cases and unit/system testing which show required validations performed during development.

Chapter 8 includes snapshots of the web application i.e. the user interface of the system.

Chapter 9 summarizes the report under the heading conclusion and also provides future scope of the system.

CHAPTER 2
SYSTEM REQUIREMENTS AND ANALYSIS

2.1 SYSTEM OVERVIEW

2.1.1 EXISTING SYSTEM

In general, ecommerce websites have features of selling and buying but they don't have a shop feature. The existing ecommerce websites doesn't have a feature of shipment details. The feature shipment details gives the seller and buyer an ability to post the courier details from which courier and the track id the seller has ship the item so that the buyer can track his order online. This gives buyer a trust for online shopping. Existing ecommerce also doesn't include wallet feature.

2.1.2 PROPOSED SYSTEM

The Proposed system ensures the complete freedom for users, where user at his own choice can sell and buy and save all the details regarding buy products, sold products and posted products in their account. Also the proposed system provides shop feature that represents user's own shop with their respective posted products. This system also provides the shipment feature. This website also holds a feature of refund complain, where the buyer, if found the item faulty or not as described by the seller can ask for a refund. Then it's our duty to rectify the problem and details with the seller and buyer for further actions. This system also have a wallet feature where the user can save their money as eMart cash which can be further used for online shopping at the same website only.

2.2 OBJECTIVES

- To provide users at ease of selling and buying.
- To provide them a simple and efficient list of products from all around the country.
- To provide them more chance of selling their products.
- To help them by saving their every transaction on the go.
- To help them provide with more varieties and various new trends in market sitting at home.

2.3 PROJECT SCOPE

This project aims to provide users with fast and vast access of the market in India that the sellers want to bring to them. Users get easy access of variety of categories sitting at home, compare products with their features and description listed. Users will get a simple and efficient environment not just to buy but also to sell their own products, which can provide them a large amount of profit.

This project will help users in their selling their old or new products or buy products from other users and keep data or transactions intact along in their respective account for future reference (if any).

CHAPTER 3
SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a complete description of the behavior of the software to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. In addition to use cases, the SRS contains functional requirements, which define the internal workings of the software: that is, the calculations, technical details, data manipulation and processing, and other specific functionality that shows how the use cases are to be satisfied. It also contains nonfunctional requirements, which impose constraints on the design or implementation (such as performance requirements, quality standards or design constraints).

The SRS phase consists of two basic activities:

1) **Problem/Requirement Analysis**

The process is order and more nebulous of the two, deals with understanding the problem, the goal and constraints.

2) Requirement Specification

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

3.1 Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

3.2 Software Interface

Developer Side:	
Operating System	Windows 7
Application System	ANY

Database	MySQL
Programming language	PHP
Development IDE	phpDesigner 8
Client Side:	
Operating System	Any
Web Browser	Any
Server side:	
Operating system	Any
Application Server	ANY
DBMS	MySQL

3.3 Hardware Interface

Developer side:			
	Processor	RAM	Disk Space
	Intel core i5	8gb	1tb
Client Side:			
	ANY	512MB	1gb
Server Side:			
	Server Environment Capable Hardware	2 GB	As per the Size of the required Data base

Communication Interface:

- Client on internet will be using HTTP/HTTPS protocol
- Client on internet will be using TCP/IP protocols

Constraints:

- GUI is only in English.
- Login and Password is used for identification of Admin, Client and there is no facility for guest.
- This System is working for single server.
- There is no Maintainability of backup so availability will get effected.
- Limited to HTTP/HTTPS

CHAPTER 4
SOFTWARE DESCRIPTION

4.1 TECHNOLOGIES USED

- **FRONT END**
- HTML
- CSS
- JAVASCRIPT
- AJAX
- **BACK END**
- PHP
- MYSQL DATABASE

4.2 FRONT END TECHNOLOGIES

4.2.1 HTML AND CSS

HTML

HyperText Markup Language(HTML) is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

DHTML

Dynamic HTML, or DHTML, is an umbrella term for a collection of technologies used together to create interactive and animated web sites by using a combination of a static markup language (such as HTML), a client-side scripting language (such as JavaScript), a presentation definition language (such as CSS), and the Document Object Model.

DHTML allows scripting languages to change variables in a web page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process. Thus the dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

By contrast, a dynamic web page is a broader concept, covering any web page generated differently for each user, load occurrence, or specific variable values. This includes pages created by client-side scripting, and ones created by server-side scripting (such as PHP, Perl, JSP or ASP.NET) where the web server generates content before sending it to the client.

DHTML is differentiated from Ajax by the fact that a DHTML page is still request/reload-based. With DHTML, there may not be any interaction between the client and server after the page is loaded; all processing happens in JavaScript on the client side. By contrast, an Ajax page uses features of DHTML to initiate a request (or 'subrequest') to the server to perform actions such as loading more content.

WYSIWYG EDITORS

There are some WYSIWYG editors (What You See Is What You Get), in which the user lays out everything as it is to appear in the HTML document using a graphical user interface (GUI), often similar to word processors. The editor renders the document rather than show the code, so authors do not require extensive knowledge of HTML.

The WYSIWYG editing model has been criticized, primarily because of the low quality of the generated code; there are voices advocating a change to the WYSIWYM model (What You See Is What You Mean).

WYSIWYG editors remain a controversial topic because of their perceived flaws such as:

- Relying mainly on layout as opposed to meaning, often using markup that does not convey the intended meaning but simply copies the layout.
- Often producing extremely verbose and redundant code that fails to make use of the cascading nature of HTML and CSS.
- Often producing ungrammatical markup often called tag soup or semantically incorrect markup (such as <em> for italics).
- As a great deal of the information in HTML documents is not in the layout, the model has been criticized for its "what you see is all you get"-nature.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

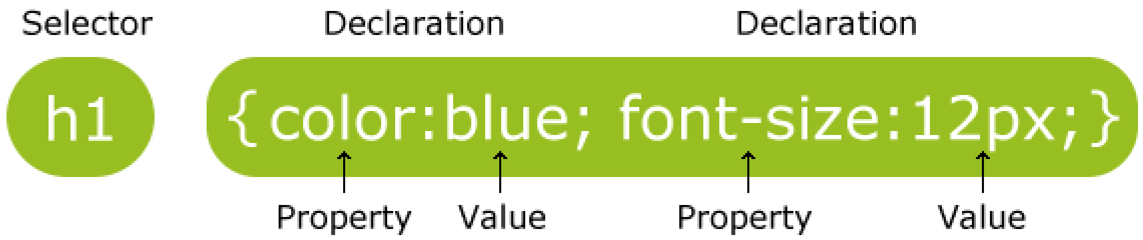
CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

SYNTAX

- A CSS rule set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

CSS EXAMPLE WITH COMMENTS

```
/*This is a multiple
lines comment*/
p
{
color:red;
/*This is another comment*/
text-align:center;
}
```

4.2.2 JAVASCRIPT

JAVASCRIPT

JavaScript (JS) is a dynamic computer programming language.[5] It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

JavaScript was formalized in the ECMAScript language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to computational objects within a host environment.

JAVASCRIPT SYNTAX AND EXAMPLE

The syntax of JavaScript is the set of rules that define a correctly structured JavaScript program.

The examples below make use of the alert function for standard text output. The JavaScript standard library lacks an official standard text output function. However, given that JavaScript is mainly used for client-side scripting within modern web browsers, and that almost all web browsers provide the alert function, alert is used in the examples.

VARIABLES

```
var x; //defines the variable x, although no value is assigned to it by default
var y = 2; //defines the variable y and assigns the value of 2 to it
```

A SIMPLE RECURSIVE FUNCTION:

```
function factorial(n) {
   if (n === 0) {
      return 1;
   }
   return n * factorial(n - 1);
}
```

JQUERY

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. It is currently developed by a team of developers led by Dave Methvin. Used by over 80% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The set of jQuery core features — DOM element selections, traversal and manipulation — enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM-data-structures; and influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio[8] for use within Microsoft's ASP.NET AJAX framework and ASP.NET MVC Framework while Nokia has integrated it into the Web Run-Time widget development platform. jQuery has also been used in MediaWiki since version 1.16.

JQUERY FEATURES

jQuery includes the following features:

- DOM element selections using the multi-browser open source selector engine Sizzle, a spin-off of the jQuery project.
- DOM manipulation based on CSS selectors that uses node elements name and node elements attributes (id and class) as criteria to build selectors
- Events
- Effects and animations
- AJAX
- JSON parsing
- Utilities - such as user agent information, feature detection
- Compatibility methods that are natively available in modern browsers but need fall backs for older ones - For example the inArray() and each() functions.
- Multi-browser (not to be confused with cross-browser) support.

INCLUDING THE LIBRARY

The jQuery library is a single JavaScript file, containing all of its common DOM, event, effects, and Ajax functions. It can be included within a web page by linking to a local copy, or to one of the many copies available from public servers.

From local host:

```
<script type="text/javascript" src="jquery.js"></script>
```

It is also possible to include jQuery directly from content delivery networks.

```
<script src="http://code.jquery.com/jquery-1.11.min.js"></script>
```

4.2.3 AJAX

AJAX

Ajax (an acronym for Asynchronous JavaScript and XML) is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications. With Ajax, Web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required; JSON is often used instead (see AJAJ), and the requests do not need to be asynchronous.

Ajax is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and allow the user to interact with, the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

AJAX TECHNOLOGIES

The term Ajax has come to represent a broad group of Web technologies that can be used to implement a Web application that communicates with a server in the background, without interfering with the current state of the page. In the article that coined the term Ajax, Jesse James Garrett explained that the following technologies are incorporated:

- HTML (or XHTML) and CSS for presentation
- The Document Object Model (DOM) for dynamic display of and interaction with data
- XML for the interchange of data, and XSLT for its manipulation
- The XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

4.3 BACK END TECHNOLOGIES

4.3.1 PHP

PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, a recursive backronym.

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

PHP SYNTAX AND SEMANTICS

The PHP syntax and semantics are the format (syntax) and the related meanings (semantics) of the text and symbols in the PHP programming language. They form a set of rules that define how a PHP program can be written and interpreted. PHP is a procedural and object-oriented language for coding webpage markup text to be transformed into HTML format.

Basic language constructs

Each PHP statement is terminated by semicolon (";"). The PHP markup can display text by using "echo" with variables named by dollar-prefix "$" on case-sensitive names ($xx, $xX, $NewX, etc.). The assignment operator is "=". The markup can be modularized into functions (or methods) defined with keyword "function" within optional classes named by "class xx". The control structures include: if, while, for, foreach, & switch. Grouping of text can be specified by curly braces ("{...}"), but some control structures can use colon syntax with end keywords, such as in statement if ($x==0) : echo "zero"; endif; <html>

Delimiters

The PHP processor only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and not parsed by PHP. The most common open/close delimiters are "<?php" and "?>" so that a section of PHP markup appears within the set of angle-brackets "<?php.....?>" (see examples below in boxes). Other delimiters, in the form "<script language="php">" and "</script>", are also always available, so these two forms are the most portable. The first form of delimiters, <?php and ?>, in XHTML and other XML documents, creates correctly formed XML processing instructions. Therefore, in either of these two cases, the resulting mixture of PHP and other markup is well-formed, and so probably valid, as XML and XHTML on the server before PHP processing. This may be helpful if the source code documents ever need to be processed in other ways during the life of the software.

Short opening tags (<? or <?=) are also available for use, but are, along with ASP style tags (<% or <%=), less-portable, as they can be disabled in the PHP configuration. For this reason, the use of Short tags and ASP style tags is discouraged. The purpose of these delimiters is to separate PHP code from non-PHP code (notably HTML). Everything outside the delimiters is ignored by the PHP parser and is passed through as output.

Variables and comments

One of the language characteristic features is implicit variable declaration. Variables are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case-sensitive. Both double-quoted ("") and heredoc strings allow the ability to embed a variable's value into the string. PHP treats newlines as whitespace, in the manner of a free-form language (except when inside string quotes). Statements are terminated by a semicolon. PHP has three types of comment syntax: /* */ which serves as block comments, and // as well as # which are used for inline comments. Many examples use the print function instead of the echo function. Both functions are nearly identical; the major difference being that print is slower than echo because the former will return a status indicating if it was successful or not in addition to text to output, whereas the latter does not return a status and only returns the text for output.

The usual "Hello World" code example for PHP is:

```
<?php
echo "Hello World!\n";
?>
```

The example above outputs the following:

```
Hello World!
```

Instead of using <? and the echo statement, an optional "shortcut" is the use of <?= instead of <? which implicitly echoes data. For example, to show the page_title:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
 <title><?=$page_title;?></title>
 </head>
 <body>
 <p>Hello World!</p>
 </body>
</html>
```

The above example also illustrates that text not contained within enclosing PHP tags will be directly output. So the simplest form of Hello World in PHP is a plain text file containing "Hello World".

Data types

PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit signed integers. Integer variables can be assigned using decimal (positive and negative), octal and hexadecimal notations. Real numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type, named "boolean", similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl. The null data type represents a variable that has no value. The only value in the null data type is NULL. 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 with both keys and values, and the two can be intermingled. Objects can syntactically be used as Arrays.

Functions

PHP has hundreds of base functions and thousands more from extensions. Functions are not first-class functions and can only be referenced by their name prior to PHP version 5.3.0, whereas PHP 5.3.0 introduces closures. User-defined functions can be created at any time and without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined.

An example function definition is the following:

```
<?php
function hello($target='World')
{
 echo "Hello $target!\n";
}
hello();
?>
```

Prior to version 5.3, PHP only supports quasi-anonymous functions through the create_function() function. These are not true anonymous functions because anonymous functions are nameless but functions can only be referenced by name in PHP. As of version 5.3, PHP supports true anonymous functions.

Function calls may be made via variables, where the value of a variable contains the name of the function to call. This is illustrated in the following example:

```
<?php
function hello()
{
  return 'Hello';
}
function world()
{
  return "World!";
}
$function1 = 'hello';
$function2 = 'world';
echo $function1() . ' ' . $function2();
?>
```

PHP does not support named parameters or parameter skipping. Some core PHP developers have publicly expressed disappointment with this decision. Others have suggested workarounds for this limitation.

4.3.2 MYSQL

MYSQL

MySQL is (as of March 2014) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius's daughter, My. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

Features

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM
- Triggers
- Cursors
- Updatable views
- Information schema

- Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB and NDB Cluster storage engines; savepoints with InnoDB
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master. Multi-master replication is provided in MySQL Cluster, and multi-master support can be added to unclustered configurations using Galera Cluster.
- Full-text indexing and searching (initially a MyISAM-only feature; supported by InnoDB since the release of MySQL 5.6)
- Embedded database library
- Unicode support (however prior to 5.5.3 UTF-8 and UCS-2 encoded strings are limited to the BMP, in 5.5.3 and later use utf8mb4 for full unicode support)
- ACID compliance when using transaction capable storage engines (InnoDB and Cluster)
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time):
- Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, EXAMPLE, Aria, and InnoDB, which was made the default as of 5.5)

- Partner-developed storage engines (solidDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.
- Community-developed storage engines (memcache engine, httpd, PBXT, Revision Engine)
- Custom storage engines
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second. (PostgreSQL has an advanced form of this functionality)
- The developers release monthly versions of the MySQL Server. The sources can be obtained from MySQL's website or from MySQL's Bazaar repository, both under the GPL license.

Limitations

Like other SQL databases, MySQL does not currently comply with the full SQL standard for some of the implemented functionality, including foreign key references when using some storage engines other than the 'standard' InnoDB (or third-party engines which supports foreign keys).

Up until MySQL 5.7, triggers are limited to one per action / timing, meaning that at most one trigger can be defined to be executed after an INSERT operation, and one before INSERT on the same table. No triggers can be defined on views.

MySQL, like most other transactional relational databases, is strongly limited by hard disk performance. This is especially true in terms of write latency. Given the recent appearance of very affordable consumer grade SATA interface solid-state drives that offer zero mechanical latency, a fivefold speedup over even an eight drive RAID array can be had for a smaller investment.

MySQL database's inbuilt functions like UNIX_TIMESTAMP() will return 0 after 03:14:07 UTC on 19 January 2038.

CHAPTER 5
SYSTEM DESIGN

5.1 INTRODUCTION

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

5.1.1 Unified Modeling Language:

UML stands for Unified Modeling Language. It is a third generation method for specifying, visualizing and documenting the artifacts of an object oriented system under development. Object modeling is the process by which the logical objects in the real world (problem space) are represented (mapped) by the actual objects in the program (logical or a mini world). This visual representation of the objects, their relationships and their structures is for the ease of understanding. This is a step while developing any product after analysis.

The goal from this is to produce a model of the entities involved in the project which later need to be built. The representations of the entities that are to be used in the product being developed need to be designed.

Software design is a process that gradually changes as various new, better and more complete methods with a broader understanding of the whole problem in general come into existence.

The Unified Modeling Language encompasses a number of models.

- Use case diagrams
- Class diagrams
- Sequence diagrams

5.1.2 Use Case Diagram

Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from user’s perspective.
- The use cases are the functions that are to be performed in the module.

5.1.3 Class Diagram

Class Diagram consists of the classes and the objects and the interaction between them. It mainly deals with the interaction between classes in the system, their behavior and properties of the system. Apart from classes this also provides inheritance relationships in the project. Class diagrams consist of basically two parts: first one is the member variables and class variables and the second part consists of the total number of methods available in the class.

5.1.4 Sequence Diagram

The purpose of sequence diagram is to show the flow of functionality through a use case. In other words, we call it a mapping process in terms of data transfers from the actor through the corresponding objects.

The key points are:

- The main purpose is to represent the logical flow of data with respect to a process
- A sequence diagram displays the objects and not the classes.

5.2 USE CASE DIAGRAM

USER

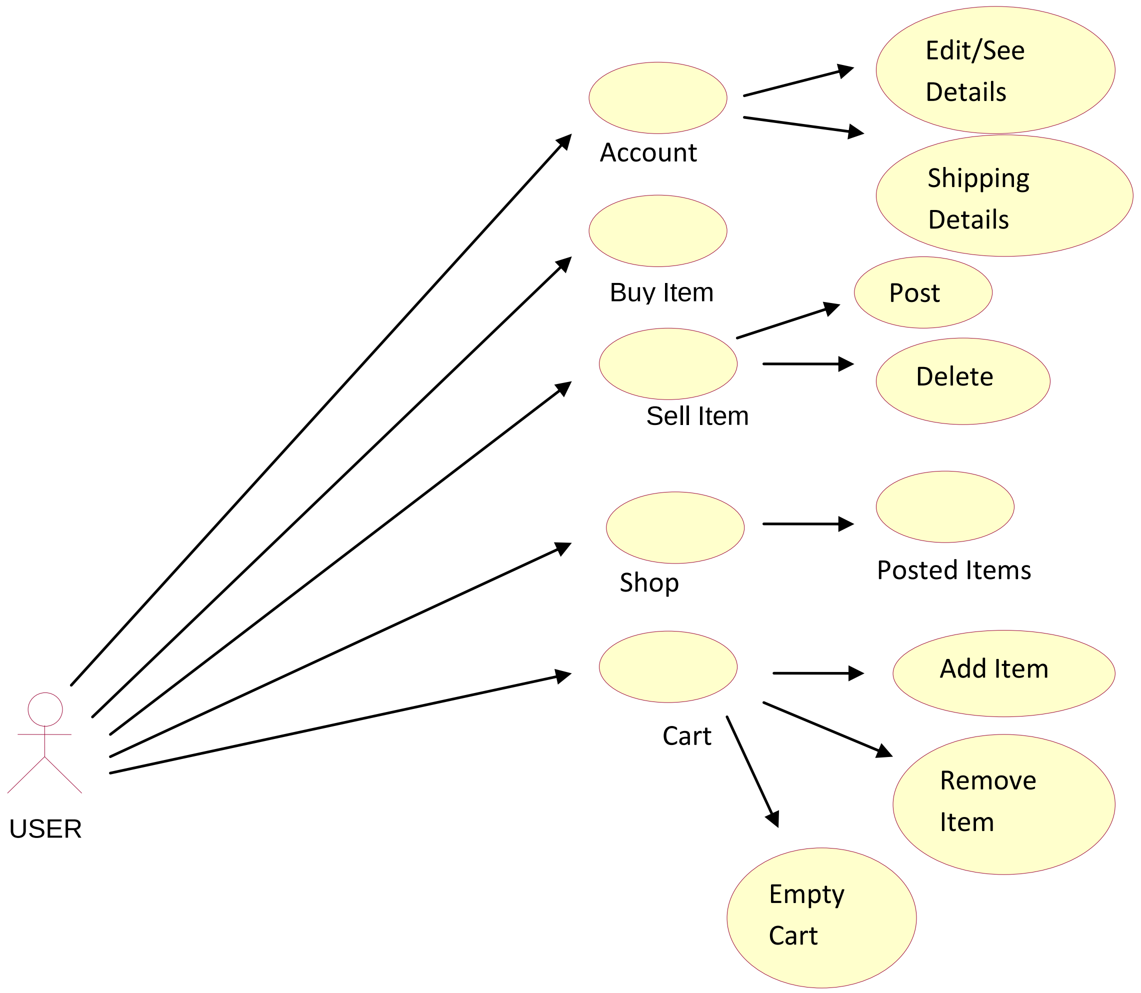
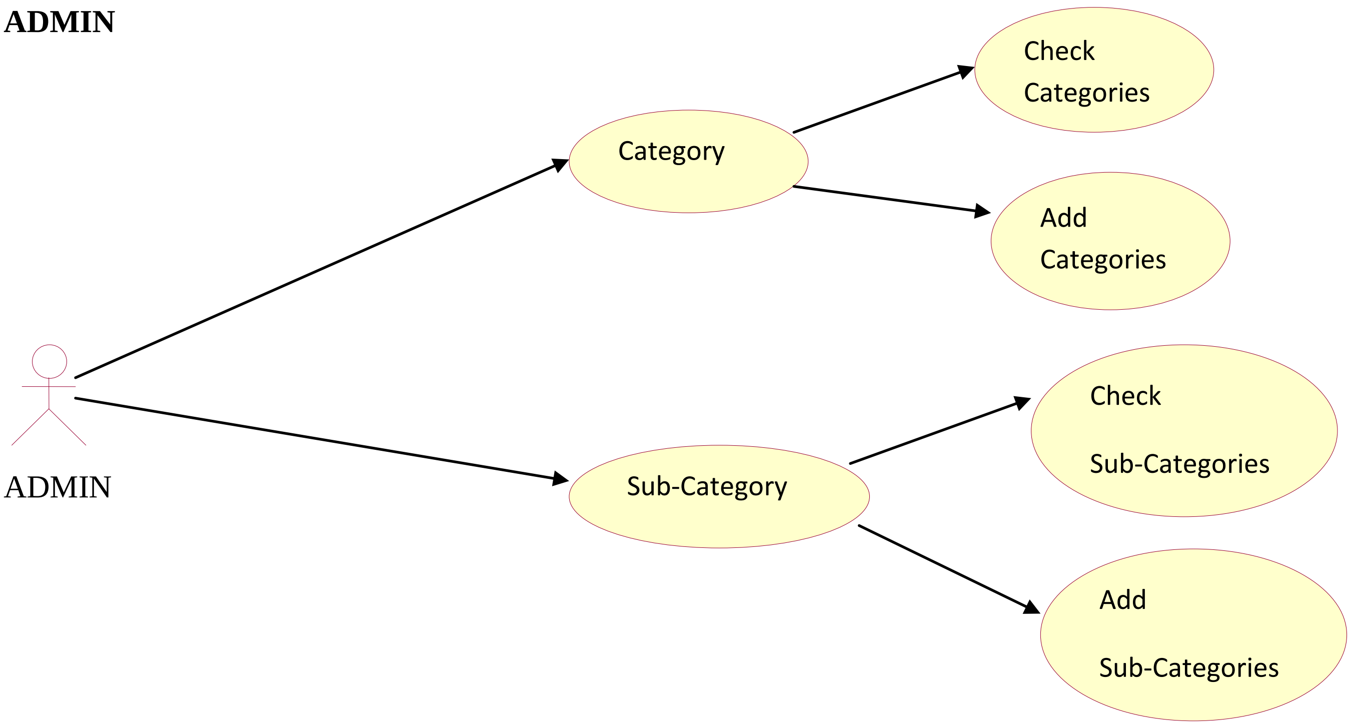


Figure 5.1: User Diagram

ADMIN



<u>Figure 5.2</u>: Admin Diagram

5.3 DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called a "*context diagram*".

5.3.1 Context Diagram

It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

A DFD is also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

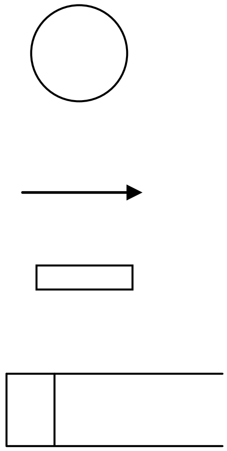
5.3.2 DFD SYMBOLS

In the DFD, there are four symbols:

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows. Data move in a specific direction from an origin to a destination.

3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

5.3.3 Symbols Elementary references



Process that transforms data flow

Data Flow

Source or Destination of data

Data Store

5.3.4 Constructing a DFD

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy interface. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

5.3.5 Salient features of DFD'S

The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.

1. The DFD does not indicate the time factor involved in any process whether the data flow take place daily, weekly or monthly.
2. The sequence of events is not brought out on the DFD.

5.3.6 Types of data flow diagrams

DFD's are of two types

(a) Physical DFD

(b) Logical DFD

1. Physical DFD

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensurethat the current system has been clearly understood. Physical DFDs shows actual devices, departments, and people etc., involved in the current system

2. Logical DFD

Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

5.3.7 Rules Governing the DFD'S

Process

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase level.

Data Store

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store, a process, which retrieves, must move data from the source and place the data into data store.
3. A data store has a noun phrase level.

5.4 DATAFLOW DIAGRAMS FOR eMart

LEVEL 0: DFD

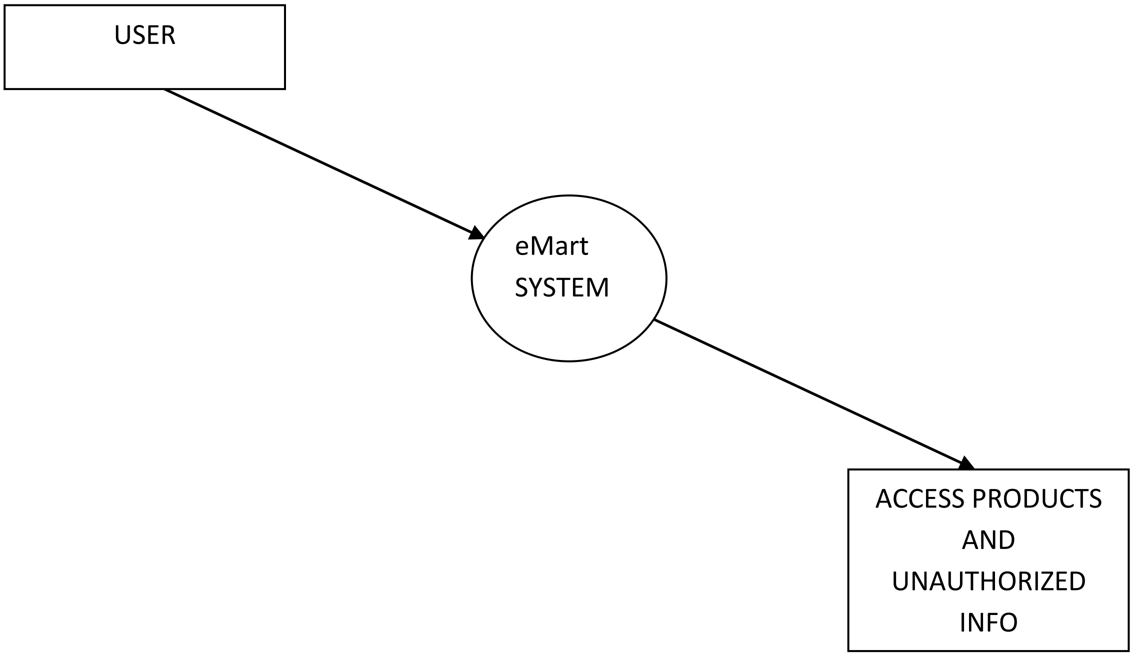


Figure 5.3: Level 0: DFD

LEVEL 1: DFD

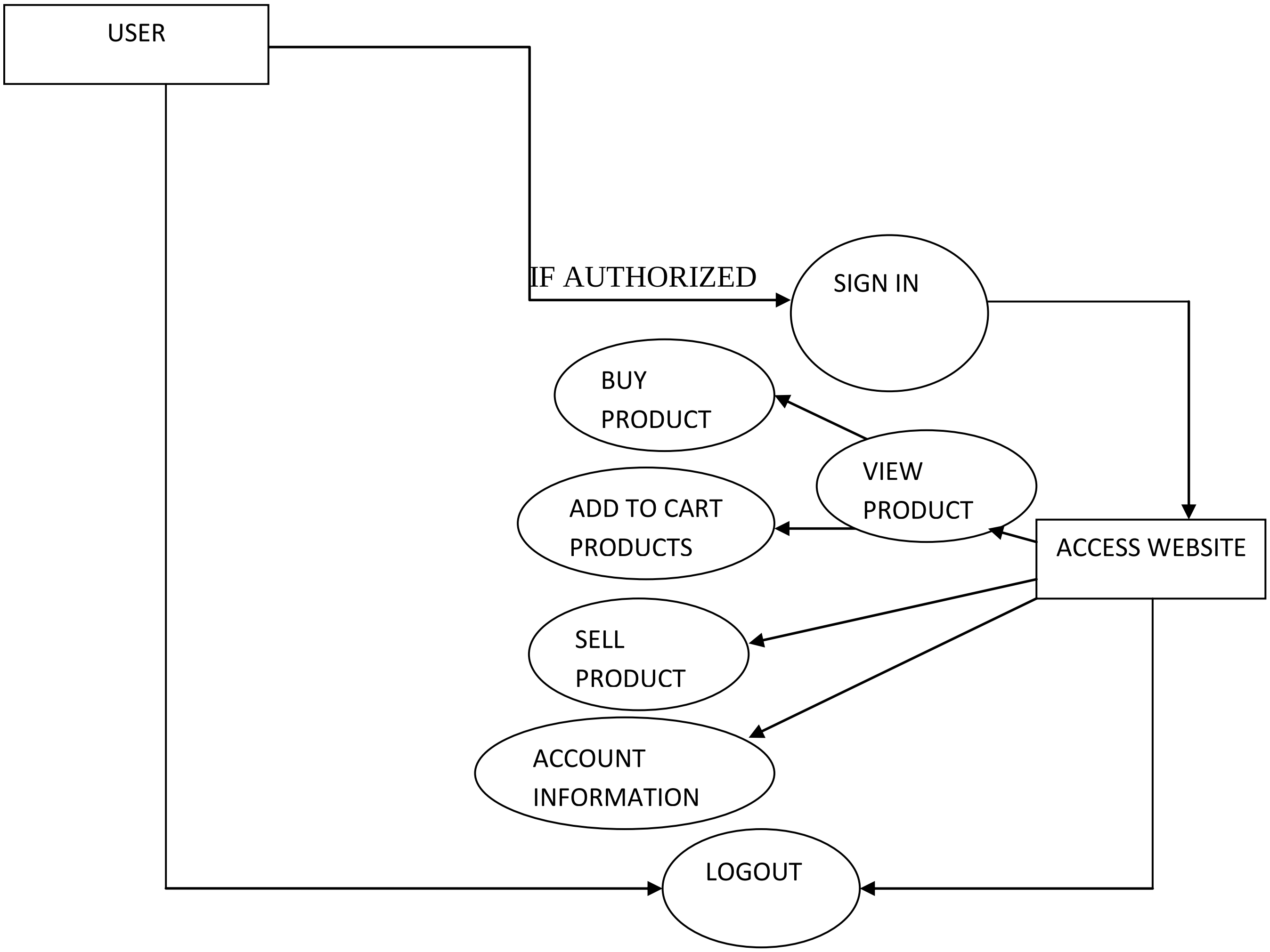


Figure 5.4: Level 1: DFD

LEVEL 2: DFD

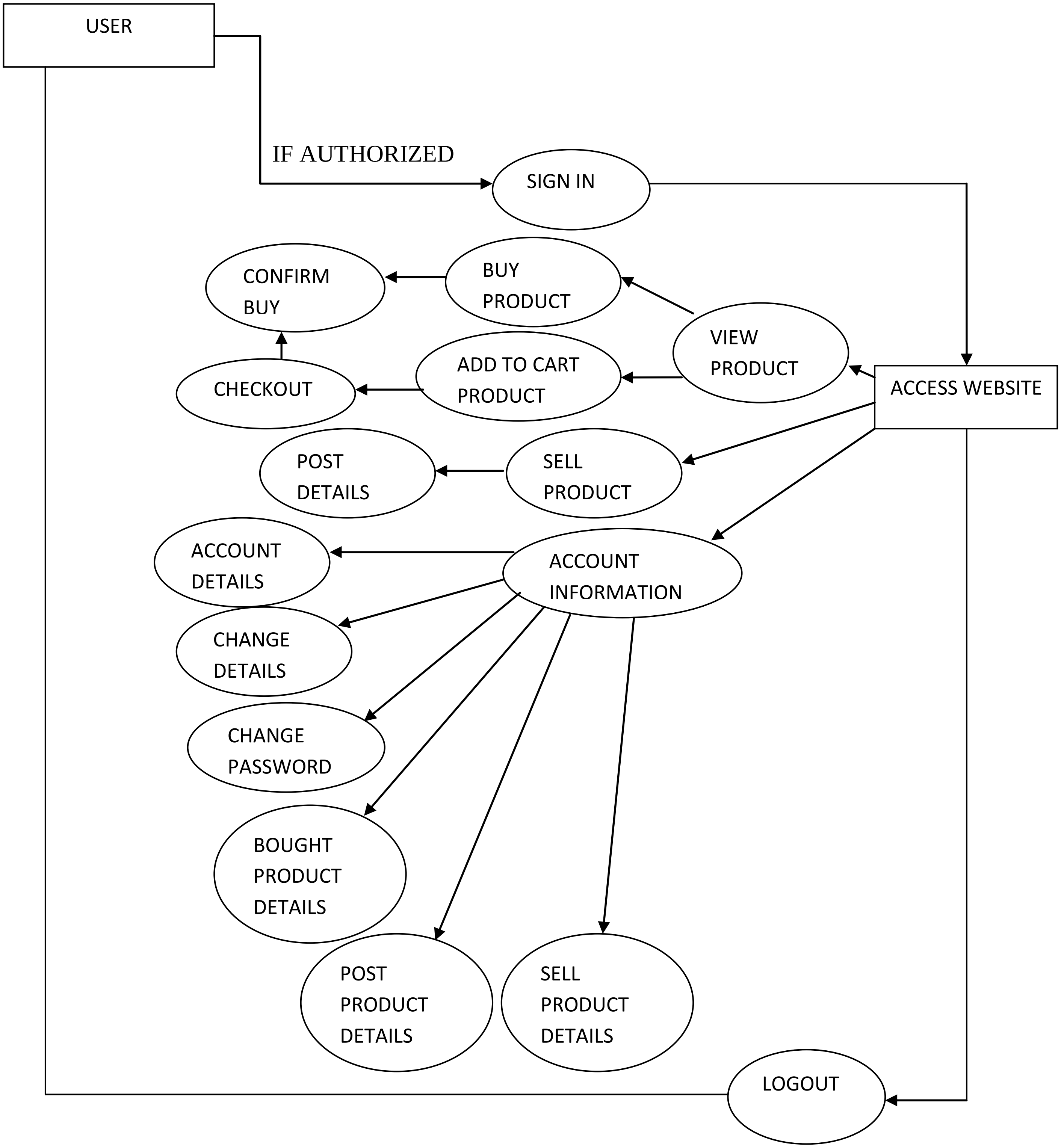


Figure 5.5: Level 2: DFD

LEVEL 3: DFD

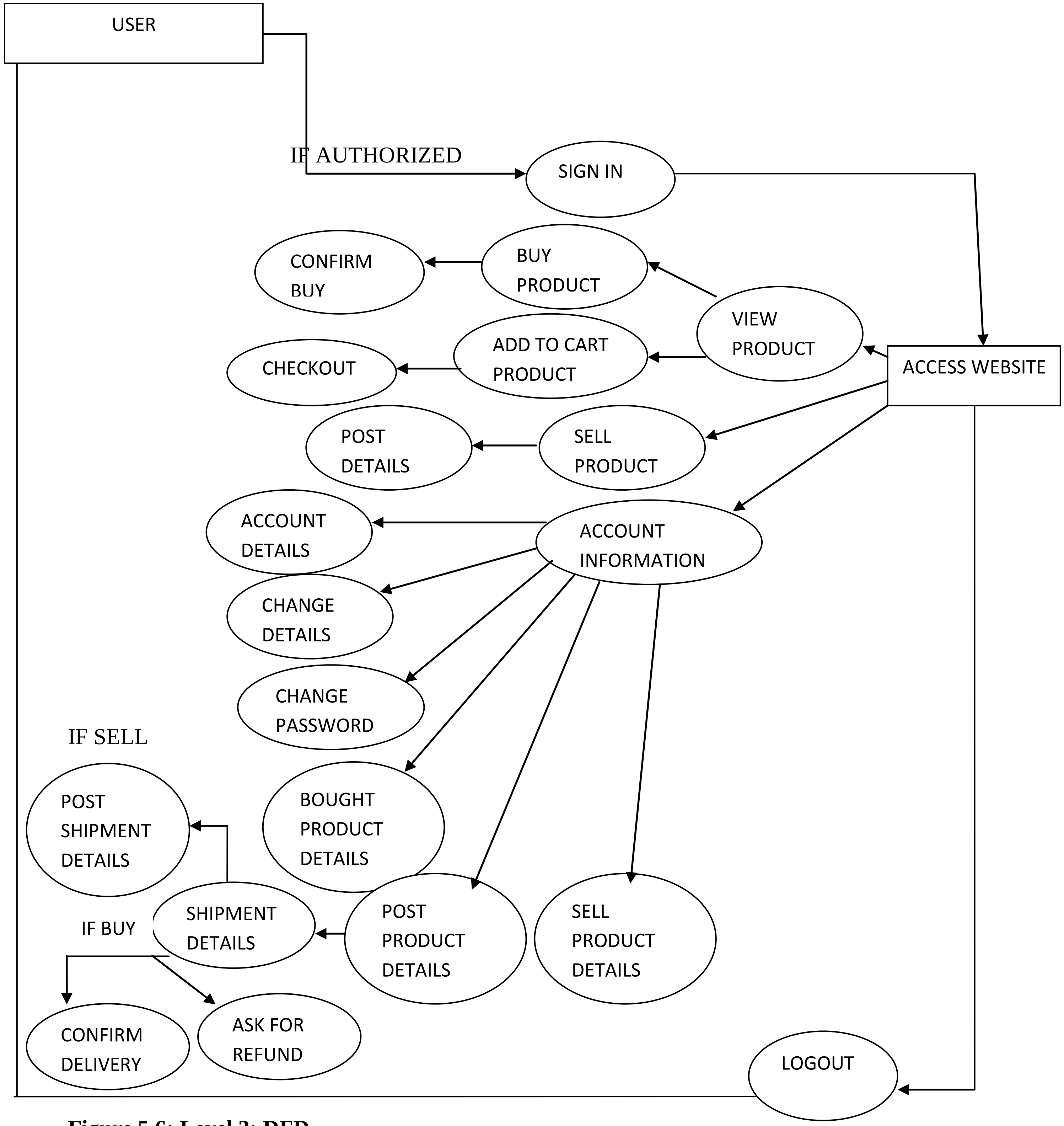


Figure 5.6: Level 3: DFD

CHAPTER 6

PROJECT CODINGS

6.1 Home Page

```
<?php
include_once 'storescripts/start.php';
?>
<!DOCTYPE HTML>
<html>
<head><title>eMart Home</title>
<link rel="stylesheet" href="styles/default.css" />
<script type="text/javascript" src="scripts/jquery-2.0.3.min.js"></script>
</head>
<body>
<?php
include_once('header.php');
?>
<div id="pageContent">
<br />
<div><img src="styles/logo.png" alt="EasyMart Logo" /></div><br />
<div id="categoryTab">
<table>
<tr>
<td><ul><li><a class="animateLink" href="#">Categories</a><div id="categoryDiv">
<?php
   $categoryQry=mysqli_query($con,"Select * from category Order By category ASC");
   while($num=mysqli_fetch_array($categoryQry)){
      $cid=$num['cid'];
      echo '<div class="categoryBlocks"><b><u>'.$num['category'].'</u></b><br /><br />';
      $subcategoryQry=mysqli_query($con,"Select * from subcategory where cid='$cid' Order
By subcategory ASC");
      while($num2=mysqli_fetch_array($subcategoryQry)){
         echo '<a href="product.php?scid='.$num2['scid'].'">'.$num2['subcategory'].'</a><br />';
      }
      echo'</div>';
   }
   ?>
   </div></li></ul></td>
         <td><form action="search.php" method="get"><input type="text" id="search"
name="search" /></td>
         <td><input type="submit" value="Search" /></form></td>
</tr>
</table></div></div></div>
</body></html>
```

6.2 Search Page

```
<?php
include_once 'storescripts/start.php';
?>
<!DOCTYPE HTML>
<html>
<head>
<title>
Sell Your Products
</title>
<script type="text/javascript" src="scripts/jquery-2.0.3.min.js"></script>
<link rel="stylesheet" href="styles/default.css" />
<link rel="stylesheet" href="styles/product.css" />
</head>
<body>
<?php include_once 'header.php';
$searchQuery=mysqli_query($con,"SELECT * FROM sell where title LIKE '%$searchTag%'
OR description LIKE '%$searchTag%' ORDER By date desc");
$searchCount=mysqli_num_rows($searchQuery);
?>
<div id="pageContent"><br />
<h3>Results found for <span style="text-decoration: underline;color: #494f6e;font-family:
ARIAL;"><?php echo ucwords($searchTag); ?></span></h3>
<?php
if($searchCount>=1){
echo '<div id="productBox">';
echo '<table id="myTable" class="tablesorter"><thead><tr><th
style="width:50%">Product</th><th>Product Information</th><th>Price</th></tr></thead>';
while($searchNum=mysqli_fetch_array($searchQuery)){
   $photo=$searchNum['photo'];
   $scidPhoto=explode("../",$photo);
   echo '<tbody><tr><td><a href="description.php?pid='.$searchNum['sid'].'"><img
style="float:left;" src="'.end($scidPhoto).'" alt="Unable To display Image"
/><span>'.$searchNum['title'].'</span></a></td><td><span id="pInfo"><span>Condition:
'.$searchNum['type'].'</span><br /><span>'.$searchNum['warrantyDuration'].'
'.$searchNum['warranty'].'</span></span></td>';
   echo '<td><span id="pPrice">Rs '.$searchNum['price'].'/-</span></td></tr></tbody>';
}
echo '</table></div><hr />';
}
?>
</div>
</body>
</html>
```

6.3 Registration Page

```
<script>
$(function(){
   $("#getdate").datepicker();
});
</script>
<script type="text/javascript" src="scripts/user.js"></script>
</head>
<body>
<?php
include_once('header.php');
?>
<div id="pageContent">
<div><span style="color: #fff;">Welcome</span></div>
<div id="rightSide">
<form name="regForm" method="post" action="backend/reg.php" onsubmit="return reg();">
<table>
<tr><td><label>New User! Register Here</label></td></tr>
<tr><td><span>First Name:</span></td><td><input type="text" name="first" /></td></tr>
<tr><td><span>Last Name:</span></td><td><input type="text" name="last" /></td></tr>
<tr><td><span>Email:</span></td><td><input type="email" name="email" /></td></tr>
<tr><td><span>Password:</span></td><td><input type="password" name="pass" /></td></tr>
<tr><td><span>Re-Enter Password:</span></td><td><input type="password" name="cpass" /></td></tr>
<tr><td><span>Address:</span></td><td><textarea style="max-height: 80px;max-width: 300px;" name="address"></textarea></td></tr>
<tr><td><span>State:</span></td><td><input type="text" name="state" /></td></tr>
<tr><td><span>City:</span></td><td><input type="text" name="city" /></td></tr>
<tr><td><span>Pin Code:</span></td><td><input type="text" name="pin" /></td></tr>
<tr><td><span>Country:</span></td><td><input type="text" name="country" /></td></tr>
<tr><td><span>Gender:</span></td><td><span><input type="radio" value="Male" name="gender" id="male" />Male<input type="radio" value="Female" name="gender" id="female" />Female</span></td></tr>
<tr><td><span>Date Of Birth:</span></td><td><input type="text" name="dob" id="getdate" /></td></tr>
<tr><td colspan="2"><span><input type="checkbox" name="tc" />I Agree to the <a href="" target="_blank">Terms And Conditions</a></span></td></tr>
<tr><td><input type="submit" value="Register" /></td></tr>
</table>
<div id="result" class="errors"></div>
</form>
</div></div>
```

6.4 Sell Page

```
<h4>Title</h4>
<input type="text" maxlength="50" name="title" id="title" /><br />
<h4>Category</h4>
<select name="categorySel" id="categorySel" onchange="subCategory()">
<option value="0">Select</option>
<?php
$categoryQry=mysqli_query($con,"Select * from category Order By category ASC");
while($num=mysqli_fetch_array($categoryQry)){
    echo '<option value="'.$num['cid'].'">'.$num['category'].'</option>';
}?></select><br />
<select id="subCategorySelect" name="subCategorySelect">
</select></div><hr />
<div id="warrantyDiv">
<h4>Type</h4>
<select id="productType" name="productType">
<option value="New">New</option>
<option value="Used">Used</option>
</select><br />
<h4>Warranty</h4>
<select id="warranty" name="warranty" onchange="warrantyCheck()">
<option value="Manufacturer Warranty">Manufacturer Warranty</option>
<option value="Seller Warranty">Seller Warranty</option>
<option value="No Warranty">No Warranty</option>
</select><br />
<h4>Warranty Duration</h4>
<select id="duration" name="duration">
<option value="6 Months">6 Months</option>
<option value="1 Year">1 Year</option>
<option value="2 Years">2 Years</option>
</select></div><hr />
<div id="photoDiv">
<h4>Product Image</h4>
<input type="file" name="photoSelect" id="photoSelect" onchange="preview(this)" /><br />
<img id="productImage" src="" />
</div><hr />
<div id="descriptionDiv">
<h4>Description</h4>
<textarea id="description" name="description" ></textarea>
</div><hr />
<div id="priceDiv">
<h4>Price</h4>
<input type="text" name="price" id="price" />(Enter In number)<br />
<h4>Quantity</h4>
<input type="text" name="quantity" id="quantity" />(Enter In number)</div></form>
```

6.5 User Shop Page

```
<div id="userInfo">
<h3><?php echo $userD['first'].' '.$userD['last'].'\'s Shop'; ?></h3><br />
<b>Contact at Email ID:<br /><?php echo ' '.$userD['email']; ?></b><br /><br /><br />
<table>
<tr><th>Products Posted</th><th>Products Sold</th></tr>
<tr><td><?php echo $countNum; ?></td><td><?php echo $countNum2; ?></td></tr>
</table>
</div>
<div id="userProducts">
<?php
$countQuery=mysqli_query($con,"SELECT * FROM sell WHERE user='$ssid'");
while($countFetch2=mysqli_fetch_array($countQuery)){
   $img=$countFetch2['photo'];
   $img1=explode("../",$img);

   $expDate=explode(" ",$countFetch2['date']);
   $covDate=date("F j",strtotime($expDate[0]));

   echo '<div id="userProd">';
   echo '<a href="description.php?pid='.$countFetch2['sid'].'"><img src="'.end($img1).'"
alt="Display Problem" /></a><br />';
   echo 'Name: <span>'.$countFetch2['title'].'</span><br />';
   echo 'Price: <span>'.$countFetch2['price'].'</span><br />';
   echo 'Posted On: <span>'.$covDate.'</span>';
   echo '</div>';
}
?>
</div>
```

6.6 Product Page

```
<div>
<?php
if($scidCount>=1){
echo '<div id="productBox">';
echo '<table id="myTable" class="tablesorter"><thead><tr><th
style="width:50%">Product</th><th>Product Information</th><th>Price</th></tr></thead>';
while($scidNum=mysqli_fetch_array($scidQuery)){
   $photo=$scidNum['photo'];
   $scidPhoto=explode("../",$photo);
   echo '<tbody><tr><td><a href="description.php?pid='.$scidNum['sid'].'"><img
style="float:left;" src="'.end($scidPhoto).'" alt="Unable To display Image"
/><span>'.$scidNum['title'].'</span></a></td><td><span id="pInfo"><span>Condition:
'.$scidNum['type'].'</span><br /><span>'.$scidNum['warrantyDuration'].'
'.$scidNum['warranty'].'</span></span></td>';
   echo '<td><span id="pPrice">Rs '.$scidNum['price'].'/-</span></td></tr></tbody>';
}
echo '</table></div><hr />';
}
else
{
   echo '<div>No Products To display Under this Category<br /><a href="sell.php">Post one
yourself now</a></div>';
}
?>
</div>
```

6.7 Shipment Page

```
<form action="backend/shipBack.php" method="post">
<table>
<tr><td><span>Product:</span></td><td><img width="60px" height="60px" src="<?php echo
end($shipmentPhoto); ?>" alt="Image Cannot be displayed" /><br /><span><?php echo
$shipmentNum2['title']; ?></span></td></tr>
<tr><td colspan="2"><input type="hidden" value="<?php echo $transId; ?>" name="transID"
/></td></tr>
<tr><td><span>Courier Company Name:</span></td><td><input type="text"
name="courierName" /></td></tr>
<tr><td><span>Shipping Date:</span></td><td><input type="text" id="courierDate"
name="courierDate" /></td></tr>
<tr><td><span>Track ID:</span></td><td><input type="text" name="courierTrack"
/></td></tr>
<tr><td><span>Website:</span></td><td><input type="text" name="courierWebsite"
/></td></tr>
<tr><td colspan="2"><input type="submit" value="Save" /></td></tr>
</table>
</form>
<form action="backend/shipBack.php" method="post">
<table>
<tr><td><span>Product:</span></td><td><img width="60px" height="60px" src="<?php echo
end($shipmentPhoto); ?>" alt="Image Cannot be displayed" /><br /><span><?php echo
$shipmentNum2['title']; ?></span></td></tr>
<tr><td colspan="2"><input type="hidden" value="<?php echo $transId; ?>" name="transID"
/></td></tr>
<tr><td><span>Courier Company Name:</span></td><td><input type="text" value="<?php
echo $shipNum['courier']; ?>" name="editCourier" /></td></tr>
<tr><td><span>Track ID:</span></td><td><input type="text" value="<?php echo
$shipNum['track']; ?>" name="editTrack" /></td></tr>
<tr><td><span>Shipping Date:</span></td><td><input type="text" value="<?php echo
$shipNum['shipDate']; ?>" id="editDate" name="editDate" /></td></tr>
<tr><td><span>Website:</span></td><td><input type="text" value="<?php echo
$shipNum['website']; ?>" name="editWebsite" /></td></tr>
<tr><td colspan="2"><input type="submit" value="Update" /></td></tr>
</table>
</form>
```

6.8 Gateway Page

```
<div id="mainDiv">
<img src="styles/payment/payment-gateway.png" width="400px" height="200px" alt="Payment Gateway" />
<table>
<tr><td><span>Payable Amount:</span></td><td>
<input type="hidden" id="hiddenAmount" value="<?php echo $ordernum; ?>" />
<input type="hidden" id="totalAmount" value="<?php echo $total; ?>" />
<?php echo "Rs ".$total."/-"; ?></td></tr>
<tr><td><span>Credit Card Number:</span></td><td><input type="text" id="credit" name="credit" /></td></tr>
<tr><td><span>Name on Card:</span></td><td><input type="text" id="card" name="card" /></td></tr>
<tr><td><span>CVV Number:</span></td><td><input type="text" id="cvv" name="cvv" /></td></tr>
<tr><td><span>Expiry Date:</span></td><td><input type="text" id="expire" name="expire" /></td></tr>
<tr><td colspan="2"><button onclick="validatePay()">Pay Now</button><button onclick="cancelPay('<?php echo $ordernum; ?>')">Cancel</button></td></tr>
</table>
<div><img width="75%" height="75%" src="styles/payment/paypal-verified.jpg" />
</div>
</div>
```

6.9 Help Page

```
<!DOCTYPE HTML>
<html>
<head>
<title>Help Page</title>
<link rel="stylesheet" href="styles/default.css" />
<link rel="stylesheet" href="styles/help.css" />
</head>
<body>
<?php include_once 'header.php'; ?>
<div id="pageContent">
<h2>EasyMart Help Page</h2>
<span>This is where you will get information of everything, you will need to know.</span>
<div id="howBuy">
<h3>How To Buy</h3>
<ol>
<li>First select a category to buy product from.</li>
<li>Browse for your choice of product.</li>
<li>If found click it and buy.</li>
<li>Confirm your product & delivery address and Click on buy</li>
<li>We currently have COD(Cash on Delivery) payment.<br />So you have to pay, when the product will be delivered to you at your doorstep.</li>
</ol>
</div>
<div id="howSell">
<h3>How To Sell</h3>
<ol>
<li>Just click on sell found on the top bar</li>
<li>Enter the specific details and upload the Image of product.</li>
<li>Click on post and that's it, you have made yourself a seller on easymart.</li>
</ol>
</div>
<h3>Note:</h3>
<span>You have to be logged in or registered to Buy/Sell</span><br />
<span>For any other info, please feel free to contact us at helpdesk@easymart.com</span>
</div>
</body>
</html>
```

6.10 Ads Div Page

```
<div id="adss">
<marquee behavior="alternate" scrolldelay="-10000">
<img id="image1" src="styles/images/ads/ads1.png" />
<img id="image2" src="styles/images/ads/ads2.jpg" />
<img id="image3" src="styles/images/ads/ads3.jpg" /></marquee>
</div>
<style>
#adss
{
    width: 820px;
    height: 120px;
    overflow: hidden;
}
#adss img
{
    width:820px;
    height: 120px;
}
</style>
```

CHAPTER 7
SYSTEM TESTING

7.1 Test Cases

Registration Form

Fields	**Errors**	**Solutions**
First Name	If left empty :- show "All fields mandatory"	Do not left the field empty
Last Name	If left empty :- show "All fields mandatory"	Do not left the field empty
Email	If left empty :- show "All fields mandatory" If invalid email type:- show "Invalid email"	Do not left the field empty Enter correct email type such as "xyz@zyx.com"
Password	If left empty :- show "All fields mandatory" If length less than 6:- show "Password length must be at least 6 characters long"	Do not left the field empty Enter password that is at least 6 characters long
Re-Enter Password	If left empty :- show "All fields mandatory" If doesn't match Password field:- show "Confirm password doesn't match"	Do not left the field empty Type same password as above field
Address	If left empty :- show "All fields mandatory"	Do not left the field empty
State	If left empty :- show "All fields mandatory"	Do not left the field empty
City	If left empty :- show "All fields mandatory"	Do not left the field empty
Pincode	If left empty :- show "All fields mandatory"	Do not left the field empty
Country	If left empty :- show "All fields mandatory"	Do not left the field empty
Gender	If left unchecked :- show "All fields mandatory"	Do not left the field empty
Date of Birth	If left empty :- show "All fields mandatory"	Do not left the field empty
Terms and Conditions	If left unselected:- show "All fields mandatory"	Do not left the field empty

Table 7.1

Login Form

Fields	**Errors**	**Solutions**
Email	If left empty :- show "Please enter both email and password" If email doesn't exist:- show "User with this email does not exist"	Do not left the field empty Enter email if you have registered before and activated your account
Password	If left empty :- show "Please enter both email and password" If password doesn't match with above email:- show "Password entered does not match"	Do not left the field empty Enter password that you used at the time of registration or when changing your password

Table 7.2

Sell Form

Fields	**Errors**	**Solutions**
Title	If left empty :- show "Please enter both email and password"	Do not left the field empty
Category	If left empty :- show "Please enter both email and password"	Do not left the field empty
Product Image	If left empty :- show "Please enter both email and password"	Do not left the field empty
Description	If left empty :- show "Please enter both email and password" If entered except numbers:- "Quantity in number only"	Do not left the field empty Enter in number only
Price	If left empty :- show "Please enter both email and password" If entered except numbers:- "Price in number only"	Do not left the field empty Enter in number only

Table 7.3

Change Account Details

Fields	**Errors**	**Solutions**
First Name	If left empty :- show "Enter First Name"	Do not left the field empty
Last Name	If left empty :- show "Enter Last Name"	Do not left the field empty
Address	If left empty :- show "Enter Address"	Do not left the field empty
State	If left empty :- show "Enter State"	Do not left the field empty
City	If left empty :- show "Enter City"	Do not left the field empty
Pin code	If left empty :- show "Enter Pin Code"	Do not left the field empty
Country	If left empty :- show "Enter Country"	Do not left the field empty
Date of Birth	If left empty :- show "Enter date of birth"	Do not left the field empty

Table 7.4

Change Password

Fields	**Errors**	**Solutions**
Old Password	If left empty :- show "All Fields Mandatory" If does not match your old password :- show "Old password doesn't match"	Do not left the field empty Enter your correct old password
New Password	If left empty :- show "All Fields Mandatory" If length less than 6 :- show "Password should be at least 6 Characters long"	Do not left the field empty Enter proper length password
Confirm Password	If left empty :- show "All Fields Mandatory" If doesn't matches above password :- show "Confirm password doesn't match"	Do not left the field empty Enter exact password as above field

Table 7.5

Buy Form

Field	**Error**	**Solution**
Address	If left unchecked and empty :- show "Enter proper address"	Do not left the field unchecked and empty

Table 7.6

Gateway Form

Fields	Errors	Solutions
Credit card number	If left empty :- show "All details mandatory" If length not equal to 16 numbers long :- show "Card number length should be 16"	Do not left the field empty Enter 16 digit card number properly
Name on card	If left empty :- show "All details mandatory"	Do not left the field empty
Cvv number	If left empty :- show "All details mandatory" If length not equal to 3 numbers long :- show "Card number length should be 3"	Do not left the field empty Enter 3 digit cvv number properly
Expiry date	If left empty :- show "All details mandatory"	Do not left the field empty

Table 7.7

Verified By Visa Form

Fields	Erros	Solutions
Password	If left empty :- show "Enter password" If length less than 6 characters :- show "Password length should 6 characters long"	Do not left the field empty Enter atleast 6 characters long password

Table 7.8

7.2 Unit /System Testing

The software literature (notably the military standards) define a unit along the lines of the smallest collection of code which can be [usefully] tested. Typically this would be a source file, a package (as in Ada), or a non-trivial object class. A hardware development analog might be a PC board. Unit Testing is just one of the levels of testing which go together to make the "big picture" of testing a system. It complements integration and system level testing. It should also complement (rather than compete with) code reviews and walkthroughs. Unit testing is generally seen as a "white box" test class. That is, it is biased to looking at and evaluating the code as implemented, rather than evaluating conformance to some set of requirements.

Importance of Unit Testing

- For any system of more than trivial complexity, it is highly inefficient and ineffective to test the system solely as a "big black box". Any attempt to do so quickly gets lost in a mire of assumptions and potential interactions. The only viable approach is to perform a hierarchy of tests, with higher level tests assuming“reasonable and consistent behaviour” by the lower level components, and separate lower level tests to demonstrate these assumptions.

- It would be infeasible to test a space shuttle as a system if you had to simultaneously question the design of every electrical component. It is similarly infeasible to test a large software system as a whole if you have to simultaneously question whether every line of code, every “if statement”, was correctly written.
- Boris Beizer has defined a progression of levels of sophistication in software testing. At the lowest level, testing is considered no different to debugging. At the higher levels, testing becomes a mindset which aims to maximise the system reliability. His approach stresses that you should “test” in the way which returns the greatest reliability improvement for resources spent rather than mindlessly performing some “theoretically neat” collection of tests.
- Experience has shown that unit-level testing (and reviewing) is very cost effective. It provides a much greater reliability improvement for resources expended than system level testing. In particular, it tends to reveal bugs which are otherwise insidious and are often catastrophic ─like the strange system crashes that occur in the field when something unusual happens.

<u>What type of documentation is required?</u>

Like the required level of formality, the appropriate level of documentation for unit testing varies from project to project, and even within a project. There may be minimum standards imposed by outside agencies, but generally there are not. The minimum requirements for the documentation are:

- It must be reviewable. That is, the records must be sufficient for others to review the adequacy of the testing.
- It must be sufficient for the tests to be repeatable. This is important for regression testing - unless you are sure you can repeat a test, you can never be sure if you have fixed the cause of a test failure. Repeatability is also important for analysing failures ─both failures during the initial testing, and subsequent failures. Knowing exactly what was and was not tested, and exactly what passed and what failed during testing is an invaluable aid in

isolating difficult-to-reproduce field failures. Repeatability not only implies the need to record in reasonable detail how the test is run and what data is used, but also implies identification of the version of code under test.

- The records must be archivable. That is, they must be sufficiently well kept and identified that they can be found if required, at a later time (perhaps years later when analysing a field failure).

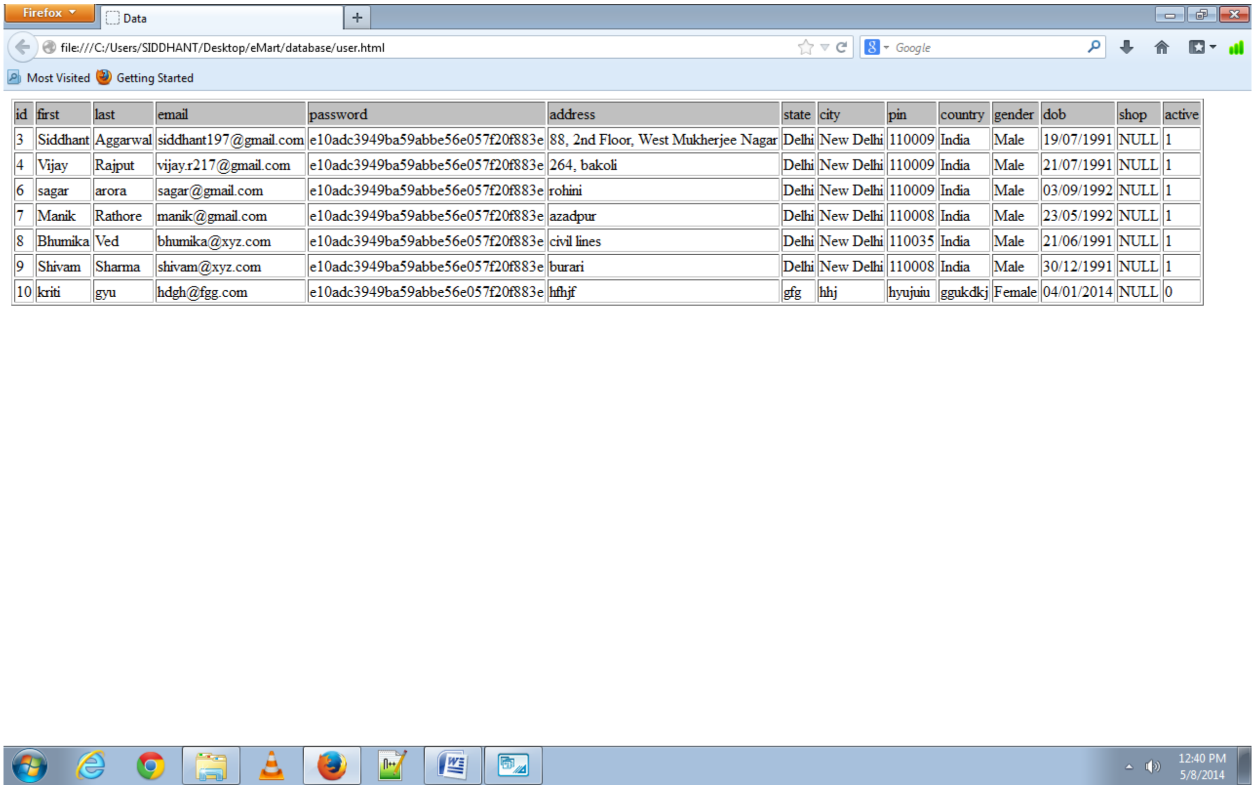
What "test environment" should be used?

- As a general rule of thumb "the rest of the system is the best test harness" for unit testing. Performing unit tests in a system environment maximises your likelihood of identifying problems. On the other hand, the tester should not allow this "rule" to limit or hinder their testing. They should use the rest of the system to generate and analyse test scenarios, but should not feel constrained from intruding into the system with debuggers, special test code, or other aids.
- Some people feel that for testing to be valid, it must be performed on exactly the code to be delivered, running exactly in its final environment. Although this is appropriate for final acceptance testing at the system level, it can actually be counter-productive at the lower levels. At the unit test level it is far preferable to "put in some debug statements" to help perform a particular test, than to avoid the test altogether in a mistaken attempt to ensure fidelity.
- It is often easy to make the system an almost ideal test harness. For example, removing restrictions on selectable system parameters when in a "system test mode" may make it trivial to force otherwise difficult "should not occur" special cases. Providing a capability to inject arbitrary byte sequence for internal messages may be trivial to implement buy extremely useful for testing. When considered early in the design process, these sorts of capabilities are often trivial to provide.

CHAPTER 8

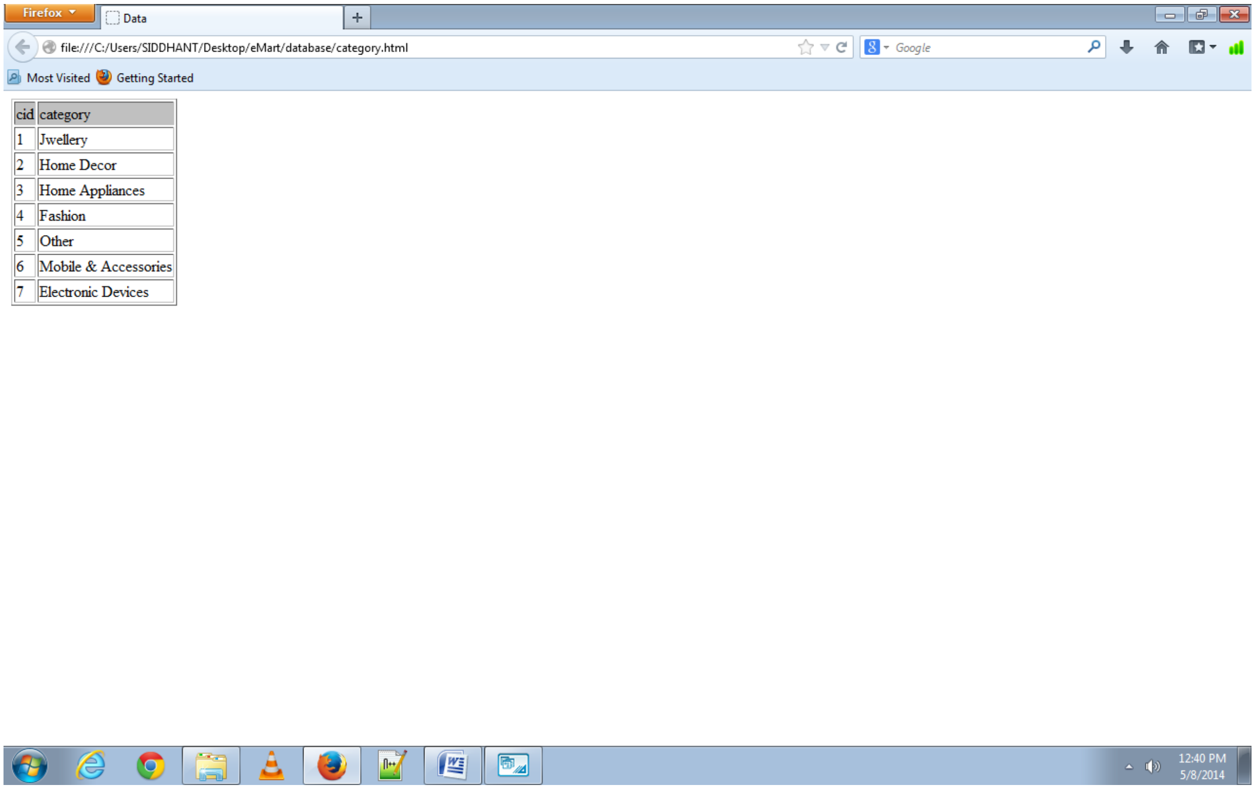
SCREENSHOTS OF THE PROJECT

8.1 DATABASE SCREENSHOTS



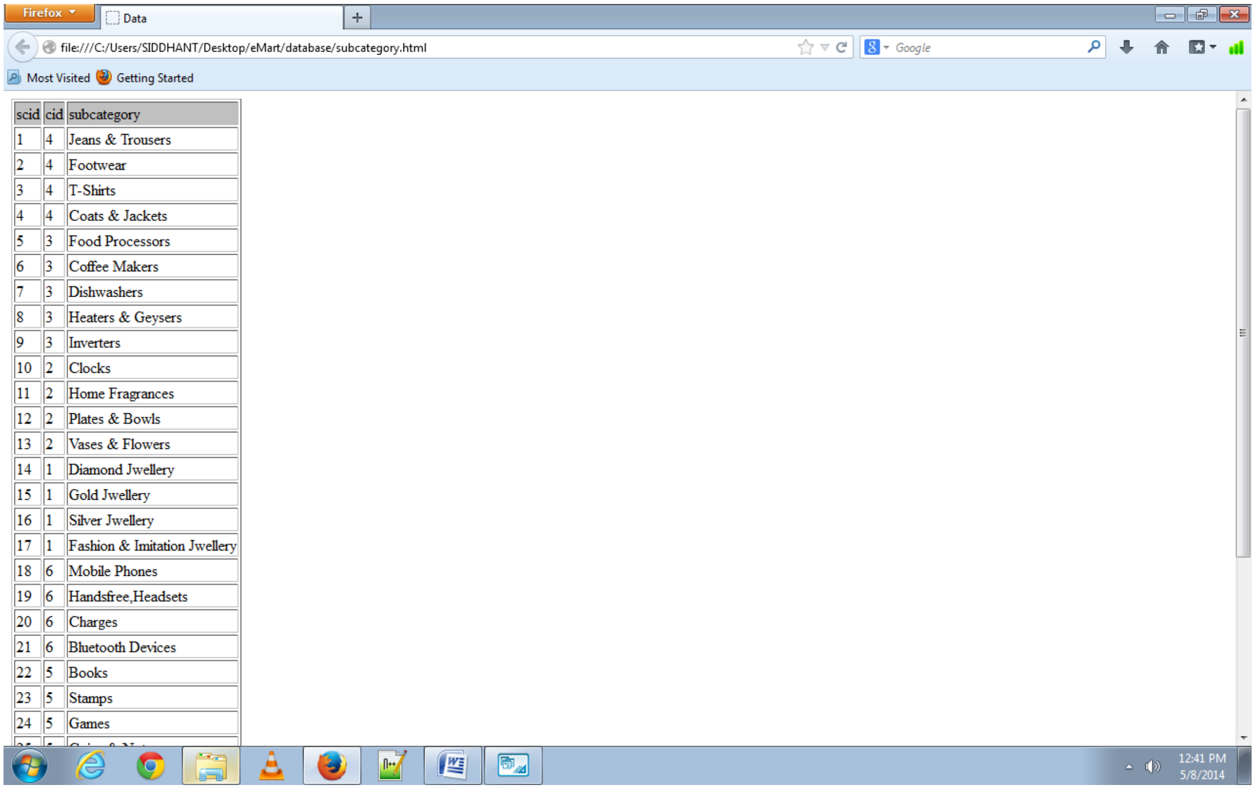
id	first	last	email	password	address	state	city	pin	country	gender	dob	shop	active
3	Siddhant	Aggarwal	siddhant197@gmail.com	e10adc3949ba59abbe56e057f20f883e	88, 2nd Floor, West Mukherjee Nagar	Delhi	New Delhi	110009	India	Male	19/07/1991	NULL	1
4	Vijay	Rajput	vijay.r217@gmail.com	e10adc3949ba59abbe56e057f20f883e	264, bakoli	Delhi	New Delhi	110009	India	Male	21/07/1991	NULL	1
6	sagar	arora	sagar@gmail.com	e10adc3949ba59abbe56e057f20f883e	rohini	Delhi	New Delhi	110009	India	Male	03/09/1992	NULL	1
7	Manik	Rathore	manik@gmail.com	e10adc3949ba59abbe56e057f20f883e	azadpur	Delhi	New Delhi	110008	India	Male	23/05/1992	NULL	1
8	Bhumika	Ved	bhumika@xyz.com	e10adc3949ba59abbe56e057f20f883e	civil lines	Delhi	New Delhi	110035	India	Male	21/06/1991	NULL	1
9	Shivam	Sharma	shivam@xyz.com	e10adc3949ba59abbe56e057f20f883e	burari	Delhi	New Delhi	110008	India	Male	30/12/1991	NULL	1
10	kriti	gyu	hdgh@fgg.com	e10adc3949ba59abbe56e057f20f883e	hfhjf	gfg	hhj	hyujuiu	ggukdkj	Female	04/01/2014	NULL	0

FIGURE 8.1: USER TABLE OF WEBSITE


Firefox
Data
file:///C:/Users/SIDDHANT/Desktop/eMart/database/category.html
Google
Most Visited
Getting Started
12:40 PM
5/8/2014

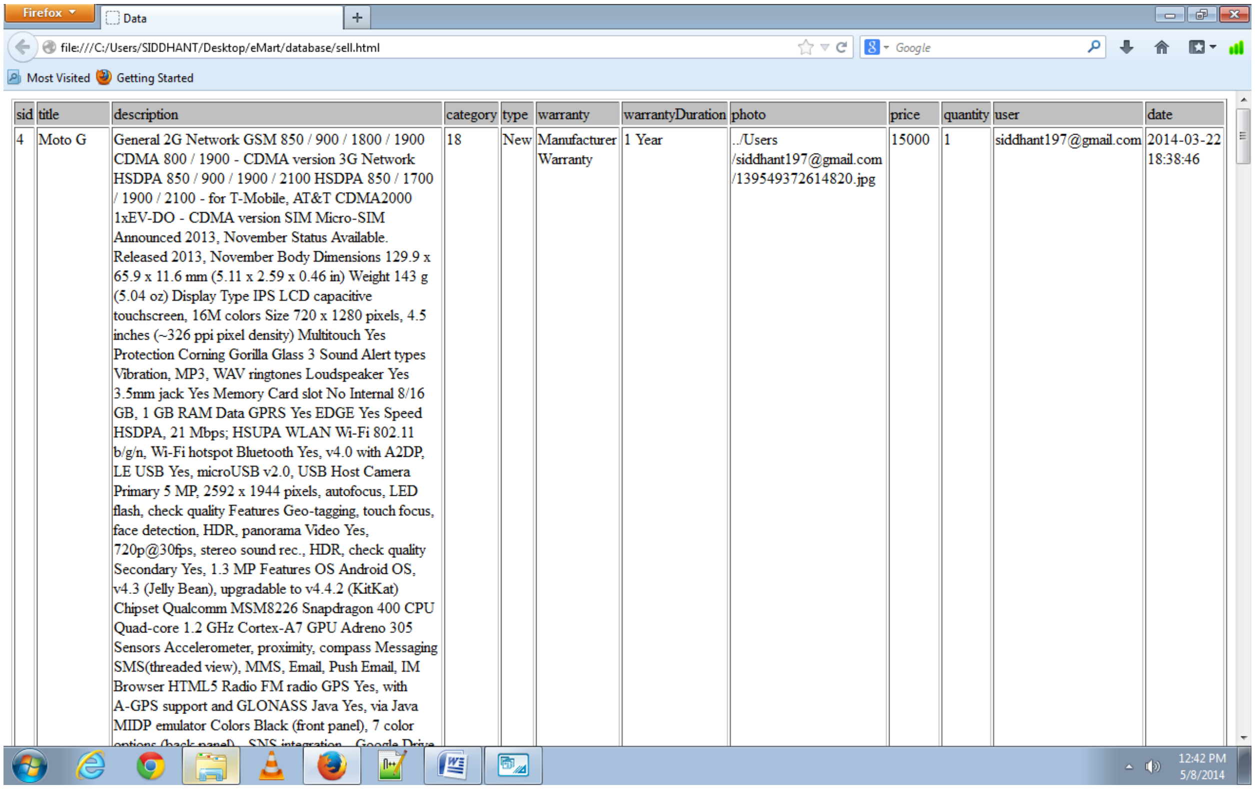
cid	category
1	Jwellery
2	Home Decor
3	Home Appliances
4	Fashion
5	Other
6	Mobile & Accessories
7	Electronic Devices

FIGURE 8.2: CATEGORY TABLE OF WEBSITE


Firefox
Data
file:///C:/Users/SIDDHANT/Desktop/eMart/database/subcategory.html
Google
Most Visited
Getting Started
12:41 PM
5/8/2014

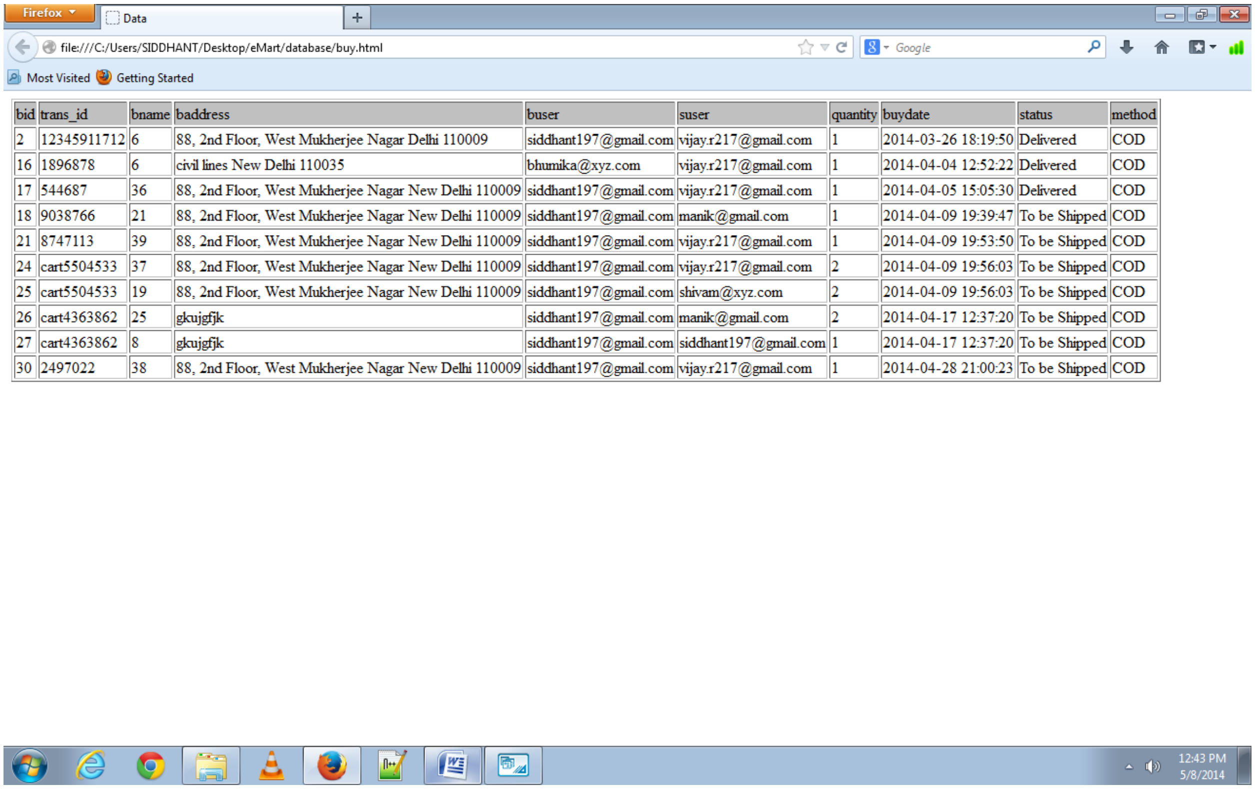
scid	cid	subcategory
1	4	Jeans & Trousers
2	4	Footwear
3	4	T-Shirts
4	4	Coats & Jackets
5	3	Food Processors
6	3	Coffee Makers
7	3	Dishwashers
8	3	Heaters & Geysers
9	3	Inverters
10	2	Clocks
11	2	Home Fragrances
12	2	Plates & Bowls
13	2	Vases & Flowers
14	1	Diamond Jwellery
15	1	Gold Jwellery
16	1	Silver Jwellery
17	1	Fashion & Imitation Jwellery
18	6	Mobile Phones
19	6	Handsfree,Headsets
20	6	Charges
21	6	Bluetooth Devices
22	5	Books
23	5	Stamps
24	5	Games

FIGURE 8.3: SUB-CATEGORY TABLE OF WEBSITE



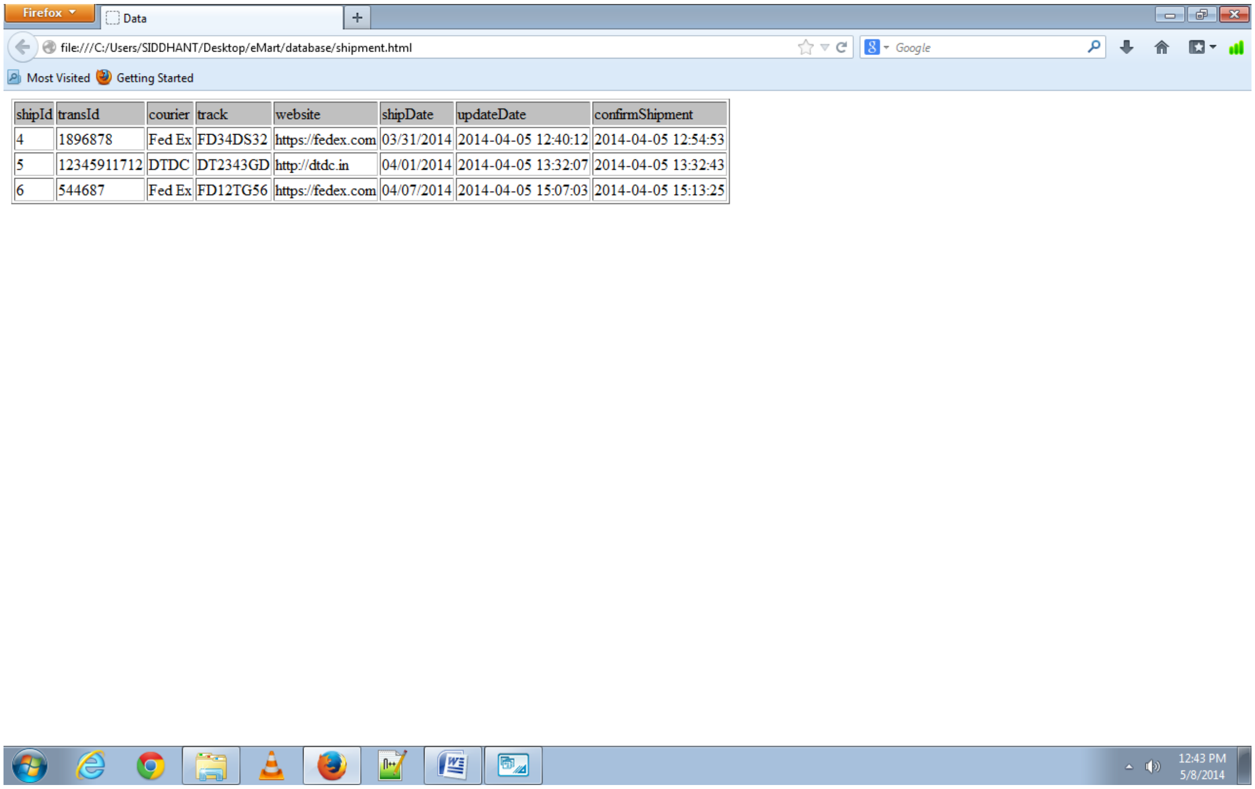
sid	title	description	category	type	warranty	warrantyDuration	photo	price	quantity	user	date
4	Moto G	General 2G Network GSM 850 / 900 / 1800 / 1900 CDMA 800 / 1900 - CDMA version 3G Network HSDPA 850 / 900 / 1900 / 2100 HSDPA 850 / 1700 / 1900 / 2100 - for T-Mobile, AT&T CDMA2000 1xEV-DO - CDMA version SIM Micro-SIM Announced 2013, November Status Available. Released 2013, November Body Dimensions 129.9 x 65.9 x 11.6 mm (5.11 x 2.59 x 0.46 in) Weight 143 g (5.04 oz) Display Type IPS LCD capacitive touchscreen, 16M colors Size 720 x 1280 pixels, 4.5 inches (~326 ppi pixel density) Multitouch Yes Protection Corning Gorilla Glass 3 Sound Alert types Vibration, MP3, WAV ringtones Loudspeaker Yes 3.5mm jack Yes Memory Card slot No Internal 8/16 GB, 1 GB RAM Data GPRS Yes EDGE Yes Speed HSDPA, 21 Mbps; HSUPA WLAN Wi-Fi 802.11 b/g/n, Wi-Fi hotspot Bluetooth Yes, v4.0 with A2DP, LE USB Yes, microUSB v2.0, USB Host Camera Primary 5 MP, 2592 x 1944 pixels, autofocus, LED flash, check quality Features Geo-tagging, touch focus, face detection, HDR, panorama Video Yes, 720p@30fps, stereo sound rec., HDR, check quality Secondary Yes, 1.3 MP Features OS Android OS, v4.3 (Jelly Bean), upgradable to v4.4.2 (KitKat) Chipset Qualcomm MSM8226 Snapdragon 400 CPU Quad-core 1.2 GHz Cortex-A7 GPU Adreno 305 Sensors Accelerometer, proximity, compass Messaging SMS(threaded view), MMS, Email, Push Email, IM Browser HTML5 Radio FM radio GPS Yes, with A-GPS support and GLONASS Java Yes, via Java MIDP emulator Colors Black (front panel), 7 color options (back panel) - SNS integration - Google Drive	18	New	Manufacturer Warranty	1 Year	../Users /siddhant197@gmail.com /139549372614820.jpg	15000	1	siddhant197@gmail.com	2014-03-22 18:38:46

FIGURE 8.4: SELL TABLE OF WEBSITE



bid	trans_id	bname	baddress	buser	suser	quantity	buydate	status	method
2	12345911712	6	88, 2nd Floor, West Mukherjee Nagar Delhi 110009	siddhant197@gmail.com	vijay.r217@gmail.com	1	2014-03-26 18:19:50	Delivered	COD
16	1896878	6	civil lines New Delhi 110035	bhumika@xyz.com	vijay.r217@gmail.com	1	2014-04-04 12:52:22	Delivered	COD
17	544687	36	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	vijay.r217@gmail.com	1	2014-04-05 15:05:30	Delivered	COD
18	9038766	21	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	manik@gmail.com	1	2014-04-09 19:39:47	To be Shipped	COD
21	8747113	39	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	vijay.r217@gmail.com	1	2014-04-09 19:53:50	To be Shipped	COD
24	cart5504533	37	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	vijay.r217@gmail.com	2	2014-04-09 19:56:03	To be Shipped	COD
25	cart5504533	19	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	shivam@xyz.com	2	2014-04-09 19:56:03	To be Shipped	COD
26	cart4363862	25	gkujgfjk	siddhant197@gmail.com	manik@gmail.com	2	2014-04-17 12:37:20	To be Shipped	COD
27	cart4363862	8	gkujgfjk	siddhant197@gmail.com	siddhant197@gmail.com	1	2014-04-17 12:37:20	To be Shipped	COD
30	2497022	38	88, 2nd Floor, West Mukherjee Nagar New Delhi 110009	siddhant197@gmail.com	vijay.r217@gmail.com	1	2014-04-28 21:00:23	To be Shipped	COD

FIGURE 8.5: BUY TABLE OF WEBSITE

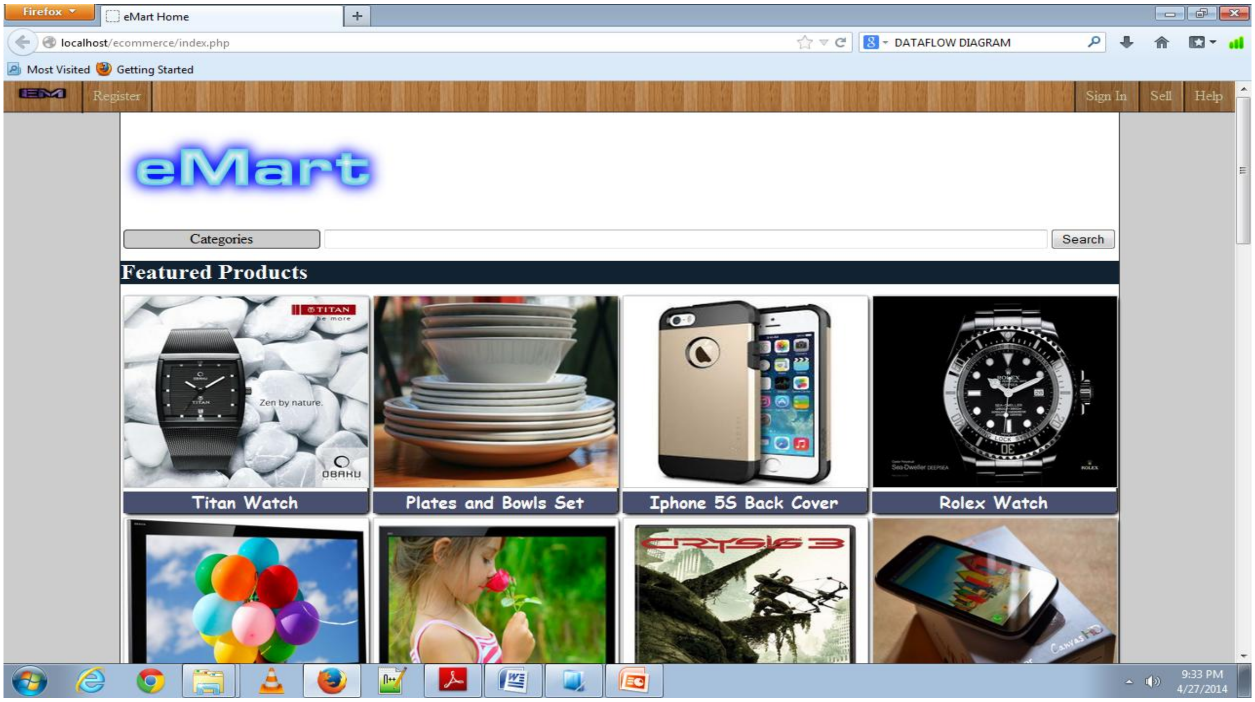


Firefox
Data
file:///C:/Users/SIDDHANT/Desktop/eMart/database/shipment.html
Google
Most Visited
Getting Started
12:43 PM
5/8/2014

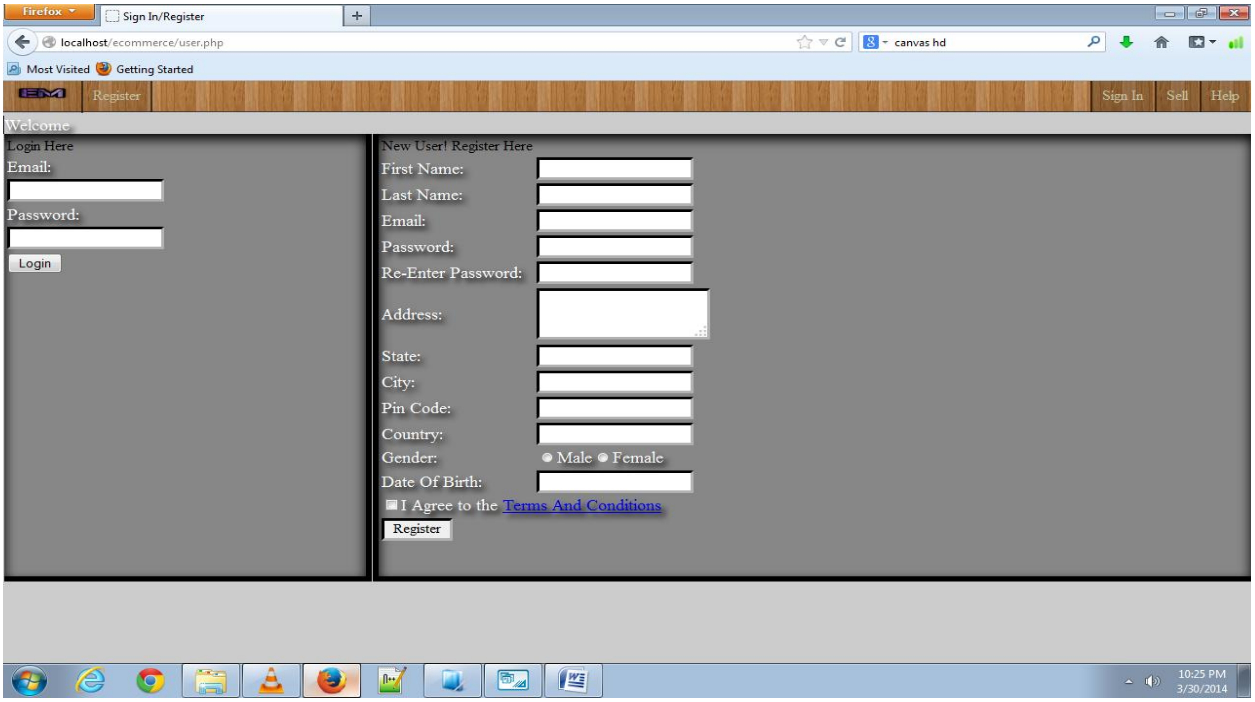
shipId	transId	courier	track	website	shipDate	updateDate	confirmShipment
4	1896878	Fed Ex	FD34DS32	https://fedex.com	03/31/2014	2014-04-05 12:40:12	2014-04-05 12:54:53
5	12345911712	DTDC	DT2343GD	http://dtdc.in	04/01/2014	2014-04-05 13:32:07	2014-04-05 13:32:43
6	544687	Fed Ex	FD12TG56	https://fedex.com	04/07/2014	2014-04-05 15:07:03	2014-04-05 15:13:25

<u>FIGURE 8.6</u>: SHIPMENT TABLE OF WEBSITE

8.2 WEBPAGES SCREENSHOTS



<u>FIGURE 8.7</u>: HOME PAGE OF WEBSITE



Welcome

Login Here

Email:

Password:

Login

New User! Register Here

First Name:

Last Name:

Email:

Password:

Re-Enter Password:

Address:

State:

City:

Pin Code:

Country:

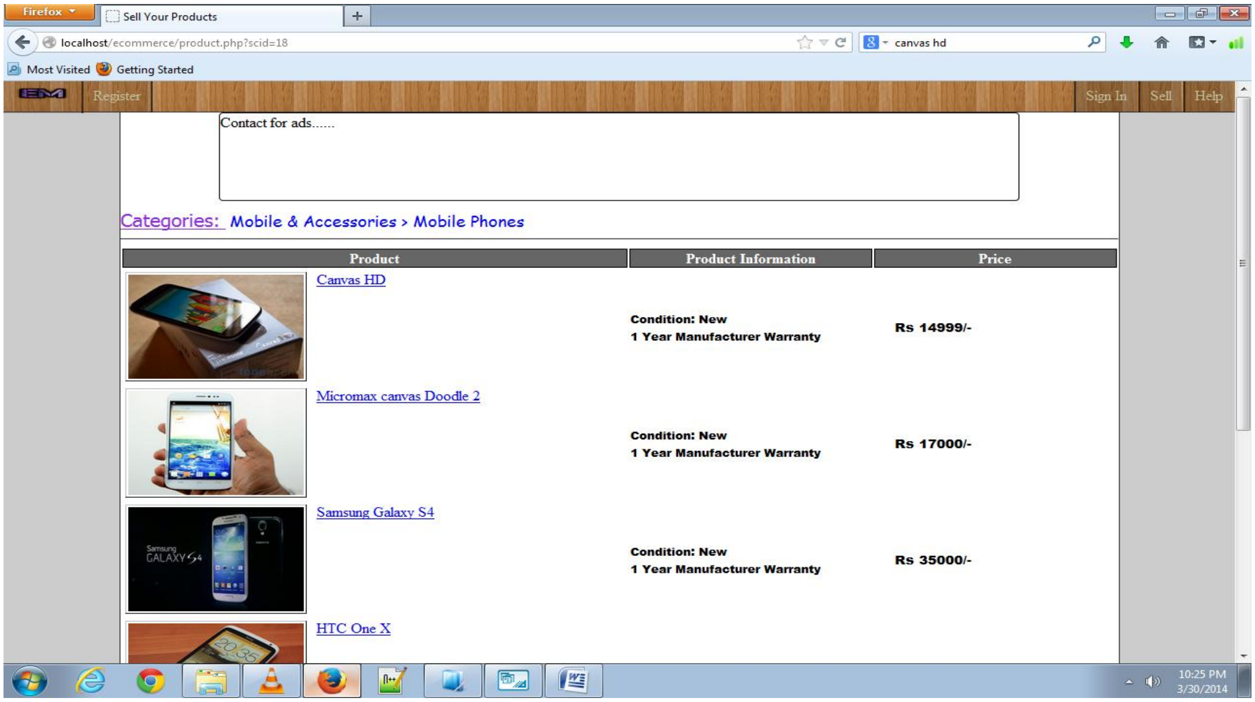
Gender: ○ Male ○ Female

Date Of Birth:

□ I Agree to the Terms And Conditions

Register

FIGURE 8.8: REGISTRATION PAGE OF WEBSITE

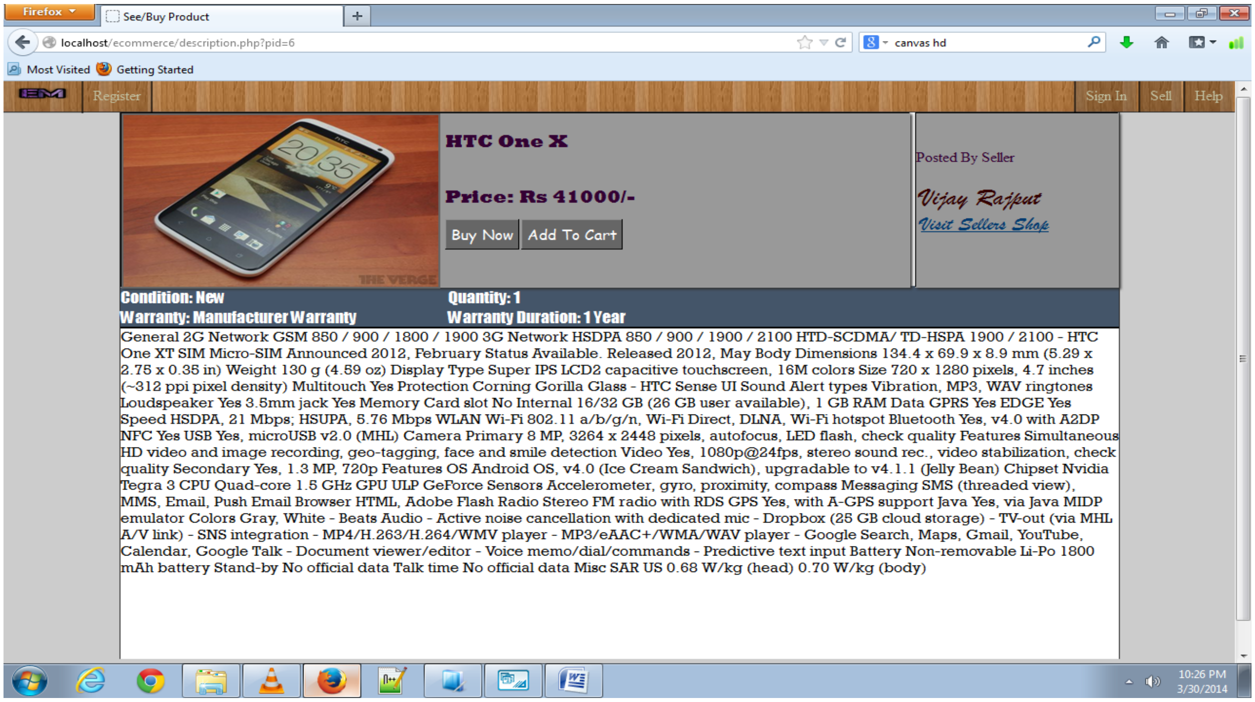


Contact for ads......

Categories: Mobile & Accessories > Mobile Phones

Product	Product Information	Price
Canvas HD	Condition: New 1 Year Manufacturer Warranty	Rs 14999/-
Micromax canvas Doodle 2	Condition: New 1 Year Manufacturer Warranty	Rs 17000/-
Samsung Galaxy S4	Condition: New 1 Year Manufacturer Warranty	Rs 35000/-
HTC One X		

FIGURE 8.9: PRODUCT DISPLAY PAGE OF WEBSITE



HTC One X

Price: Rs 41000/-

Posted By Seller

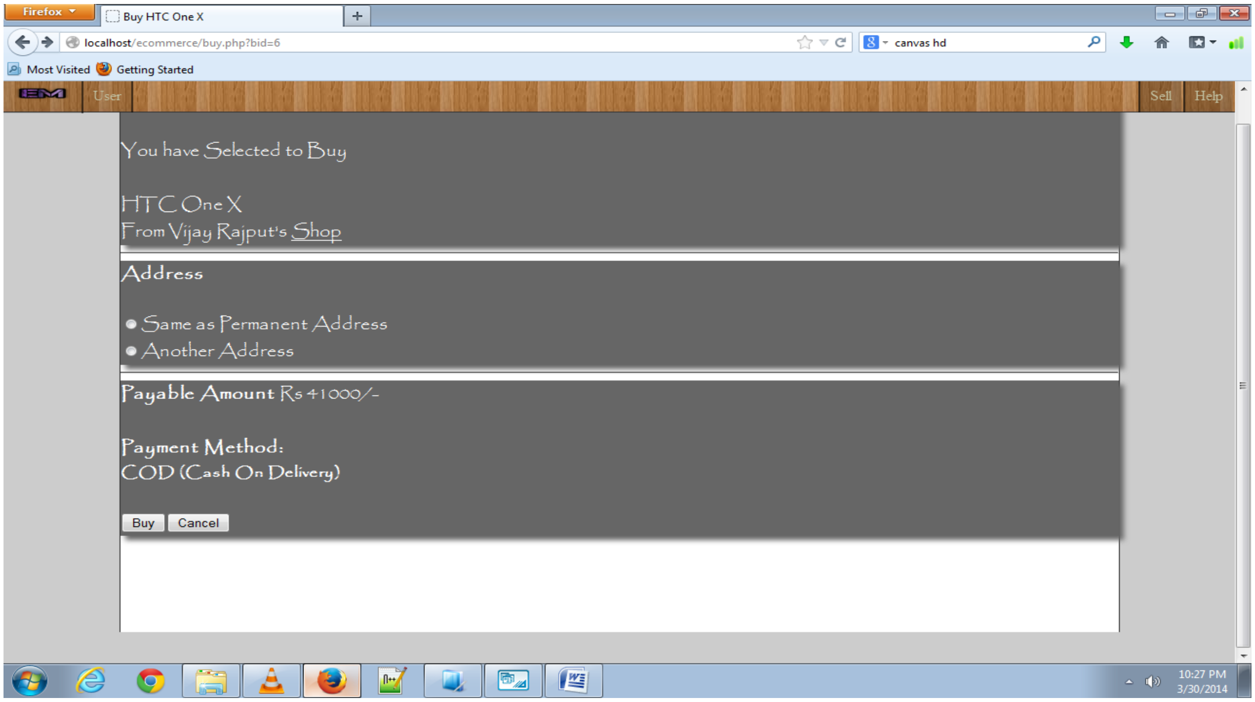
Vijay Rajput

Condition: New
Warranty: Manufacturer Warranty

Quantity: 1
Warranty Duration: 1 Year

General 2G Network GSM 850 / 900 / 1800 / 1900 3G Network HSDPA 850 / 900 / 1900 / 2100 HTD-SCDMA/ TD-HSPA 1900 / 2100 - HTC One XT SIM Micro-SIM Announced 2012, February Status Available. Released 2012, May Body Dimensions 134.4 x 69.9 x 8.9 mm (5.29 x 2.75 x 0.35 in) Weight 130 g (4.59 oz) Display Type Super IPS LCD2 capacitive touchscreen, 16M colors Size 720 x 1280 pixels, 4.7 inches (~312 ppi pixel density) Multitouch Yes Protection Corning Gorilla Glass - HTC Sense UI Sound Alert types Vibration, MP3, WAV ringtones Loudspeaker Yes 3.5mm jack Yes Memory Card slot No Internal 16/32 GB (26 GB user available), 1 GB RAM Data GPRS Yes EDGE Yes Speed HSDPA, 21 Mbps; HSUPA, 5.76 Mbps WLAN Wi-Fi 802.11 a/b/g/n, Wi-Fi Direct, DLNA, Wi-Fi hotspot Bluetooth Yes, v4.0 with A2DP NFC Yes USB Yes, microUSB v2.0 (MHL) Camera Primary 8 MP, 3264 x 2448 pixels, autofocus, LED flash, check quality Features Simultaneous HD video and image recording, geo-tagging, face and smile detection Video Yes, 1080p@24fps, stereo sound rec., video stabilization, check quality Secondary Yes, 1.3 MP, 720p Features OS Android OS, v4.0 (Ice Cream Sandwich), upgradable to v4.1.1 (Jelly Bean) Chipset Nvidia Tegra 3 CPU Quad-core 1.5 GHz GPU ULP GeForce Sensors Accelerometer, gyro, proximity, compass Messaging SMS (threaded view), MMS, Email, Push Email Browser HTML, Adobe Flash Radio Stereo FM radio with RDS GPS Yes, with A-GPS support Java Yes, via Java MIDP emulator Colors Gray, White - Beats Audio - Active noise cancellation with dedicated mic - Dropbox (25 GB cloud storage) - TV-out (via MHL A/V link) - SNS integration - MP4/H.263/H.264/WMV player - MP3/eAAC+/WMA/WAV player - Google Search, Maps, Gmail, YouTube, Calendar, Google Talk - Document viewer/editor - Voice memo/dial/commands - Predictive text input Battery Non-removable Li-Po 1800 mAh battery Stand-by No official data Talk time No official data Misc SAR US 0.68 W/kg (head) 0.70 W/kg (body)

FIGURE 8.10: PRODUCT DESCRIPTION PAGE OF WEBSITE



You have Selected to Buy

HTC One X
From Vijay Rajput's Shop

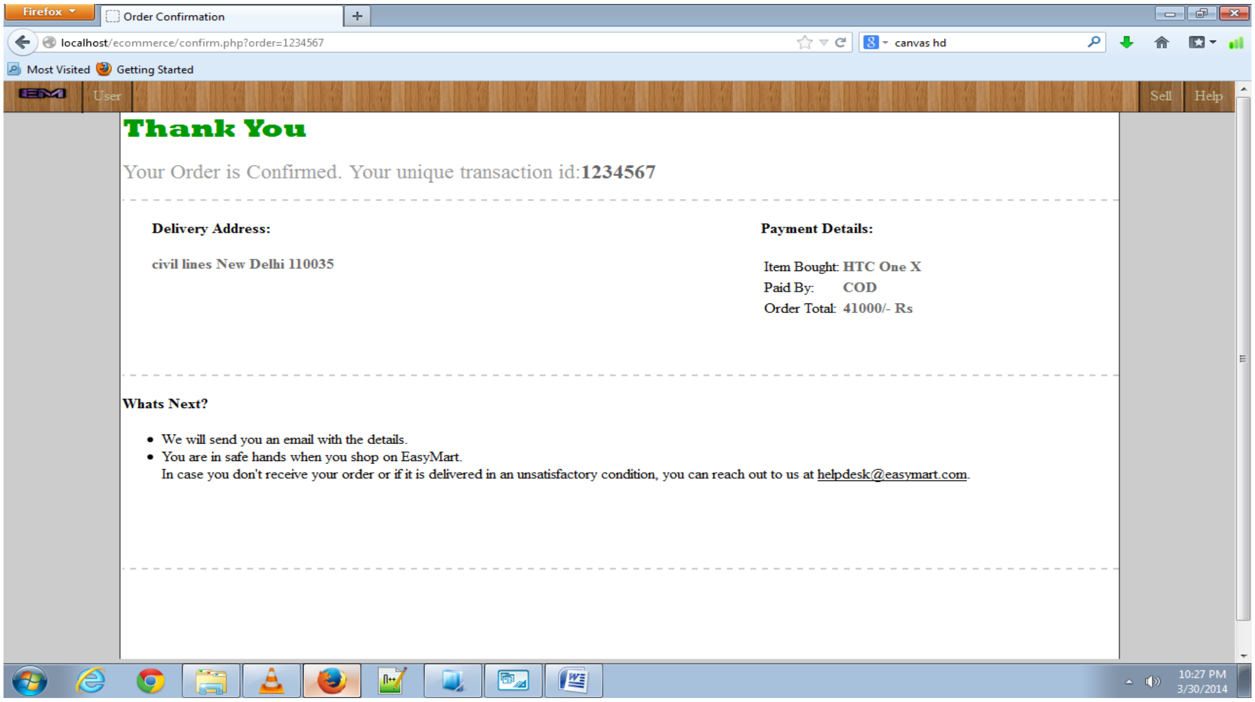
Address

● Same as Permanent Address
● Another Address

Payable Amount Rs 41000/-

Payment Method:
COD (Cash On Delivery)

FIGURE 8.11: BUY CONFIRM PAGE OF WEBSITE



Thank You

Your Order is Confirmed. Your unique transaction id:**1234567**

Delivery Address:

civil lines New Delhi 110035

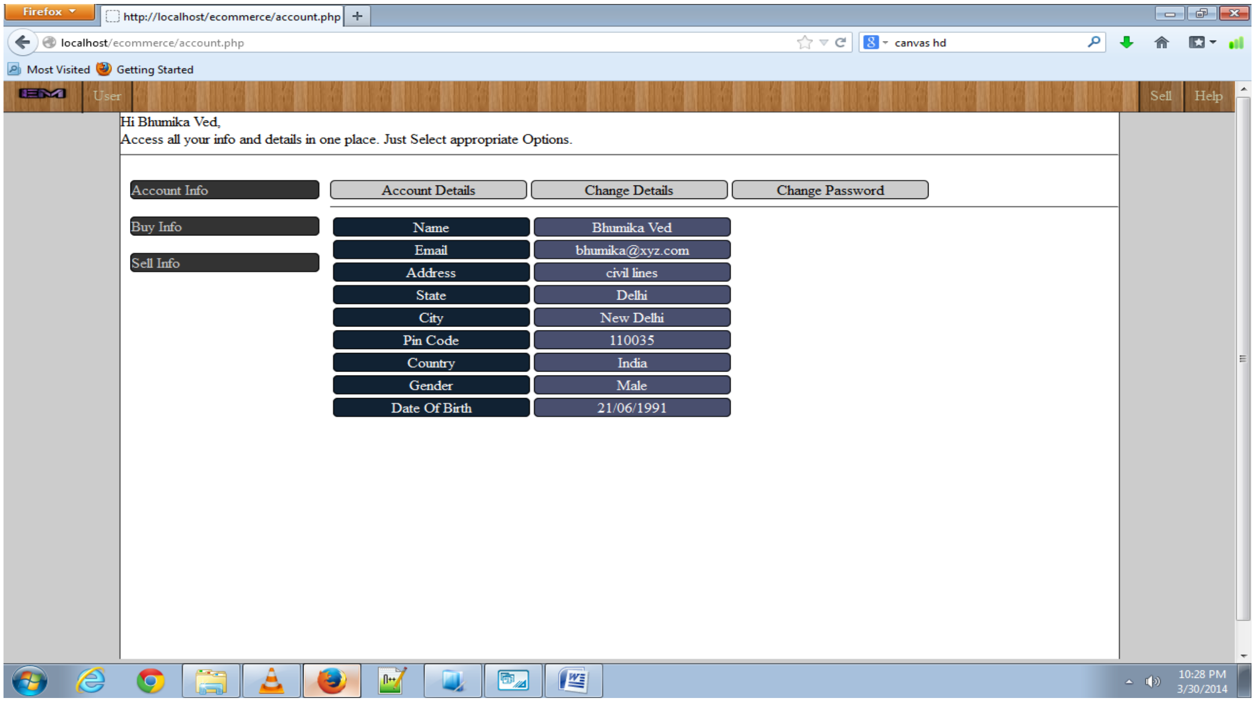
Payment Details:

Item Bought: HTC One X
Paid By: COD
Order Total: 41000/- Rs

Whats Next?

- We will send you an email with the details.
- You are in safe hands when you shop on EasyMart.
  In case you don't receive your order or if it is delivered in an unsatisfactory condition, you can reach out to us at helpdesk@easymart.com.

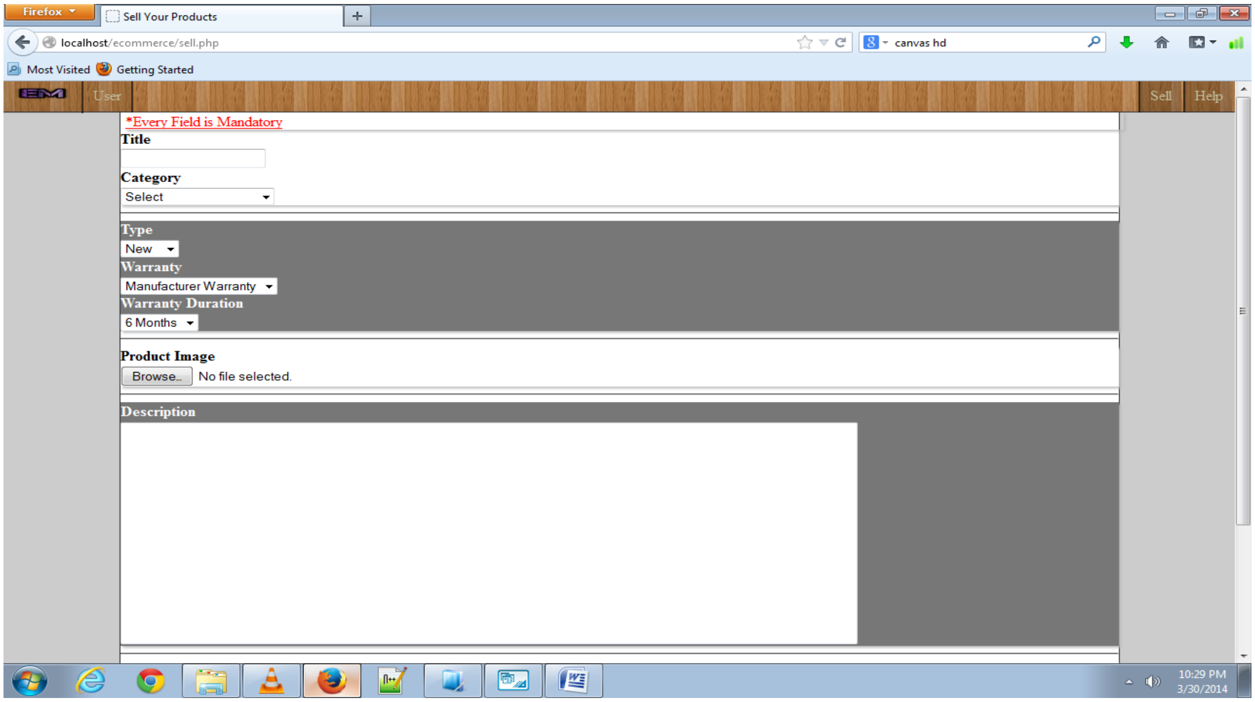
FIGURE 8.12: BUY PAGE OF WEBSITE



Hi Bhumika Ved,
Access all your info and details in one place. Just Select appropriate Options.

Name	Bhumika Ved
Email	bhumika@xyz.com
Address	civil lines
State	Delhi
City	New Delhi
Pin Code	110035
Country	India
Gender	Male
Date Of Birth	21/06/1991

FIGURE 8.13: ACCOUNT PAGE OF USER



*Every Field is Mandatory

Title

Category
Select

Type
New

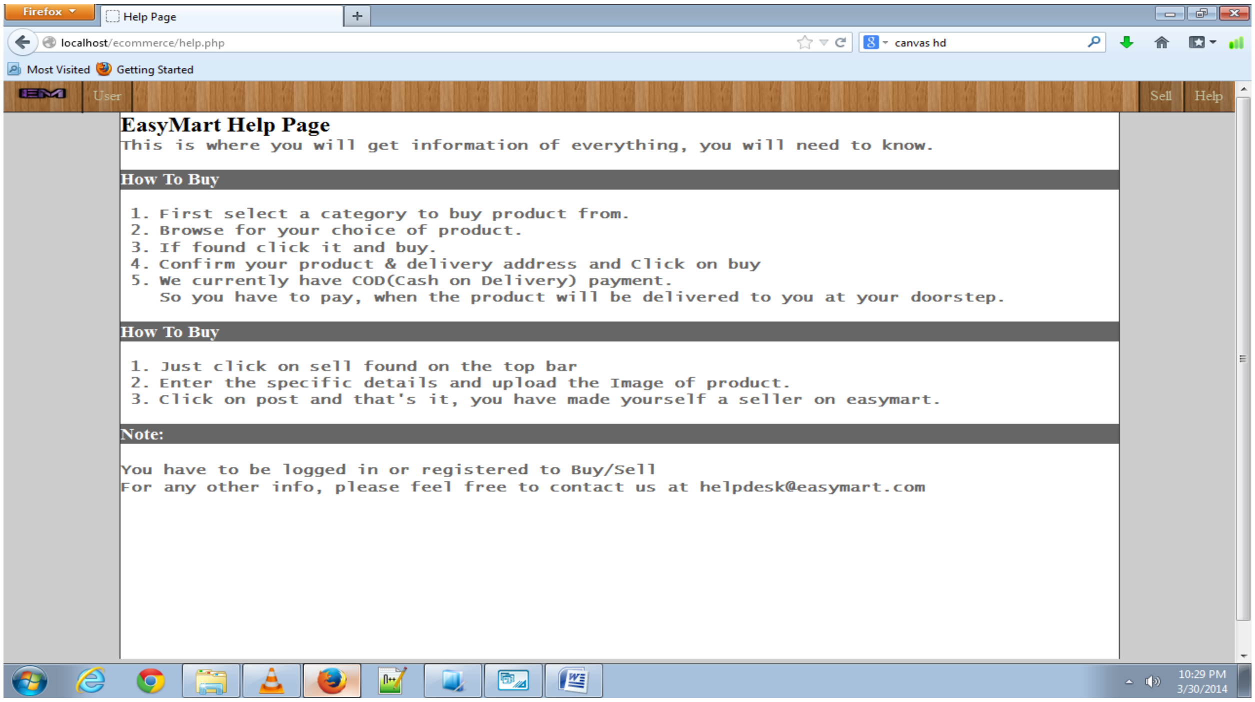
Warranty
Manufacturer Warranty

Warranty Duration
6 Months

Product Image
Browse... No file selected.

Description

FIGURE 8.14: SELL PAGE FOR USER



EasyMart Help Page

This is where you will get information of everything, you will need to know.

How To Buy

1. First select a category to buy product from.
2. Browse for your choice of product.
3. If found click it and buy.
4. Confirm your product & delivery address and Click on buy
5. We currently have COD(Cash on Delivery) payment.
   So you have to pay, when the product will be delivered to you at your doorstep.

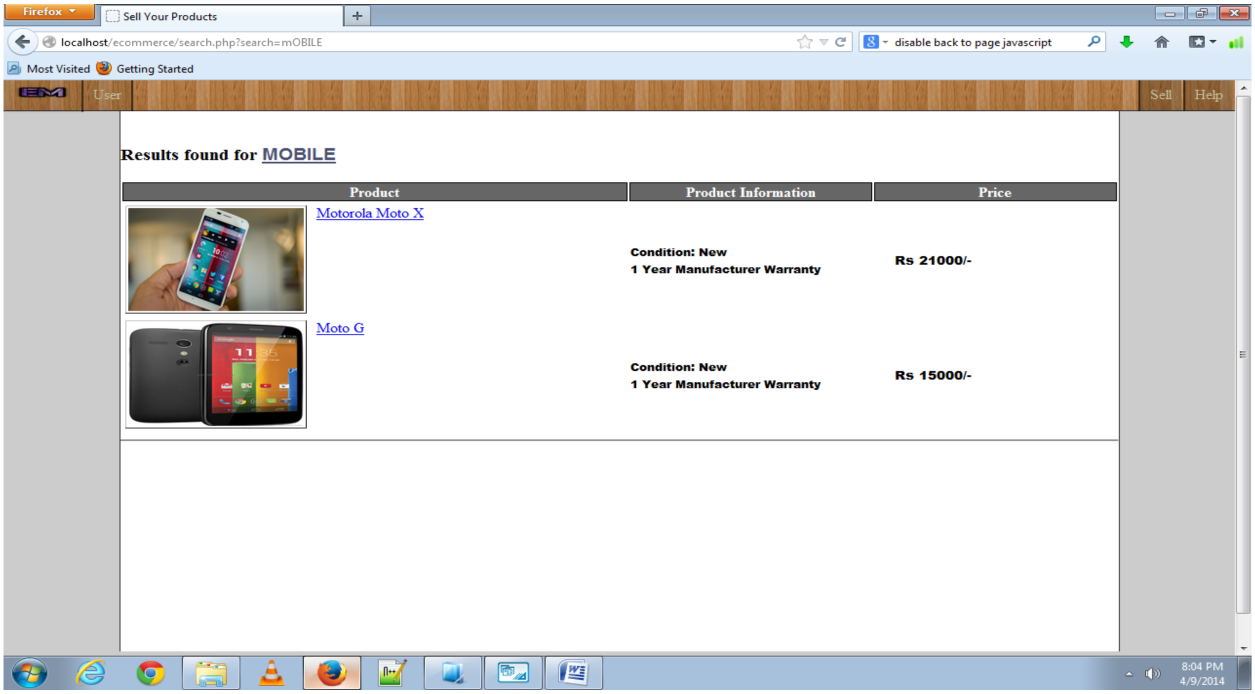
How To Buy

1. Just click on sell found on the top bar
2. Enter the specific details and upload the Image of product.
3. Click on post and that's it, you have made yourself a seller on easymart.

Note:

You have to be logged in or registered to Buy/Sell
For any other info, please feel free to contact us at helpdesk@easymart.com

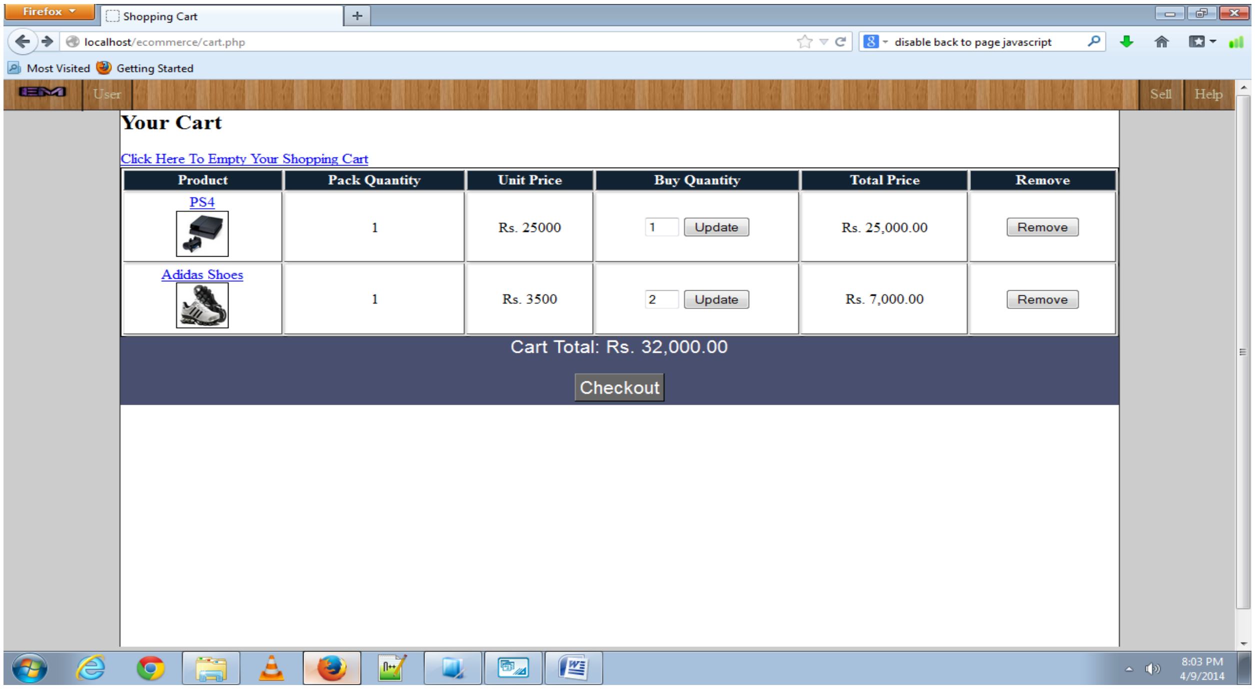
FIGURE 8.15: HELP PAGE FOR USER



Results found for MOBILE

Product	Product Information	Price
Motorola Moto X	Condition: New 1 Year Manufacturer Warranty	Rs 21000/-
Moto G	Condition: New 1 Year Manufacturer Warranty	Rs 15000/-

FIGURE 8.16: RESULT PAGE FOR SEARCH



Your Cart

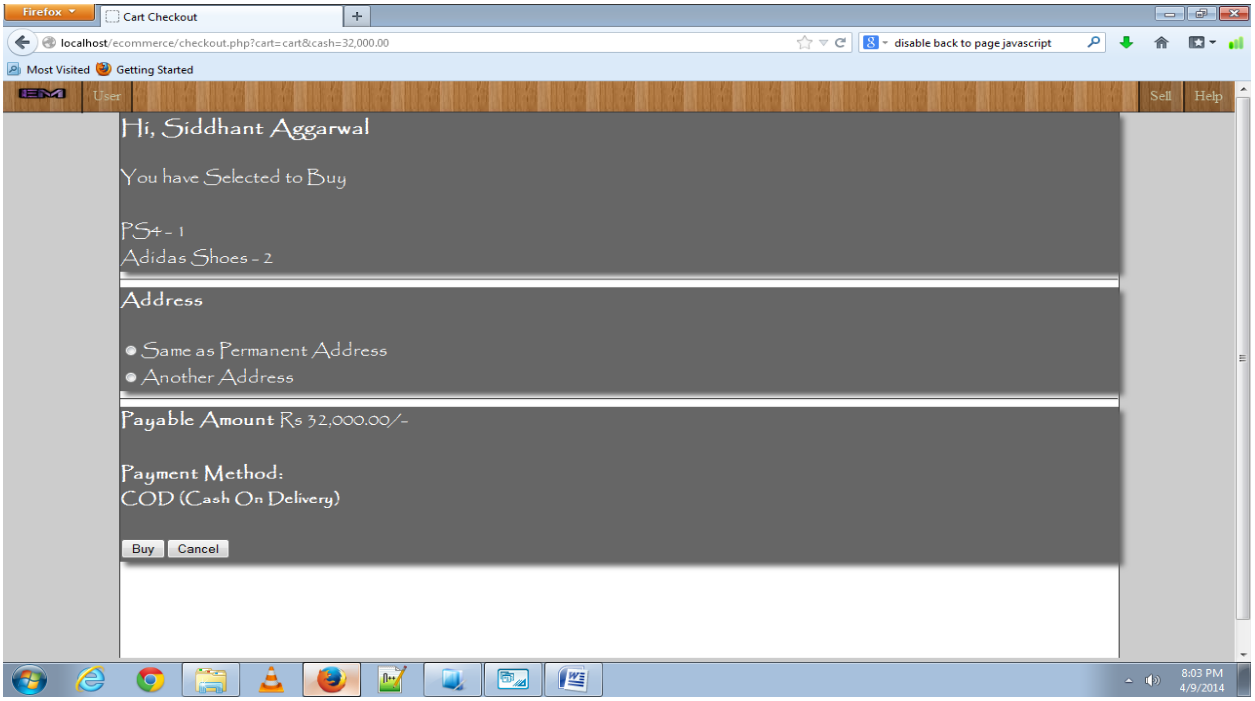
Click Here To Empty Your Shopping Cart

Product	Pack Quantity	Unit Price	Buy Quantity	Total Price	Remove
PS4	1	Rs. 25000	1 Update	Rs. 25,000.00	Remove
Adidas Shoes	1	Rs. 3500	2 Update	Rs. 7,000.00	Remove

Cart Total: Rs. 32,000.00

Checkout

FIGURE 8.17: ADD TO CART PAGE FOR USER



Hi, Siddhant Aggarwal

You have Selected to Buy

PS4 - 1
Adidas Shoes - 2

Address

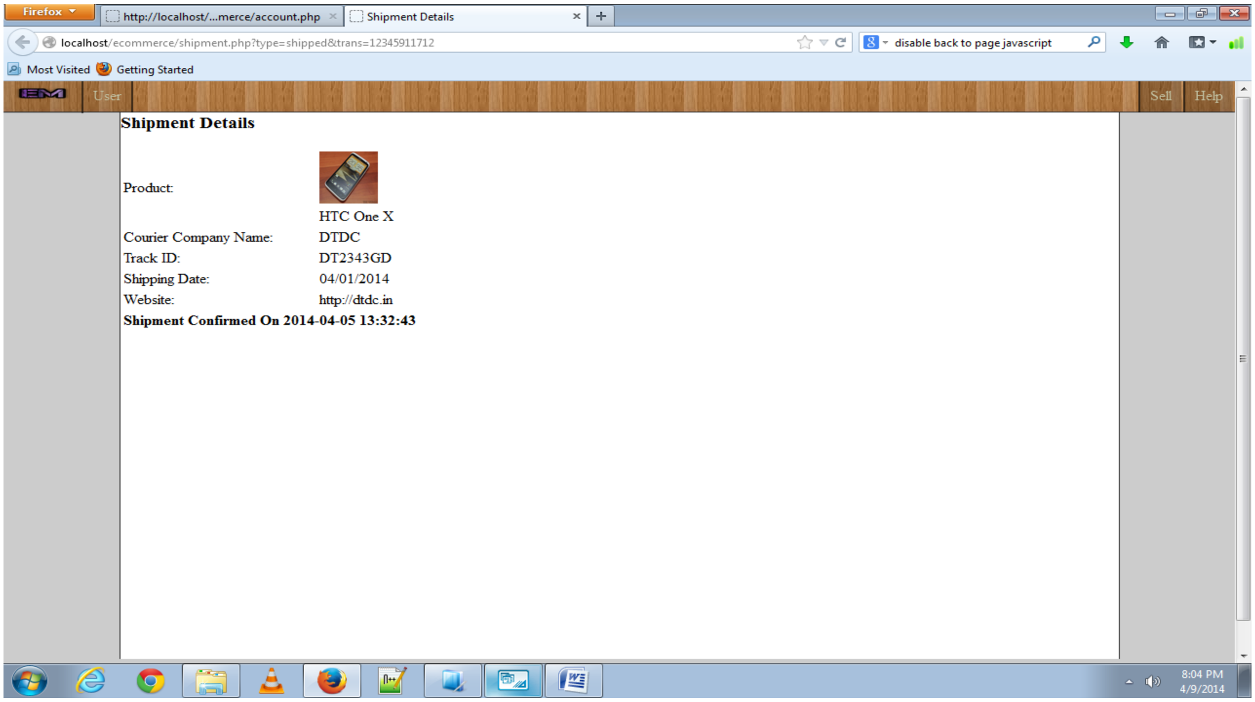
○ Same as Permanent Address
○ Another Address

Payable Amount Rs 32,000.00/-

Payment Method:
COD (Cash On Delivery)

Cart Checkout
localhost/ecommerce/checkout.php?cart=cart&cash=32,000.00
disable back to page javascript
Most Visited
Getting Started
User
Sell
Help
Buy
Cancel
8:03 PM
4/9/2014

FIGURE 8.18: CHECKOUT PAGE FOR CART



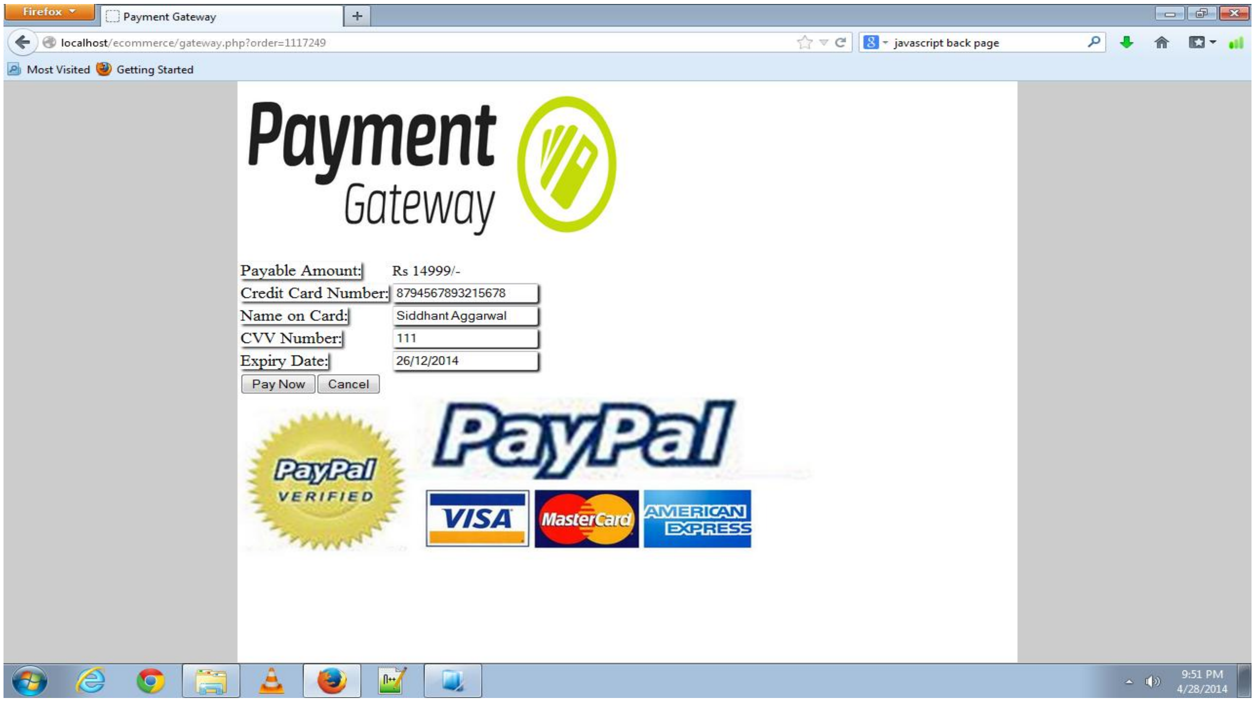
Firefox
http://localhost/...merce/account.php
Shipment Details
localhost/ecommerce/shipment.php?type=shipped&trans=12345911712
disable back to page javascript
Most Visited
Getting Started
User
Sell
Help
8:04 PM
4/9/2014

Shipment Details

Product:	HTC One X
Courier Company Name:	DTDC
Track ID:	DT2343GD
Shipping Date:	04/01/2014
Website:	http://dtdc.in

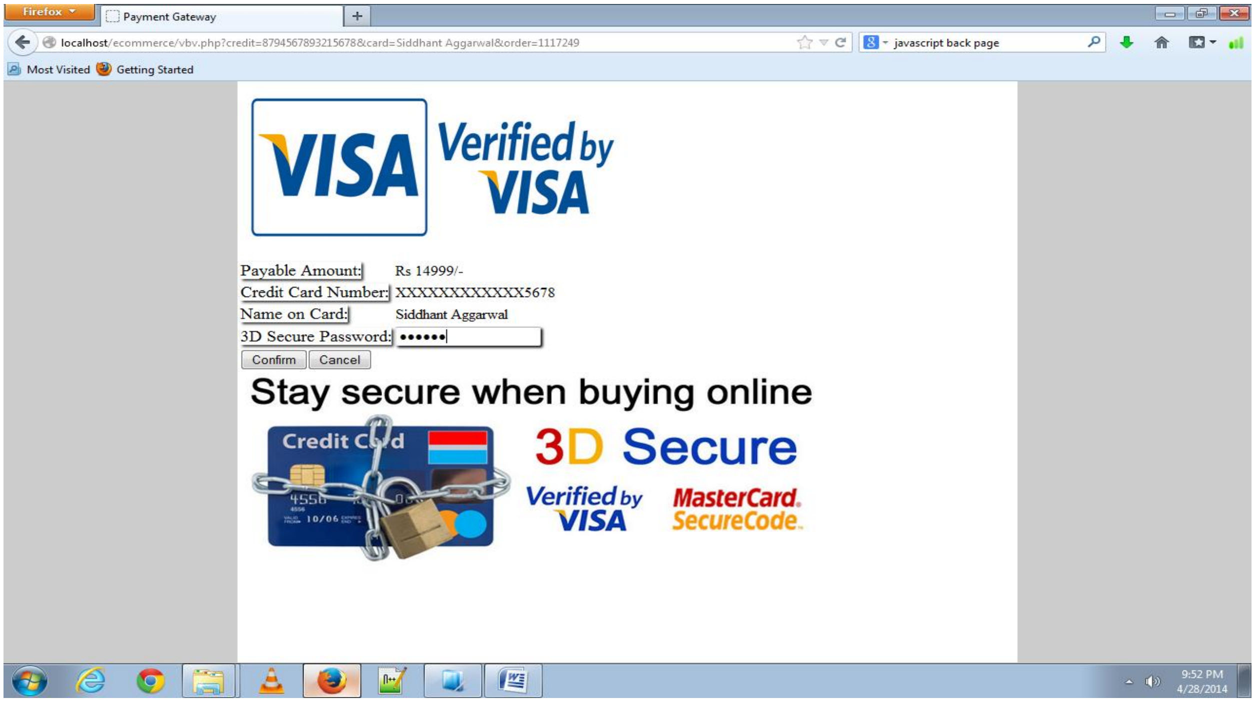
Shipment Confirmed On 2014-04-05 13:32:43

FIGURE 8.19: SHIPMENT PAGE FOR BUYER AND SELLER



Payable Amount: Rs 14999/-
Credit Card Number: 8794567893215678
Name on Card: Siddhant Aggarwal
CVV Number: 111
Expiry Date: 26/12/2014

FIGURE 8.20: PAYMENT GATEWAY FOR ONLINE PAYMENTS



localhost/ecommerce/vbv.php?credit=8794567893215678&card=Siddhant Aggarwal&order=1117249

Payable Amount:	Rs 14999/-
Credit Card Number:	XXXXXXXXXXXX5678
Name on Card:	Siddhant Aggarwal
3D Secure Password:	••••••

FIGURE 8.21: VERIFIED BY VISA PAGE

CHAPTER 9
CONCLUSION

This Project is of large significance for all companies whether it be a small company or a huge one. With the help of this web application one can easily shop from home with no worries. Also its interface is very easy to understand. It has following advantages:

- With this project, the efforts made to perform the required operations are reduced.
- One can easily shop online in future reference.
- Once online, this website can be accessed on any platform.

9.1 FUTURE SCOPE OF THIS PROJECT

eMart Project is an ecommerce website. It ensures user friendly and easy to use website with quality products that can be purchased. eMart website also provides provisions for administrator to easily manage the content of the website. It reduces human efforts and resources (transport) to go to shops and buy products or sell their products. In this very busy lifestyle this system can save the precious time of the customers by obviating the requirement of going to a shop in a heavy traffic. Today’s world is definitely a world of internet, there is a computer in almost every second home and this radio will further improve in few years. So, it’s this online shopping and selling system which is going to rule in the future.

BIBLIOGRAPHY

1. Website Forums - PHP and jQuery.

2. Websites – www.google.com, www.w3schools.com, www.wikipedia.com, www.youtube.com, www.jquery.com.

3. For textbooks - Head First with PHP, Head First with HTML and CSS.