E-COMMERCE WEBSITE (ELECTRONIC SHOPPING)

A Project Report

Submitted By:

MAHESH KUMAR SHARMA

BTE ROLL NO. 1504051027

In partial fulfillment for the award of the Diploma

in

COMPUTER ENGINEERING

UNDER THE SUPERVISION OF:

Mr. SUSHIL KUMAR

(HEAD OF DEPTT., DEPTT. OF COMPUTER ENGINEERING)

at

GURU NANAK DEV INSTITUTE OF TECHNOLOGY

(BOARD OF TECHNICAL EDUCATION)

SECTOR-15, ROHINI, DELHI-110085

DECLARATION

I hereby declare that the project entitled "**E-COMMERCE WEBSITE (ELECTRONIC SHOPPING)**" submitted for the Diploma in Computer Engineering is my original work and the project has not formed the basis for the award of any other diploma, fellowship or any other similar titles.

Place: Delhi **(MAHESH KUMAR SHARMA)**

Date:

CERTIFICATE

This is to certify that the project titled “**E-COMMERCE WEBSITE(ELECTRONIC SHOPPING)**” is the bonafide work carried out by **Mahesh Kumar sharma** (1504051027) students of Diploma in Computer Engineering of Guru Nanak Dev Institute of Technology Sector-15, Rohini, Delhi-110085 during the academic year 2017-18, in partial fulfillment of the requirements for the award of the diploma in Computer Engineering and that the project has not formed the basis for the award previously of any other diploma, fellowship or any other similar title.

Date:

Place: Delhi **(Mr. SUSHIL KUMAR, H.O.D.)**

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our guide Mr**. Sushil Kumar, Head of Department, Department of Computer Engineering**, Guru Nanak Dev Institute of Technology, Sector-15, Rohini, Delhi-110085 for his constant encouragement, valuable guidance and benevolent help, which was of greatest support to bring this work in its present shape. This work is the result of inspiration, support, guidance, motivation, cooperation and facilities that were extended to us at their best at all levels. The discussion with them regarding various issues of our project have been very beneficial and gave us a new direction of thinking. All these discussions have indeed played a vital role in progress of our work at many critical points during my endeavor.

We are highly indebted to Mr. **Sushil Kumar, H.O.D., Department of Computer Engineering**, Guru Nanak Dev Institute of Technology, Sector-15, Rohini, Delhi-110085 for providing us all the necessary facilities and guidance.

We are thankful to our faculties for their valuable lectures in UML design, etc. which helped us in designing this project. We would also like to acknowledge ones, who, from behind the scenes have contributed their ideas and energies.

Table of contents

INTRODUCTION

E-commerce website (Electronic shopping) designed for online purchase an electronic item is presented to a user in an efficient and systematic manner so that they can avail the facilities of electronic items.

As the modern time is automated and computers are working as per the instructions, it becomes essential for the coordination of client's commodity and seller at present time.

This website is designed for the users to save their time and efforts to purchase an electronics items or the availability of the items. Shopping online electronic items having various options at one website. Design to facilitate administrator and user-friendly interface complete and secure information is provided to user scope.

SYSTEM ANALYSIS

EXISTING SYSTEM

Existing system is the manual system in which the customer has to visit the shop seller to purchase any kind of electronic items as per there need. It is very difficult to maintain coordination and right choices of item between the customer and seller.

DISADVANTAGES

1. Customer have to meet a large number of seller available near out.
2. Limited number of choices by the seller.
3. This is a very time consuming process.

PROPOSED SYSTEM

The Online Electronic shopping system is a website designed to fulfill the requirements of electronic items by various seller under one website. This website is designed for the users to save their time and efforts to purchase the electronic items. online purchasing system represents the various options at one website. Design to facilitate administrator and user-friendly interface complete and secure information is provided to user scope.

OBJECTIVE OF THE SYSTEM

Online Electronic shopping system is a website designed to fulfill the need of people to purchase the electronic items. Online shopping project saves their time of offline searching for the availability of electronic items and contact with the sellers or agents. Various electronic item options are available to choose at an affordable price which meets their requirement.

SYSTEM SPECIFICATIONS

MINIMUM HARDWARE REQUIREMENTS

- Intel Pentium (IV) Processor
- 512 MB RAM
- 512 KB Cache Memory
- 2 GB Free Disk Space
- Microsoft Compatible 101/102 Keys Keyboard

RECOMMENDED HARDWARE REQUIREMENTS

- Intel Pentium (R) Dual Core Processor
- 512 MB RAM or Higher
- 512 KB Cache Memory or Higher
- 10 GB Hard Disk Space
- Microsoft Compatible 101/102 Keys Keyboard
- Network connectivity

SOFTWARE REQUIREMENTS

- **Operating System**: Windows, LINUX, Mac OS X, Solaris
- **Web Technology**: JAVA
- **Front-End Technology**: HTML 5, CSS 3, JavaScript, Bootstrap 3
- **Back-End Technology**: JSP, SERVLET
- **Database**: MySQL 5.5
- **IDE:** NetBeans 8.2 IDE

Operating Environment:

The product can run on any browser.

Constraints:

Every user must be comfortable using computer.

All operations are in English so user must have basic knowledge of English.

JAVA: AN INTRODUCTION

Introduction to Java Programming

Java is a powerful object-oriented programming language introduced by Sun Microsystems in 1995, which has built-in support to create programs with a graphical user interface (GUI), utilize the Internet, create client-server solutions, and much more. Programs written in Java can run, without change, on any of the common computer operating systems Windows 95/NT, Macintosh, and Unix. A variant of Java programs called applets can be embedded inside a web page and execute on the computer that is viewing the page, automatically and in a secure environment.

As a language, Java is closely related to C++, which is also object-oriented but retains a lot of idiosyncrasies inherited from its predecessor language C. Java has removed the inconsistent elements from C++, is exclusively object-oriented, and can be considered a modern version of C++. Because of its logical structure Java has quickly become a popular choice as a teaching language, and because of its extensive Internet support and the promise of writing programs once and using them on every operating system Java is becoming more and more accepted in industry.

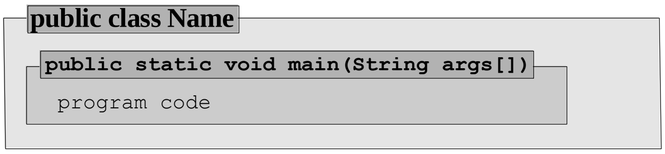
Basic Java Programming Guidelines

Every Java program must follow these guidelines:

- *Java is case sensitive, i.e. the word* `Program` *is different from* `program`.
- *Curly brackets* `{` *and* `}` *are used to group statements together.*
- *An executable Java program must contain at least the following lines as a framework:*

```
public class Name
{  public static void main(String args[])
   {  ... program code ...
   }
}
```

- *Every statement whose next statement is not a separate group must end in a semicolon.*
- *A Java program containing the above framework must be saved using the filename* `Name.java`, *where* `Name` *(including correct upper and lower cases) is the word that follows the keywords* `public class` *and the file extension is* `.java`.



In other words, to create a Java program you first create a text file containing the lines

```
public class Name
{
   public static void main(String args[])
   {
  ... more lines ...
   }
}
```

The file containing our code is called the source code file.

Source Code

> *A Java source code file is a text file that contains programming code written according to the Java language specifications, resembling a mixture of mathematical language and English. A computer cannot execute source code, but humans can read and understand it.*
>
> *Java source code files should be saved as* `Name.java`*, where* `Name` *is the name that appears in the first line of the program:* `public class Name`*. That* `Name` *is referred to as the name of the class, or program. By convention its first letter is capitalized.*

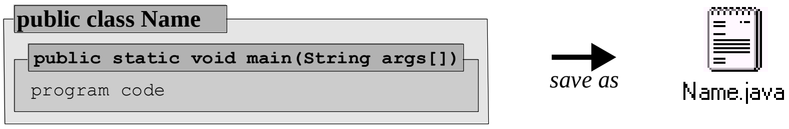


Figure: Saving a Java source code file

Here is an example of a Java source code file. We will later explain what the various lines mean; for now it is simply a text file that looks as shown.

Example: The first source code file

Create a source code file containing the necessary Java code to get the computer to write "`Hi - this is my first program`" on the screen.

Our first Java program looks as follows:

```
public class Test
{
   public static void main(String args[])
   {
```

```
            System.out.println("Hi - this is my first program");
        }
    }
```

This program, or class, is called `Test` and must be saved under the file name `Test.java`.

■

Compiling a Java Program or Class

A source code file, which is more or less readable in plain English, needs to be transformed into another format before the computer can act upon it. That translation process is called *compiling* and is accomplished using the Java compiler `javac` from the Java Developer's Kit (JDK), which could be invoked by an IDE such as BlueJ.

Compiling

> *Compiling is the process of transforming the source code file into a format that the computer can understand and process. The resulting file is called the byte-code, or class, file. The name of the class file is the same as the name of the program plus the extension* `.class`*. The program* `javac` *from the Java Developer's Kit is used to transform a source code file into a class file.*

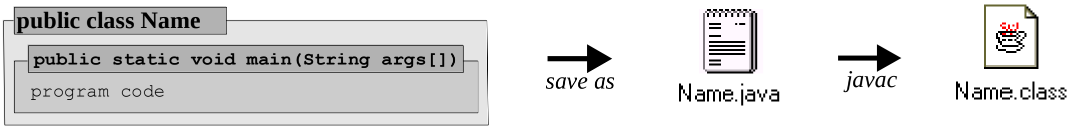


Figure: Compiling and creating class file

If a source code contains any errors, they are flagged by the compiler. You need to fix them and re-compile until there are no further errors.

■

> **Tip:** In case of an error, the `javac` compiler shows the line number and position of where *it thinks* the error occurred in your source code.
>
> - If the compiler points out an error, then there is an error *at or before* the indicated position.
> - If the compiler reports a certain number of errors, than this is *the least* amount of errors.
> - If one error is fixed, other errors may automatically disappear or new ones may appear.
>
> Fix your source code a few errors at a time. Recompile often to see if the number of errors and the error messages change until no errors are reported. If you

cannot find an error at the position indicated by the compiler, look at the code *before* that position.

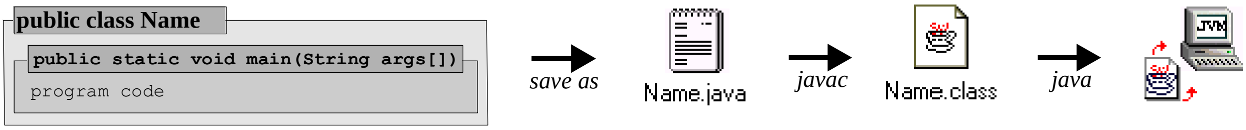
Executing a Java Program or Class

The Java compiler does not produce an executable file, so Java programs can not execute under the operating system of your machine. Instead they execute inside a *Java Virtual Machine*, which is invoked using the `java` program of the JDK.

Executing a Class File

> *To execute a Java program the Java Developer's Kit provides a program called* `java`*. When executing that program with your class file as parameter the following happens:*
>
> - *the Java Virtual Machine (JVM) is created inside your computer*
> - *the JVM locates and reads your class files*
> - *the JVM inspects your class file for any security violations*
> - *the JVM executes, or interprets, your class file according to its instructions if possible*
>
> *Under Windows and Unix, execute a program by typing at the command prompt* `java Name`*, where* `Name` *is the name of the program (no extension). On a Macintosh, double-click the* `java` *icon and select the appropriate class file.*

Most IDE’s allow for a convenient way to execute a file. In BlueJ you right-click on a compiled class and select the “main” method.



```
public class Name
  public static void main(String args[])
  program code
```

Figure: Executing a class file

A good question at this point is which line in a Java program executes *first*.

Default Program Entry Point

> *The default program entry point is that part of a class (or program) where execution begins. For every Java class (or program), the standard program entry point consists of the line:*
>
> ```
> public static void main(String args[])
> ```
>
> *If that line is not present in your source code, the JVM cannot execute your program and displays an error message.*

At this point, we need to explain what the Java Virtual Machine is and how it relates to the operating system and to Java class files.

Java Virtual Machine (JVM)

> *The Java Virtual Machine (JVM) is a platform-independent engine used to run Java applets and applications. The JVM knows nothing of the Java programming language, but it does understand the particular file format of the platform and implementation independent class file produced by a Java compiler. Therefore, class files produced by a Java compiler on one system can execute without change on any system that can invoke a Java Virtual Machine.*[1]
>
> *When invoked with a particular class file, the JVM loads the file, goes through a verification process to ensure system security, and executes the instructions in that class file.*

The JVM, in other words, forms a layer between the operating system and the Java program that is trying to execute. That explains how *one* Java program can run without change on a variety of systems: it cannot! A Java program runs on only *one* system, namely the Java Virtual Machine. That virtual system, in turn, runs on a variety of operating systems and is programmed quite differently for various systems. To the Java programmer, it provides a unified interface to the actual system calls of the operating system.

You can include graphics, graphical user interface elements, multimedia, and networking operations in a Java program and the JVM will negotiate the necessary details between the class file(s) and the underlying operating system. The JVM produces exactly the same results – in theory – regardless of the underlying operating system. In the Basic (or C, or C++) programming language, for example, you can create code that specifies to multiply two integers 1000 and 2000 and store the result as another integer. That code works fine on some systems but can produce negative numbers on others. In Java, this cannot happen: either the code fails on all platforms, or it works on all platforms.

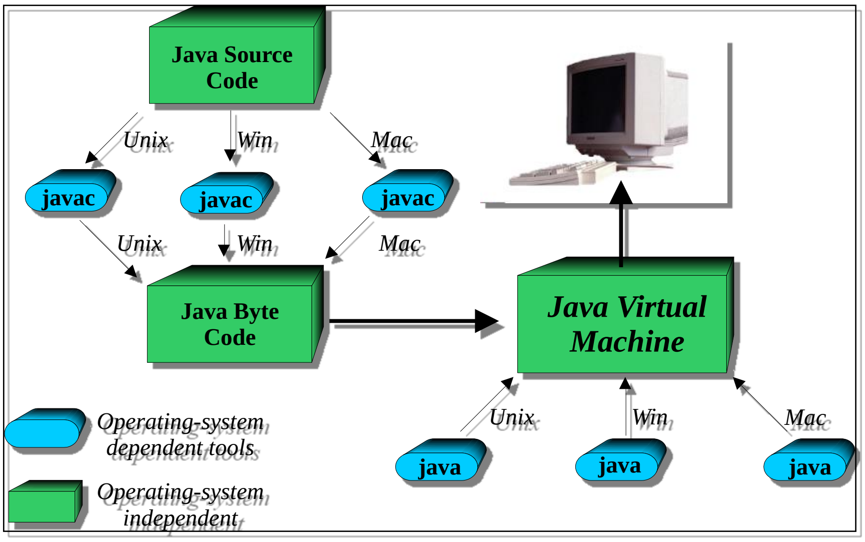


Figure 1.09: Illustrating the machine dependent/independent parts of Java programs

Because the JVM is in effect its own computer, it can shield the actual computer it is running on from potentially harmful effects of a Java program. This is especially important because Java programs known as *applets* can *automatically* start executing on your machine when you are surfing the web if the appropriate feature of your web browser is enabled. If these programs were allowed to meddle with your system, you could accidentally execute a program that would proceed to erase your entire disk. That, of course, would prompt people to disable Java on their web browser, which in turn would be bad news for anyone who supports the Java concept.

SDLC

SOFTWARE DEVELOPMENT LIFE CYCLE

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

SDLC ACTIVITIES

SDLC provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps:

- **Communication**
  This is the first step where the user initiates the request for a desired software product. The user contacts the service provider and tries to negotiate the terms, submits the request to the service providing organization in writing.

- **Requirement Gathering**
  This step onwards the software development team works to carry on the project. The team holds discussions with various stakeholders from problem domain and tries to bring out as much information as possible on their requirements. The requirements are contemplated and segregated into user requirements, system requirements and functional requirements. The requirements are collected using a number of practices as given –

  a) studying the existing or obsolete system and software,
  b) conducting interviews of users and developers,
  c) referring to the database or
  d) collecting answers from the questionnaires.

- **Feasibility Study**
  After requirement gathering, the team comes up with a rough plan of software process. At this step the team analyzes if a software can be designed to fulfill all requirements of the user, and if there is any possibility of software being no more useful. It is also analyzed if the project is financially, practically, and technologically

feasible for the organization to take up. There are many algorithms available, which help the developers to conclude the feasibility of a software project

- **System Analysis**

  At this step the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project. System analysis includes understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc. The project team analyzes the scope of the project and plans the schedule and resources accordingly.

- **Software Design**

  Next step is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design, and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams, and in some cases pseudo codes.

- **Coding**

  This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

- **Testing**

  An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing,

  in-house testing, and testing the product at user’s end. Early discovery of errors and their remedy is the key to reliable software.

- **Integration**

  Software may need to be integrated with the libraries, databases, and other program(s). This stage of SDLC is involved in the integration of software with outer world entities.

- **Implementation**

  This means installing the software on user machines. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

- **Operation and Maintenance**

  This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

FEASIBILITY STUDY

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analyst conducting the study gathers information using a variety of methods, the most popular of which are:

- Interviewing users, employees, managers, and customers.
- Developing and administering questionnaires to interested stakeholders, such as potential users of the information system.
- Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system.
- Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system.
- Modeling, observing, and simulating the work activities of the current system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components. These components are:

Economic feasibility:

The economic viability of the proposed system. The proposed project's cost and benefits are evaluated. Tangible costs include fixed and variable costs, while tangible benefits include cost savings, increased revenue, and increased profit. A project is approved only if it covers its cost in a given period of time. However, a project may be approved only on its intangible benefits such as those relating to government regulations, the image of the organization, or similar considerations.

Technical feasibility:

The possibility that the organization has or can procure the necessary resources. This is demonstrated if the needed hardware and software are available in the marketplace or can be developed by the time of implementation.

Operational feasibility:

The ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, customers, and suppliers. The stakeholders are interested in systems that are easy to operate, make few, if any, errors, produce the desired information, and fall within the objectives of the organization.

SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. Object Oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

Architectural design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis.

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

Physical design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

- Input requirement,
- Output requirements,
- Storage requirements,
- Processing requirements,
- System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

- **User Interface Design**

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them.

- **Data Design**

Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

- **Process Design**

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

PRELIMINARY INVESTIGATION

Preliminary investigation is the first phase. In this phase, the system is investigated. The objective of this phase is to conduct an initial analysis and findings of the system.

System Identification

This phase is used to recognize the need for a new system. The system is identified at this stage. It is very important step. Everything performed in future depends on this definition and identification.

System Scope

The scope of the system is also recognized at this stage. A system can be reduced in its scope due to financial, political or time problems.

Alternate Solutions

This phase is also used to find out alternate solutions for developing the system. The best available solution must be used. The best way to do this is to interview the concerned people in the organization. It may include the clients, customer, supplier and consultants etc. The competitors can also be an important source to find the best way to develop new system.

Feasibility Study

A feasibility study is conducted to find out whether the proposed system is possible, affordable and acceptable for organization. The financial, political, social and time constraints must be considered during this study.

It is important to be reasonably sure of the success of proposed system before initiating work on it. A feasibility study is a study to find out whether the proposed system is:

- **Possible**—to build it with the given technology and resources
- **Affordable**—given the time and cost constraints of the organization
- **Acceptable**—for use by the eventual users of the system.

Purpose of Feasibility Study

A feasibility study is initiated by an organization when a change is required in the current system. A feasibility study is needed to ascertain the advantages and disadvantages of the new system. Following are the important purposes of feasibility study:

- **Need Analysis**—Determine the need for change within an organization
- **Cost Benefit Analysis**—Study the effect of change on the economics of organization
- **Technical Feasibility**—Evaluate various technologies that can be used to implements the suggested change given the cost and resource constraints of an organization
- **Legal Feasibility**—Evaluate the possible legal procedures to implement the change
- **Evaluating Alternatives**—Evaluate various alternatives to resolve the problems of an organization and recommend the best one

A feasibility study is not justified for projects where benefits out-weight costs, technical risks are not high and there are no alternatives. For such projects, conducting a feasibility study will add to unnecessary expenditure of time and money for the study itself.

Time of Feasibility Study

A feasibility Study is conducted to understand the issues that face a project before the project is initiated. Ideally, it should be initiated at the beginning of a project. It will be the least inexpensive and most useful to do so.

If feasibility study is conducted during analysis phase or later, a considerable time and money would have been spent already on project. The findings of feasibility study may be much difficult to implement. The feasibility can be re-evaluated at various stages of a project.

Preliminary Plan

The final step of first phase is to submit all findings in written form for approval. It is also known as feasibility report. It is normally submitted to the top managers of the organization. They take decision about the system by studying this report. They may accept the report, suggest to modify it or may completely reject it.

Preparing the Project Proposal

Project proposal is the document that is prepared from the feasibility report. It is a proposal from the consulting team that outlines the characteristics of a system and gives the initial calendar schedule. The objectives of preparing a project proposal are as follows:

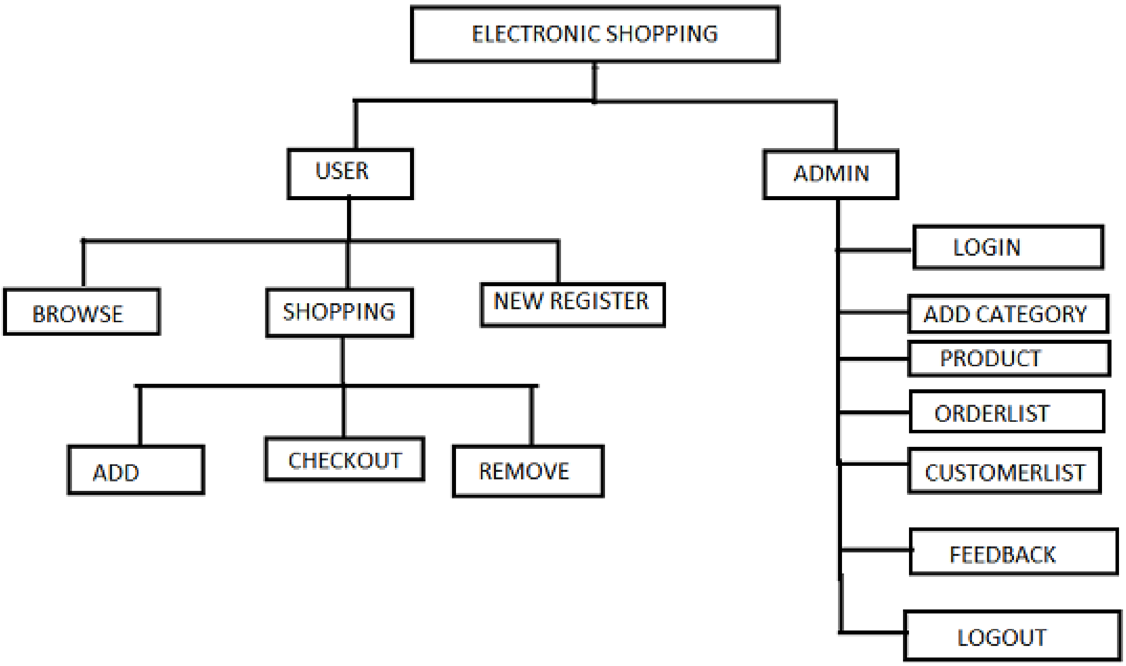
- Indication of initial calendar schedules for the project
- Description of application areas for dealing with identified problems
- Outlining hardware, software, manpower costs
- Identification of training needs
- Calculation of total costs of the project
- Enumeration of the expected benefits

The management decides whether or not to accept the proposal based on the above facts. If they decide to accept it, the next phase of requirement analysis is initiated.

E-R DIAGRAMS

ENTITY RELATIONSHIP DIAGRAMS

Entity Relationship Diagrams are a major data modeling tool and will help organize the data in your project into entities and define the relationships between the entities. This process has proved to enable the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner.



Entity

A data entity is anything real or abstract about which we want to store data. Entity types fall into five classes: roles, events, locations, tangible things or concepts. E.g. employee, payment, campus, book. Specific examples of an entity are called instances. E.g. the employee John Jones, Mary Smith's payment, etc.

Relationship

A data relationship is a natural association that exists between one or more entities. E.g. Employees process payments.

Attribute

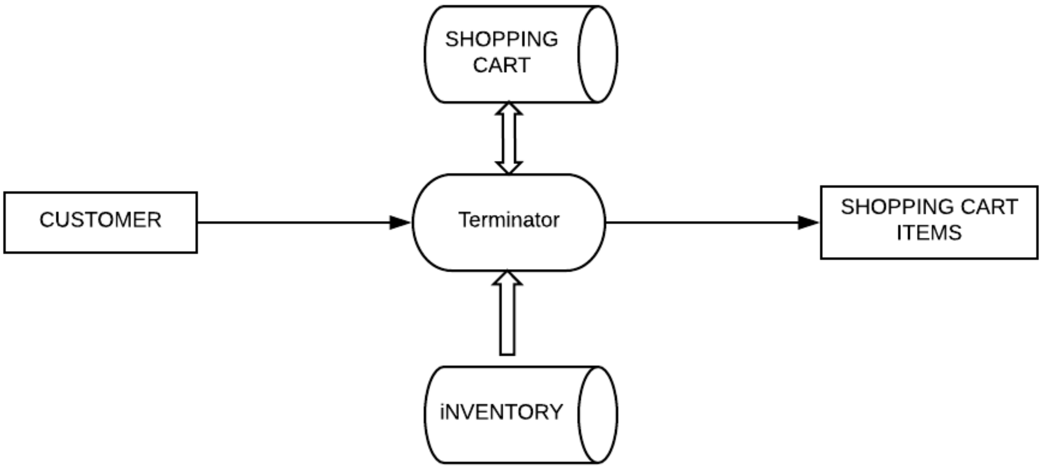
A data attribute is a characteristic common to all or most instances of a particular entity. Synonyms include property, data element, field. E.g. Name, address, Employee Number, pay rate are all attributes of the entity employee. An attribute or combination of attributes that uniquely identifies one and only one instance of an entity is called a primary key or identifier. E.g. Employee Number is a primary key for Employee.

DFD

(DATA FLOW DIAGRAMS)

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

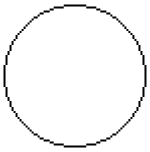
CUSTOMER SHOPPING DFD



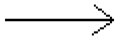
Representation of Components

DFDs only involve four symbols. They are:

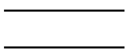
- Process
- Data Object
- Data Store
- External entity



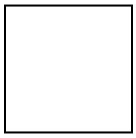
Process
Transform of incoming data flow(s) to outgoing flow(s).



Data Flow
Movement of data in the system.



Data Store
Data repositories for data that are not moving. It may be as simple as a buffer or a queue or a s sophisticated as a relational database.



External Entity
Sources of destinations outside the specified system boundary.

Relationship and Rules

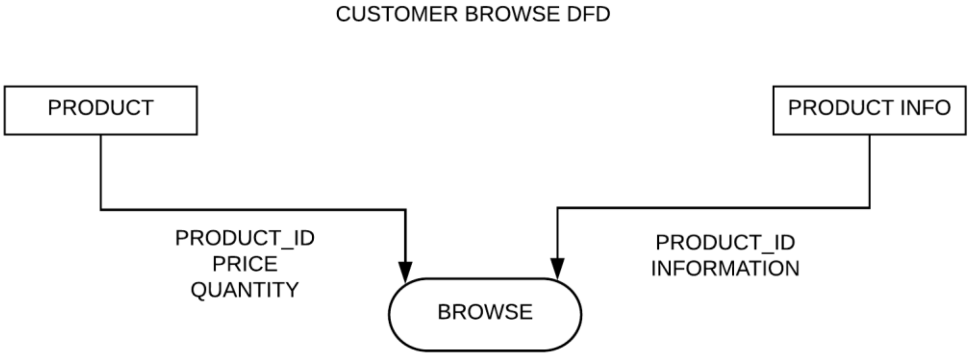
Relationship

The DFD may be used for any level of data abstraction. DFD can be partitioned into levels. Each level has more information flow and data functional details than the previous level.

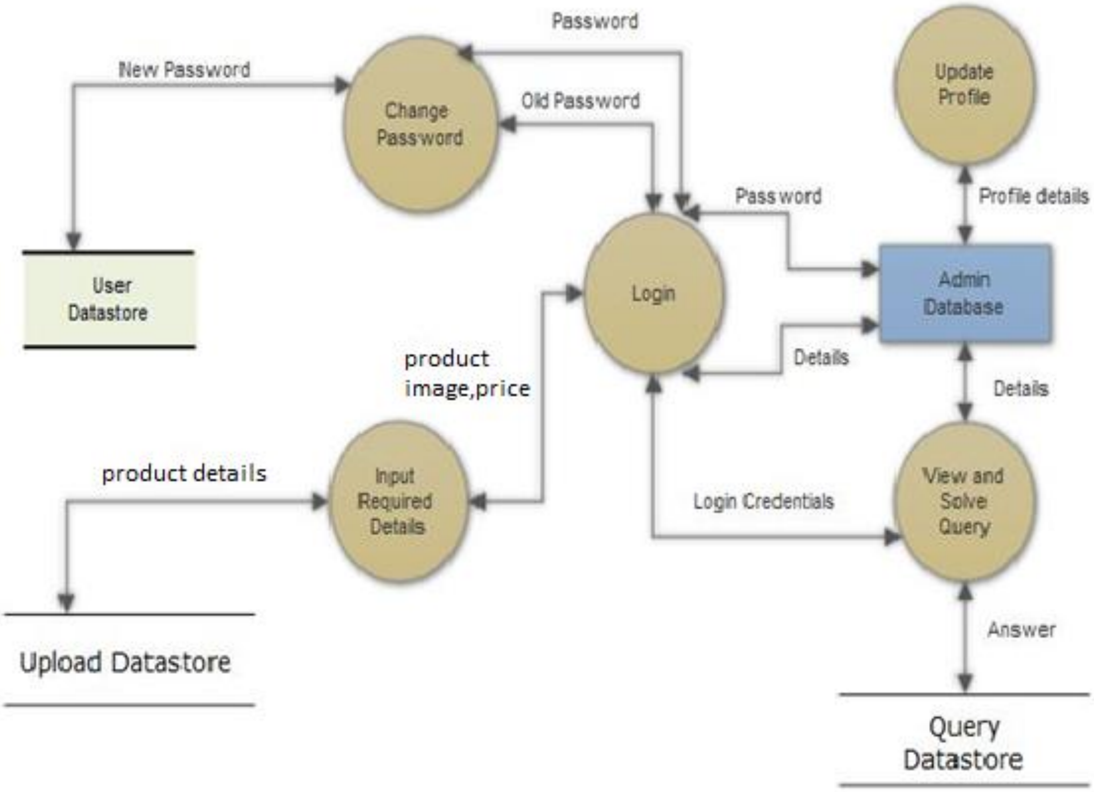
Highest level is Context Diagram. Some important points are:

- 1 bubble (process) represents the entire system.
- Data arrows show input and output.
- Data Stores NOT shown. They are within the system.

FIRST LEVEL DFD DESIGN



(ADMIN'S VIEW)



Next level is Level 1 DFD (Admin View). Some important points are:

- Level 1 DFD must balance with the Level 0 it describes.
- Input going into a process are different from outputs leaving the process.
- Continue to show data stores.

Strengths and Weaknesses of DFD Design

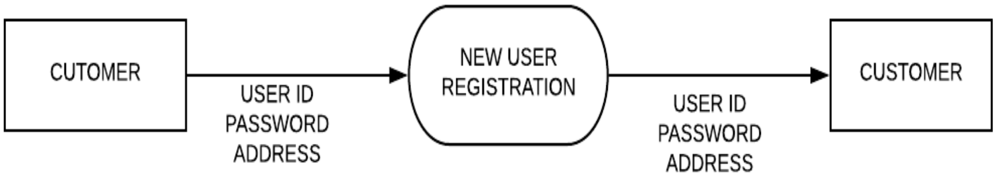
Strengths

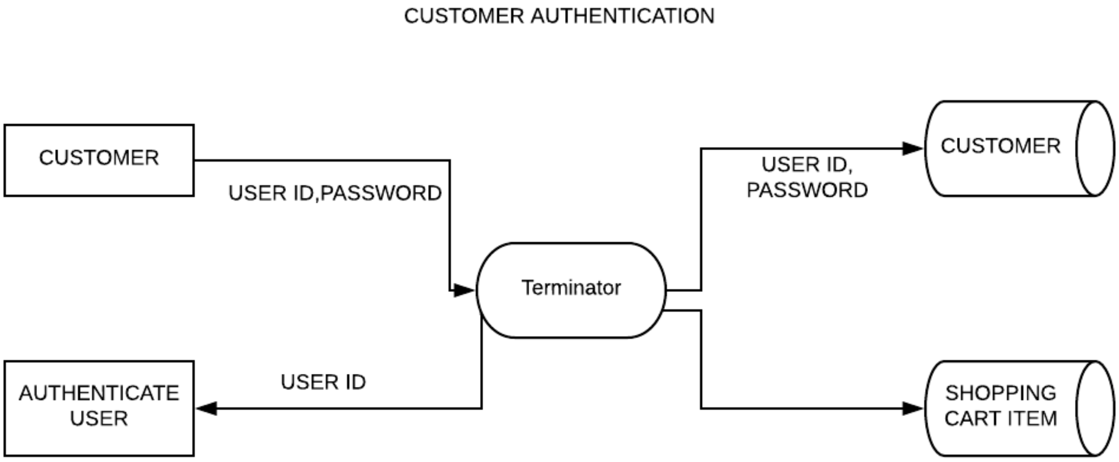
- DFDs have diagrams that are easy to understand, check and change data.
- DFDs help tremendously in depicting information about how an organization operations.
- They give a very clear and simple look at the organization of the interfaces between an application and the people or other applications that use it.

Weaknesses

- Modification to a data layout in DFDs may cause the entire layout to be changed. This is because the specific changed data will bring different data to units that it accesses. Therefore, evaluation of the possible of the effect of the modification must be considered first.
- The number of units in a DFD in a large application is high. Therefore, maintenance is harder, more costly and error prone. This is because the ability to access the data is passed explicitly from one component to the other. This is why changes are impractical to be made on DFDs especially in large system.

CUSTOMER NEW REGISTRATION





UML USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

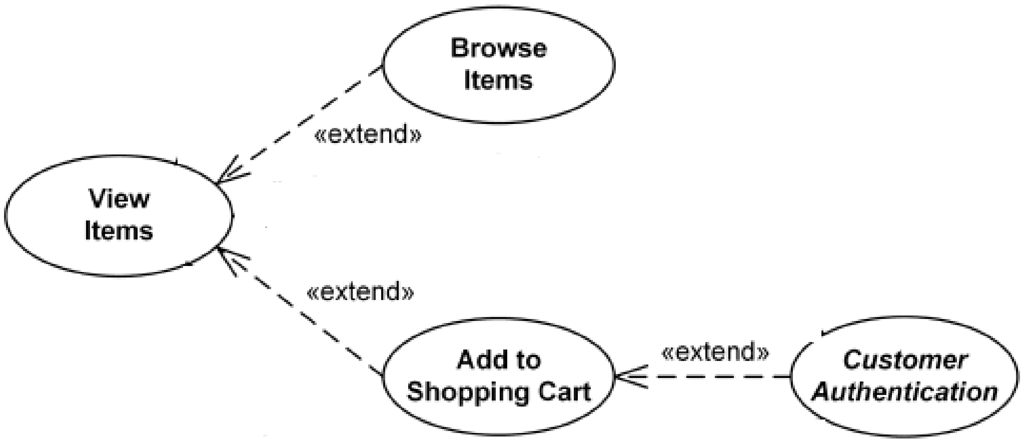
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

How to Draw a Use Case Diagram?



Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

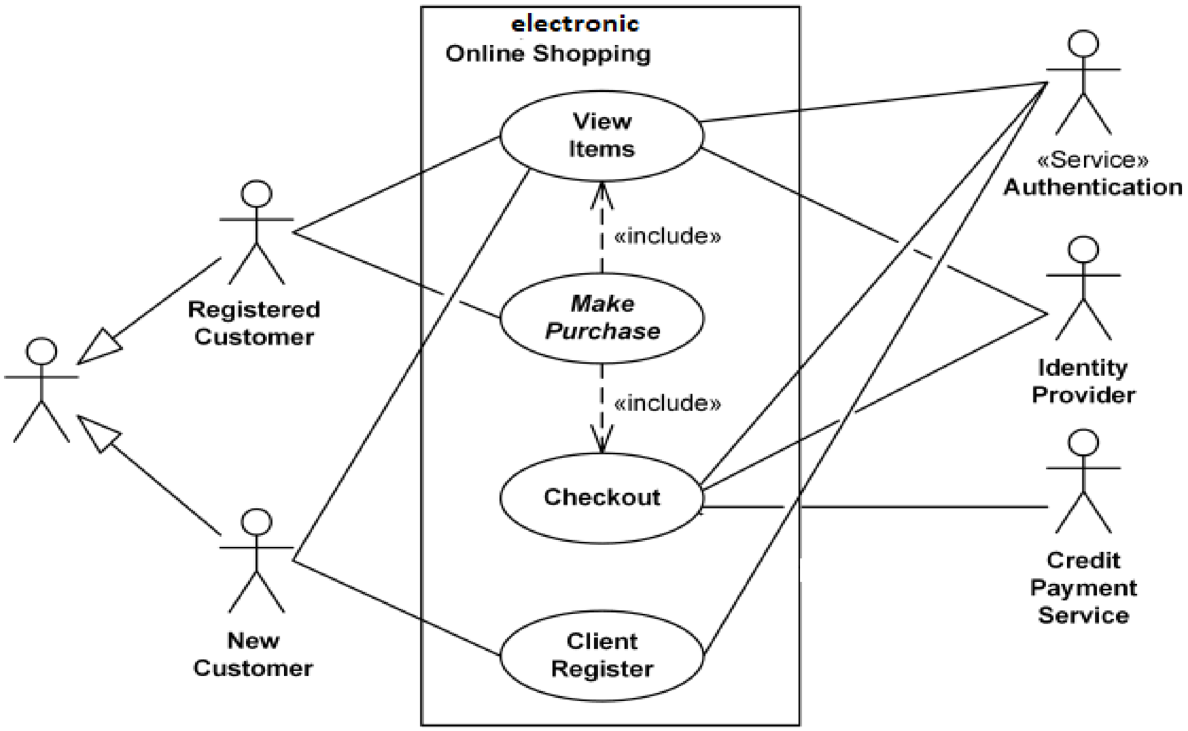
We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.



USE OF UML USE CASE DIAGRAM

To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for −

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

UML SEQUENCE DIAGRAM

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends. Along with our UML diagramming tool, use this guide to learn everything there is to know about sequence diagrams in UML.

BENEFITS OF UML SEQUENCE DIAGRAM

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

Represent the details of a UML use case.

Model the logic of a sophisticated procedure, function, or operation.

See how objects and components interact with each other to complete a process.

Plan and understand the detailed functionality of an existing or future scenario.

USE CASES FOR SEQUENCE DIAGRAMS

The following scenarios are ideal for using a sequence diagram:

Usage scenario: A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.

Method logic: Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to explore the logic of any function, procedure, or complex process.

Service logic: If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

TECHNOLOGY OVERVIEW

The technology selected for implementing Online Electronic System is java/MYSQL 5. Apache is used as the HTTP server. The development was done in a ‘Windows’ environment using NetBeans IDE 8.2.

JAVA

JAVA is a general-purpose programming language that is especially suited to server-side web development where java generally runs on a jdk environment, code is embedded into the HTML source document. Any java code in a requested file is executed by the jre runtime, usually to create dynamic web page content. It can also be used for command-line jdk and client-side GUI applications. PHP can be deployed on many web servers and operating systems, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the oracle company provides the complete source code for users to build, customize and extend for their own use. Latest stable version is java jdk10.

MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is a popular choice of database for use in web applications and is an open source product. The process of setting up a MySQL database varies from host to host, however we will end up with a database name, a user name and a password. Before using our database, we must create a table. A table is a section of the database for storing related information. In a table we will set up the different fields which will be used in that table. Creating a table in MySQL is simple, we just type the SHOW DATABASES; enter. Use <database name>; we will then be taken to a setup screen where you must create the fields for the database. We have used this method in order to create our database and tables. Latest stable version is MySQL 5.

Apache Tomcat

The Apache HTTP Server is a web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Since April 1996 Apache has been the most popular HTTP server software in use. As of November 2010 Apache, served over 59.36% of all websites and over 66.56% of the first one million busiest websites.

Glassfish server

Glassfish is an open-source application server project started by Sun Microsystems for the Java EE platform and now sponsored by Oracle Corporation. The supported version is called Oracle Glassfish Server. Glassfish is free software, dual-licensed under two free software licenses: the Common Development and Distribution License (CDDL) and the GNU General Public License (GPL) with the class path exception. Glassfish is the reference implementation of Java EE and as such supports Enterprise JavaBeans, JPA, Java Server Faces, JMS, RMI, Java Server Pages, servlets, etc. This allows developers to create enterprise applications that are portable and scalable, and that integrate with legacy technologies. Optional components can also be installed for additional services.

Built on a modular kernel powered by OSGi, Glassfish runs straight on top of the Apache Felix implementation. It also runs with Equinox OSGi or Knifefish OSGi runtimes. HK2 abstracts the OSGi module system to provide components, which can also be viewed as services. Such services can be discovered and injected at runtime.

Glassfish is based on source code released by Sun and Oracle Corporation's TopLink persistence system. It uses a derivative of Apache Tomcat as the servlet container for serving Web content, with an added component called Grizzly which uses Java New I/O (NIO) for scalability and speed..

Creating a Database and Inserting Data

Now that we have run and tested Apache and JAVA, the next step is running MySQL and creating a database and table which will hold information to be used by our website. In order to start MySQL, navigate to the MySQL command line client and run the MySQL. The MySQL command line client package contains an application called MySQL which allows developers to administer and maintain MySQL databases. We will be using MySQL to create a database and table, and enter test data., we type http://localhost/admin/ into our web browser. If successful we will be presented with jsp (java server page) start page similar to the one shown below:

The first step with MySQL running is creating a new database. We create a new database name electronic. The successful execution of the SQL query creates a database. We also inserted values in the admin table. The screenshot below shows the successful execution of the query thus creation of a database named electronic.

Thus, we have learned to create a database in MYSQL. After creating the database and tables we are now ready to use them in our website “Online electronic shopping system”.

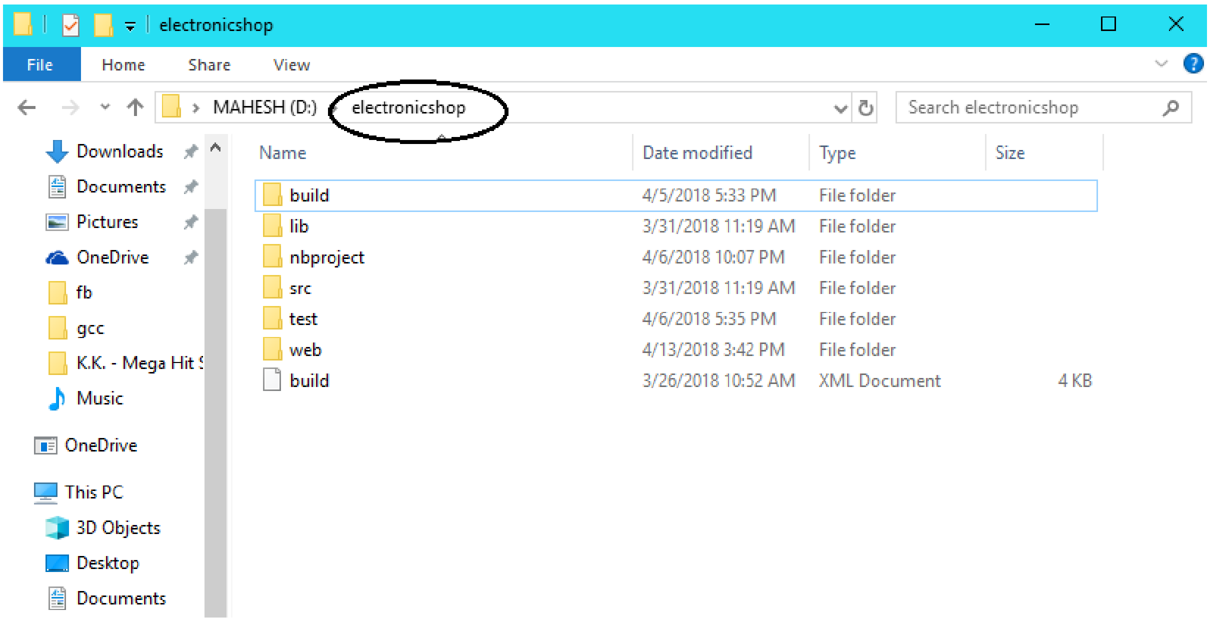
PROJECT DESCRIPTION

INTRODUCTION

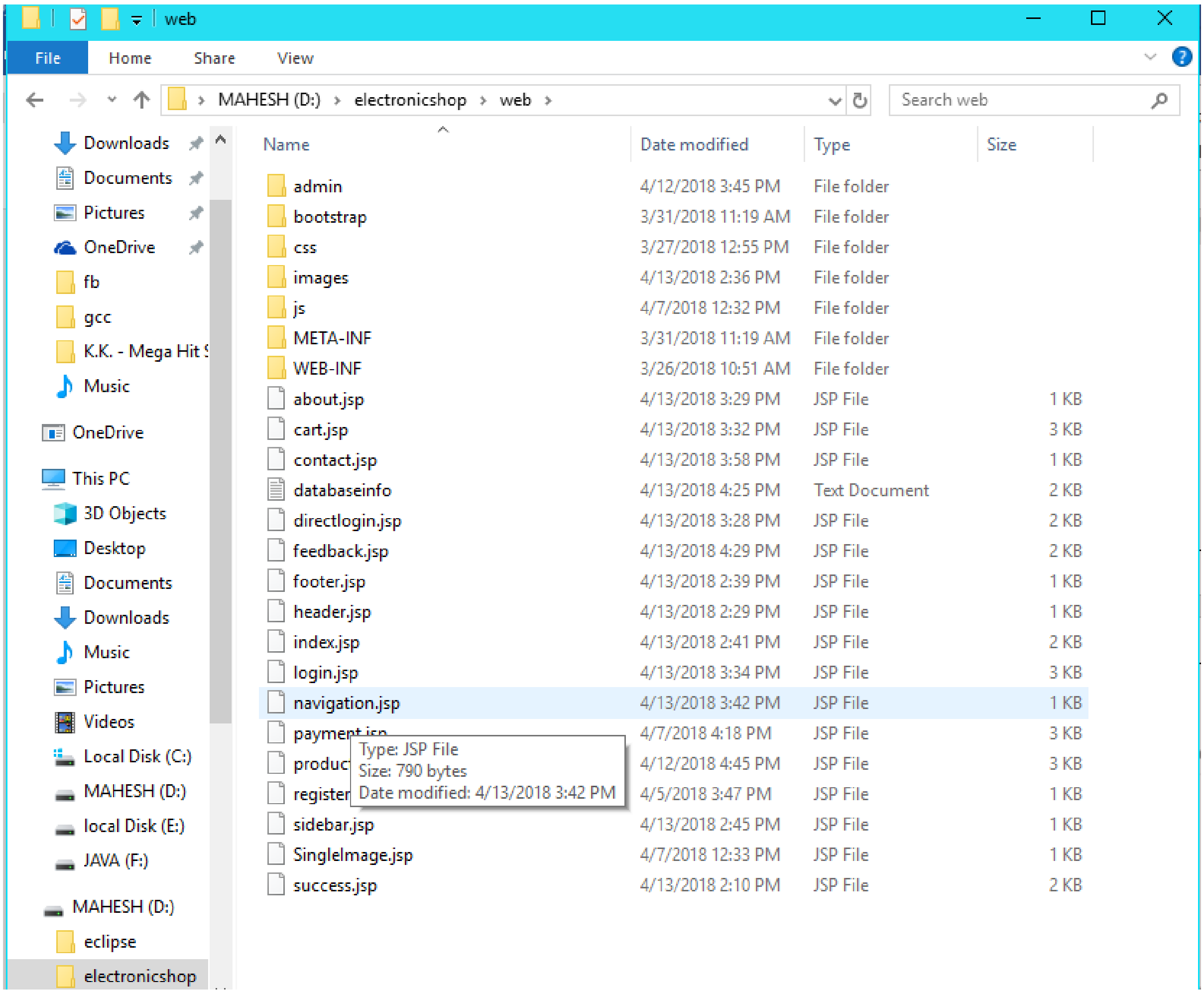
Online electronic shopping website is designed for online shopping of electronic items etc. are presented to a user in an efficient and systematic manner so that they can avail the facilities of online shopping.

This website is designed for the users to save their time and efforts to purchase directly form the website and searching for the availability of the electronic items. Online shopping of electronic items presents the various options at one website. Design to facilitate administrator and user friendly interface complete and secure information is provided to user scope.

DIRECTORY STRUCTURE OF PROJECT

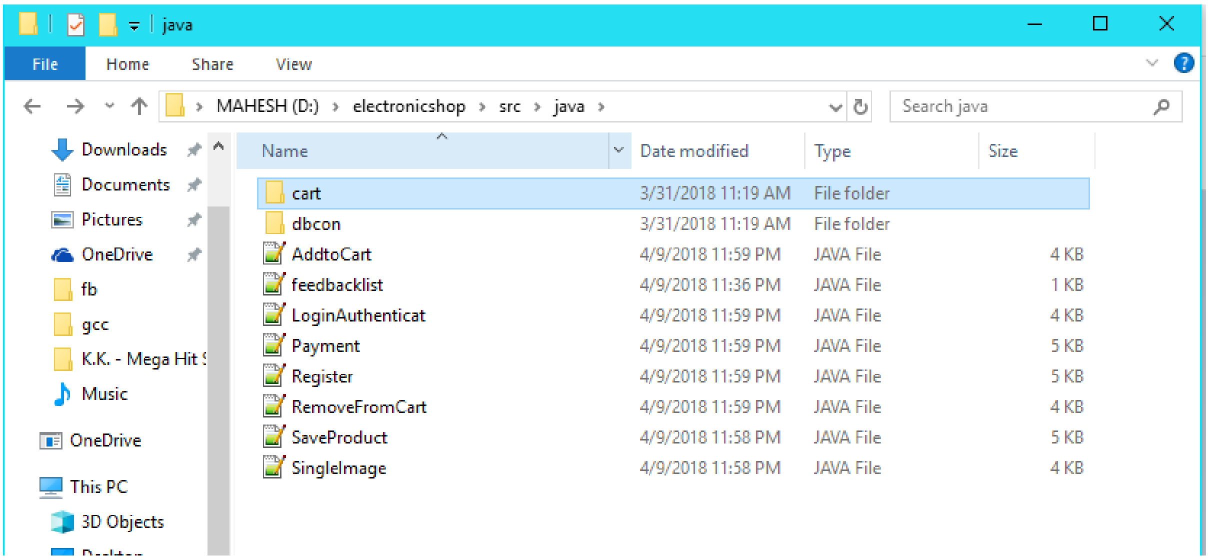


Name	Date modified	Type	Size
build	4/5/2018 5:33 PM	File folder	
lib	3/31/2018 11:19 AM	File folder	
nbproject	4/6/2018 10:07 PM	File folder	
src	3/31/2018 11:19 AM	File folder	
test	4/6/2018 5:35 PM	File folder	
web	4/13/2018 3:42 PM	File folder	
build	3/26/2018 10:52 AM	XML Document	4 KB

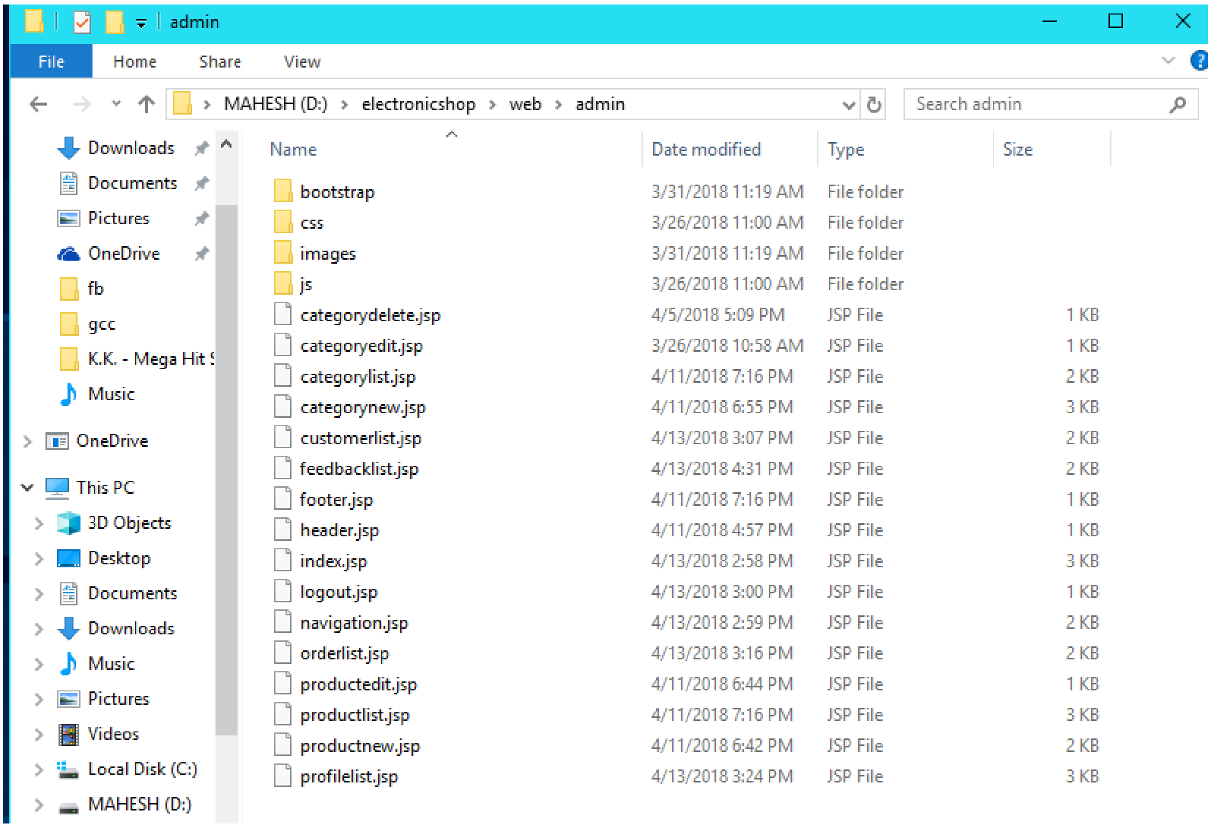


Name	Date modified	Type	Size
admin	4/12/2018 3:45 PM	File folder	
bootstrap	3/31/2018 11:19 AM	File folder	
css	3/27/2018 12:55 PM	File folder	
images	4/13/2018 2:36 PM	File folder	
js	4/7/2018 12:32 PM	File folder	
META-INF	3/31/2018 11:19 AM	File folder	
WEB-INF	3/26/2018 10:51 AM	File folder	
about.jsp	4/13/2018 3:29 PM	JSP File	1 KB
cart.jsp	4/13/2018 3:32 PM	JSP File	3 KB
contact.jsp	4/13/2018 3:58 PM	JSP File	1 KB
databaseinfo	4/13/2018 4:25 PM	Text Document	2 KB
directlogin.jsp	4/13/2018 3:28 PM	JSP File	2 KB
feedback.jsp	4/13/2018 4:29 PM	JSP File	2 KB
footer.jsp	4/13/2018 2:39 PM	JSP File	1 KB
header.jsp	4/13/2018 2:29 PM	JSP File	1 KB
index.jsp	4/13/2018 2:41 PM	JSP File	2 KB
login.jsp	4/13/2018 3:34 PM	JSP File	3 KB
navigation.jsp	4/13/2018 3:42 PM	JSP File	1 KB
payment.jsp	4/7/2018 4:18 PM	JSP File	3 KB
produc	4/12/2018 4:45 PM	JSP File	3 KB
register	4/5/2018 3:47 PM	JSP File	1 KB
sidebar.jsp	4/13/2018 2:45 PM	JSP File	1 KB
SingleImage.jsp	4/7/2018 12:33 PM	JSP File	1 KB
success.jsp	4/13/2018 2:10 PM	JSP File	2 KB

Type: JSP File
Size: 790 bytes
Date modified: 4/13/2018 3:42 PM



Name	Date modified	Type	Size
cart	3/31/2018 11:19 AM	File folder	
dbcon	3/31/2018 11:19 AM	File folder	
AddtoCart	4/9/2018 11:59 PM	JAVA File	4 KB
feedbacklist	4/9/2018 11:36 PM	JAVA File	1 KB
LoginAuthenticat	4/9/2018 11:59 PM	JAVA File	4 KB
Payment	4/9/2018 11:59 PM	JAVA File	5 KB
Register	4/9/2018 11:59 PM	JAVA File	5 KB
RemoveFromCart	4/9/2018 11:59 PM	JAVA File	4 KB
SaveProduct	4/9/2018 11:58 PM	JAVA File	5 KB
SingleImage	4/9/2018 11:58 PM	JAVA File	4 KB



Name	Date modified	Type	Size
bootstrap	3/31/2018 11:19 AM	File folder	
css	3/26/2018 11:00 AM	File folder	
images	3/31/2018 11:19 AM	File folder	
js	3/26/2018 11:00 AM	File folder	
categorydelete.jsp	4/5/2018 5:09 PM	JSP File	1 KB
categoryedit.jsp	3/26/2018 10:58 AM	JSP File	1 KB
categorylist.jsp	4/11/2018 7:16 PM	JSP File	2 KB
categorynew.jsp	4/11/2018 6:55 PM	JSP File	3 KB
customerlist.jsp	4/13/2018 3:07 PM	JSP File	2 KB
feedbacklist.jsp	4/13/2018 4:31 PM	JSP File	2 KB
footer.jsp	4/11/2018 7:16 PM	JSP File	1 KB
header.jsp	4/11/2018 4:57 PM	JSP File	1 KB
index.jsp	4/13/2018 2:58 PM	JSP File	3 KB
logout.jsp	4/13/2018 3:00 PM	JSP File	1 KB
navigation.jsp	4/13/2018 2:59 PM	JSP File	2 KB
orderlist.jsp	4/13/2018 3:16 PM	JSP File	2 KB
productedit.jsp	4/11/2018 6:44 PM	JSP File	1 KB
productlist.jsp	4/11/2018 7:16 PM	JSP File	3 KB
productnew.jsp	4/11/2018 6:42 PM	JSP File	2 KB
profilelist.jsp	4/13/2018 3:24 PM	JSP File	3 KB

DESCRIPTION OF ROOT DIRECTORY CONTENTS

admin Directory: This directory contains the administrator directories and files like:

categorydelete.jsp

categoryedit.jsp

categorylist.jsp

categorynew.jsp

Customerlist.jsp

feedbacklist.jsp

footer.jsp

header.jsp

index (main page). jsp

Logout. Jsp

Navigation. Jsp

Orderlist. Jsp

Productedit. jsp

Productlist. Jsp

Productnew.jsp

Profilelist.jsp etc.

Customer directory:

singleImage.jsp

about.jsp

cart.jsp

contact.jsp

directlogin.jsp

feedback.jsp

footer.jsp

header.jsp

index.jsp

login.jsp

navigation.jsp

payment.jsp

productdetail.jsp

register.jsp

sidebar.jsp

success.jsp

Java servlet directory:

Addtocart.java

LoginAuthenticate.java

Payment.java

Register.java

Removefromcart.java

saveproduct.java

singleimage.java

feedbacklist.java

assets Directory: This directory contains the bootstrap files and folder to make the website responsive.

includes Directory: This directory contains java files for connection to database, header, footer, color switcher, sidebar, etc.

index.jsp: It is the first page which is opened after opening the project.

Logout.jsp: This file is used for the admin to logout the session.

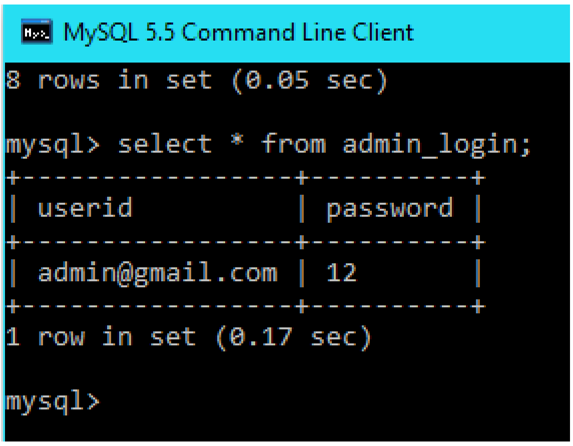
cart: This file contains the cart information of the user for his items saved in his carts.

Feedback form: it contains feedback form for user review which is directly seen by the admin.

profile.jsp: This file contains the information of a admin, where admin can change their own password which is stored in database.

DESCRIPTION OF DATABASE TABLES

admin



```
8 rows in set (0.05 sec)

mysql> select * from admin_login;
+-----------------+----------+
| userid          | password |
+-----------------+----------+
| admin@gmail.com | 12       |
+-----------------+----------+
1 row in set (0.17 sec)

mysql>
```

userid: Stores the id of the admin.

Password: Stores the encrypted password of the administrator.

Category: admin can add new category.

Add new product.

Update product.

See the customer list.

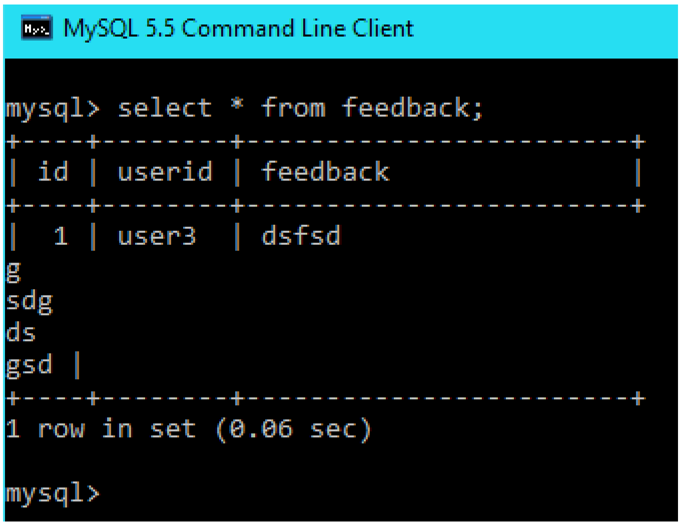
Check the orderlist

See the feedback given by the user

Admin can change the password though profile.

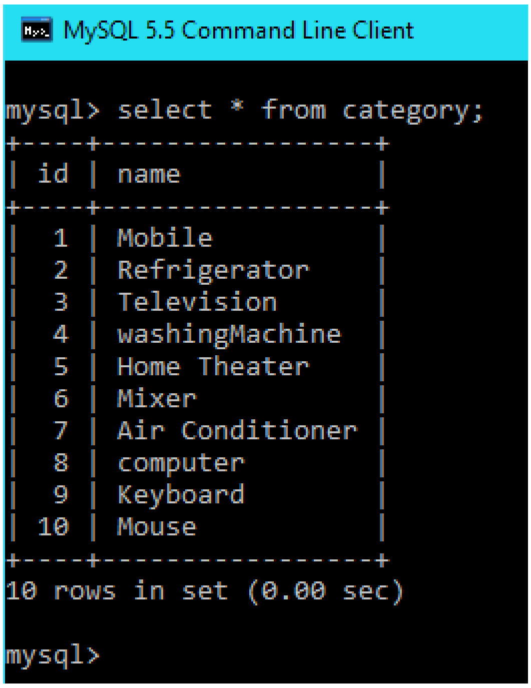
Logout.

Customer feedback list:



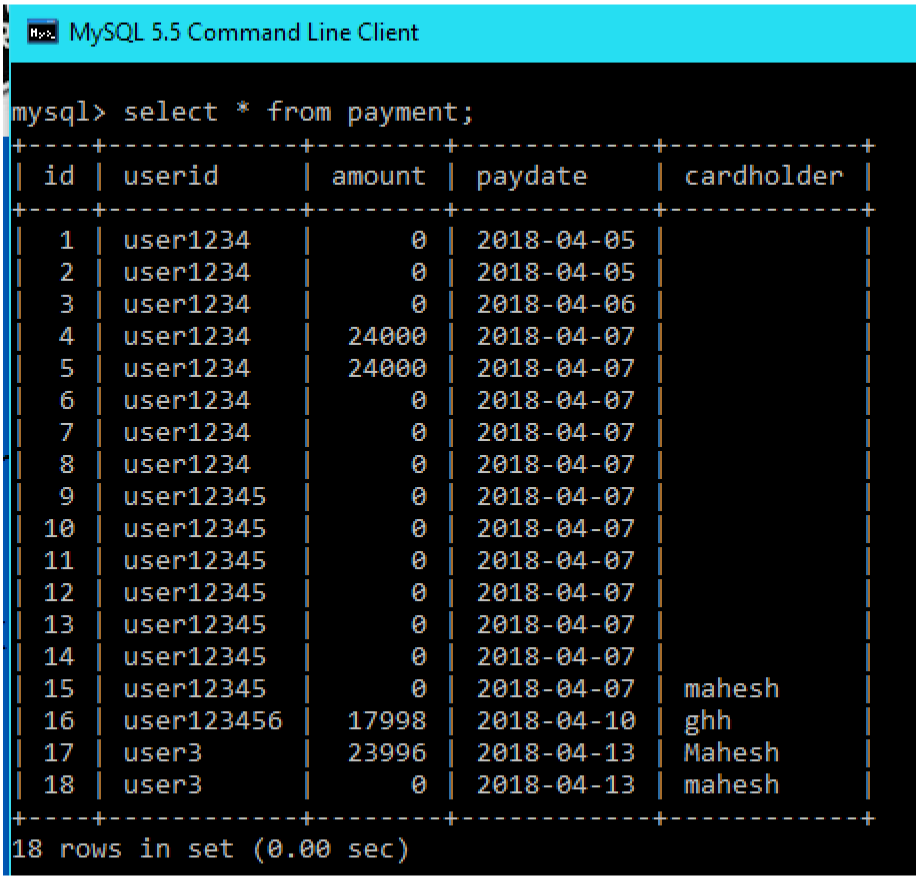
```
mysql> select * from feedback;
+----+--------+------------------------+
| id | userid | feedback               |
+----+--------+------------------------+
|  1 | user3  | dsfsd
g
sdg
ds
gsd |
+----+--------+------------------------+
1 row in set (0.06 sec)

mysql>
```

Table category:



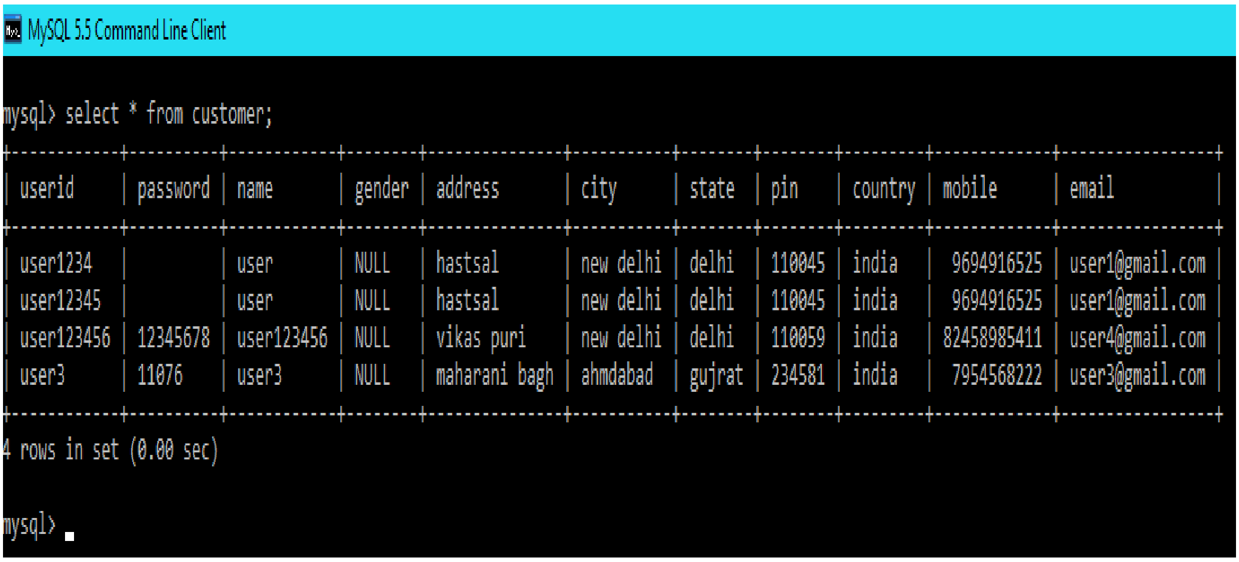
```
mysql> select * from category;
+----+-----------------+
| id | name            |
+----+-----------------+
|  1 | Mobile          |
|  2 | Refrigerator    |
|  3 | Television      |
|  4 | washingMachine  |
|  5 | Home Theater    |
|  6 | Mixer           |
|  7 | Air Conditioner |
|  8 | computer        |
|  9 | Keyboard        |
| 10 | Mouse           |
+----+-----------------+
10 rows in set (0.00 sec)

mysql>
```

Payment table:



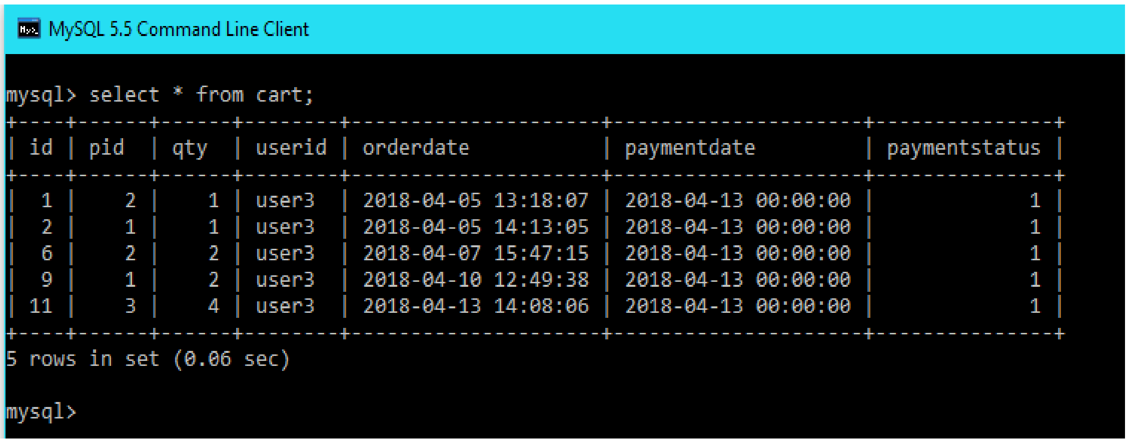
```
mysql> select * from payment;
+----+------------+--------+------------+------------+
| id | userid     | amount | paydate    | cardholder |
+----+------------+--------+------------+------------+
|  1 | user1234   |      0 | 2018-04-05 |            |
|  2 | user1234   |      0 | 2018-04-05 |            |
|  3 | user1234   |      0 | 2018-04-06 |            |
|  4 | user1234   |  24000 | 2018-04-07 |            |
|  5 | user1234   |  24000 | 2018-04-07 |            |
|  6 | user1234   |      0 | 2018-04-07 |            |
|  7 | user1234   |      0 | 2018-04-07 |            |
|  8 | user1234   |      0 | 2018-04-07 |            |
|  9 | user12345  |      0 | 2018-04-07 |            |
| 10 | user12345  |      0 | 2018-04-07 |            |
| 11 | user12345  |      0 | 2018-04-07 |            |
| 12 | user12345  |      0 | 2018-04-07 |            |
| 13 | user12345  |      0 | 2018-04-07 |            |
| 14 | user12345  |      0 | 2018-04-07 |            |
| 15 | user12345  |      0 | 2018-04-07 | mahesh     |
| 16 | user123456 |  17998 | 2018-04-10 | ghh        |
| 17 | user3      |  23996 | 2018-04-13 | Mahesh     |
| 18 | user3      |      0 | 2018-04-13 | mahesh     |
+----+------------+--------+------------+------------+
18 rows in set (0.00 sec)
```

Customer list:



```
mysql> select * from customer;
+------------+----------+------------+--------+---------------+-----------+--------+--------+---------+-------------+------------------+
| userid     | password | name       | gender | address       | city      | state  | pin    | country | mobile      | email            |
+------------+----------+------------+--------+---------------+-----------+--------+--------+---------+-------------+------------------+
| user1234   |          | user       | NULL   | hastsal       | new delhi | delhi  | 110045 | india   |  9694916525 | user1@gmail.com  |
| user12345  |          | user       | NULL   | hastsal       | new delhi | delhi  | 110045 | india   |  9694916525 | user1@gmail.com  |
| user123456 | 12345678 | user123456 | NULL   | vikas puri    | new delhi | delhi  | 110059 | india   | 82458985411 | user4@gmail.com  |
| user3      | 11076    | user3      | NULL   | maharani bagh | ahmdabad  | gujrat | 234581 | india   |  7954568222 | user3@gmail.com  |
+------------+----------+------------+--------+---------------+-----------+--------+--------+---------+-------------+------------------+
4 rows in set (0.00 sec)

mysql>
```

Cart table:



```
mysql> select * from cart;
+----+------+------+--------+---------------------+---------------------+---------------+
| id | pid  | qty  | userid | orderdate           | paymentdate         | paymentstatus |
+----+------+------+--------+---------------------+---------------------+---------------+
|  1 |    2 |    1 | user3  | 2018-04-05 13:18:07 | 2018-04-13 00:00:00 |             1 |
|  2 |    1 |    1 | user3  | 2018-04-05 14:13:05 | 2018-04-13 00:00:00 |             1 |
|  6 |    2 |    2 | user3  | 2018-04-07 15:47:15 | 2018-04-13 00:00:00 |             1 |
|  9 |    1 |    2 | user3  | 2018-04-10 12:49:38 | 2018-04-13 00:00:00 |             1 |
| 11 |    3 |    4 | user3  | 2018-04-13 14:08:06 | 2018-04-13 00:00:00 |             1 |
+----+------+------+--------+---------------------+---------------------+---------------+
5 rows in set (0.06 sec)

mysql>
```

Codding part:

Admin: -

Categorydelete.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>your product deleted successfully! </h1>
  </body>
</html>
```

Categoryedit.jsp:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Categorylist.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>



<div class="row"style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #0062cc">
    <%@include file="navigation.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: antiquewhite">
    <div class="row">
      <div class="col-sm-3">
        <a href="categorynew.jsp" style="text-decoration: none">
        <button class="btn-primary">Add New</button>
        </a>
      </div>
      <div class="col-sm-3">
        <a href="categorylist.jsp" style="text-decoration: none">
        <button class="btn-primary">List Category</button>
        </a>
      </div>
      <div class="col-sm-3"></div>
      <div class="col-sm-3"></div>
    </div>
    <div class="row" style="text-align: center;">
      <div class="col-sm-12">
        <h3>List Category</h3>
        <table class="table-bordered" style="width: 80%">
          <tr><th>ID</th><th>Name</th></tr>
          <%
          try
          {
            dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
            Class.forName(ob.DRIVER);
            Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
            Statement st=cn.createStatement();
            String query="select * from category order by name";
            ResultSet rs=st.executeQuery(query);
            while(rs.next())
            {
              out.println("<tr>");
              out.println("<td>"+rs.getString(1)+"</td>");
              out.println("<td>"+rs.getString(2)+"</td>");
              out.println("</tr>");
            }
          }
          catch(Exception ex)
          {
            out.println("Error "+ex);
          }
          %>
        </table>
         </div>
```

```
  </div>
</div>
</div>
%@include file="footer.jsp" %
```

Categorynew.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="container-fluid">
<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #0062cc">
    <%@include file="navigation.jsp" %>
  </div>
  <div class="col-sm-10">

    <div class="row">
      <div class="col-sm-3">
        <a href="categorynew.jsp" style="text-decoration: none">
          <button class="btn-primary">Add New</button>
        </a>
      </div>
      <div class="col-sm-3">
        <a href="categorylist.jsp" style="text-decoration: none">
          <button class="btn-primary">List Category</button>
        </a>
      </div>
      <div class="col-sm-3"></div>
      <div class="col-sm-3"></div>
    </div>
    <div class="row" style="text-align: center;">
      <div class="col-sm-12">
        <h3>New Category</h3>
      </div>
```

```
   </div>
   <%!dbcon.DatabaseConnection ob;%>
   <%
     String msg="";
    try {
     if (request.getParameter("submit") != null) {
       ob=new dbcon.DatabaseConnection();
       Class.forName(ob.DRIVER);
       Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
       Statement st = cn.createStatement();
       String n = request.getParameter("txtname");
       String query = "insert into category (name) values('" + n + "')";
       int re = st.executeUpdate(query);
       if (re > 0) {
         msg = "<span style='color:green'>Save successfully!</span>";
       }
     }
    } catch (Exception ex) {
     msg = "<span style='color:red'>Error " + ex + "</span>";
    }
   %>
   <form method="post">
    <div class="form-control">
     <label>Category Name</label>
     <input type="text" name="txtname" class="form-control"
          style="margin-bottom: 10px"/>
     <input type="submit" name="submit" value="Save" class="btn-success"/>
     <%=msg%>
    </div>
   </form>
  </div>
</div>
```

```
</div>
<%@include file="footer.jsp" %>
```

Customerlist.jsp

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>


<div class="row" style="height: 600px;width:100%; overflow-y: scroll;">
    <div class="col-sm-2" style="background-color: #0062cc">

        <%@include file="navigation.jsp" %>
    </div>
<div class="col-sm-10">
    <h1>Customer List</h1>
        <table class="table-striped" width="100%">
            <tr><th>Userid</th><th>Name</th><th>Address</th></tr>
<%
    try {
        dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
        Class.forName(ob.DRIVER);
        Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
        Statement st = cn.createStatement();
        String sql1 = "select * from customer";
        //out.println(sql1);
        ResultSet rs = st.executeQuery(sql1);
        while (rs.next()) {
%>
            <tr><td><%=rs.getString(1)%></td>
                <td><%=rs.getString(3)%></td>
                <td><%=rs.getString(5)%></td></tr>

<%
```

```
        }
    } catch (Exception ex) {
        out.println("error" + ex);
    }
%>
        </table>
</div>
</div>
```

%@include file="footer.jsp" %

Feedbacklist.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="row" style="height: 600px;width:100%; overflow-y: scroll;">
    <div class="col-sm-2" style="background-color: #0062cc">
        <%@include file="navigation.jsp" %>
    </div>
<div class="col-sm-10">
    <h1>FeedBack List</h1>
        <table class="table-striped" width="100%">
            <tr><th>Userid</th><th>feedback</th></tr>
<%
    try {
        dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
        Class.forName(ob.DRIVER);
        Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
        Statement st = cn.createStatement();
        String sql1 = "select * from feedback order by id desc";
        //out.println(sql1);
        ResultSet rs = st.executeQuery(sql1);
        while (rs.next()) {
%>
            <tr><td><%=rs.getString(2)%></td>
                <td><%=rs.getString(3)%></td>
                </tr>
<%
        }
    } catch (Exception ex) {
        out.println("error" + ex);
    }
%>
        </table>
</div>
</div>

%@include file="footer.jsp" %
```

Footer.jsp:

```
<div  style="height: 100px; background-color: #cc6600">
  <div  style="color:black; text-align: center;padding: 40px;" >
      <h3> Copy right @ electronicshop.com</h3>
  </div>
</div>
```

Header.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Welcome Admin</title>
    <link href="bootstrap/bootstrap.min.css" rel="stylesheet" type="text/css"/>
```

```
 </head>
 <body>
  <div class="container-fluid">
   <div class="row" style=" height: 150px;">
    <div class="col-sm-2" >
     <img src="images/logo.jpg" style="height: 150px;width: 100%"/>
    </div>
    <div class="col-sm-10" >
     <img src="images/header.png" style="height: 150px;width: 100%"/>
    </div>
   </div>
```

Index (main page). jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="row" style="height: 600px" >
  <div class="col-sm-4" >
</div>
  <div class="col-sm-4" >
    <h1>Login Form</h1>
    <div class="row">
      <div class="col-sm-12">
        <form method="post">
          <%
            if (request.getParameter("submit") != null) {
              try {
                dbcon.DatabaseConnection ob = new
dbcon.DatabaseConnection();
```

```
              Class.forName(ob.DRIVER);
              Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic",
"root", "2658");
              Statement st = cn.createStatement();
              String userid = request.getParameter("userid");
              String password =
request.getParameter("password");
              String query = "select * from admin_login where
userid='" + userid
                  + "' and password='" + password + "' ";
              ResultSet rs = st.executeQuery(query);
              if (rs.next()) {

                session.setAttribute("userid", userid);
                response.sendRedirect("productlist.jsp");
              } else {
                response.sendRedirect("index.jsp?error=true");
              }

            } catch (Exception e) {
              out.println("Error" + e);
            }
          }
        %>
        <label>User ID</label>
        <input type="text" name="userid" class="form-control"/>
        <label>Password</label>
```

```
                <input type="password" name="password" class="form-control"/>
                <input type="submit" name="submit" value="SUBMIT" class="btn-primary"/>
                <%
                if(request.getParameter("error")!=null)
                {
                   out.println("Userid or passwor is wrong!");
                }
                 %>
             </form>
          </div>
       </div>
   </div>
    <div class="col-sm-4" >
 </div>
</div>
<%@include file="footer.jsp" %>
```

Logout. Jsp:

```
<%
session.setAttribute("userid", null);
response.sendRedirect("index.jsp");
%>
```

Navigation. Jsp:

```
<div class="row">
  <div class="col-sm-1">
    </div>
  <div class="col-sm-10" style="padding-top: 20px;">
     <h3><a href="productlist.jsp" style="color:black;
```

```
        text-decoration: none">Home</a></h3>
    <h3><a href="categorylist.jsp" style="color:black; border: .5px;
        text-decoration: none">Category</a></h3>
    <h3><a href="productlist.jsp" style="color:black;
        text-decoration: none">Product</a></h3>
    <h3><a href="customerlist.jsp" style="color:black;
        text-decoration: none">Customer</a></h3>
    <h3><a href="orderlist.jsp" style="color:black;
        text-decoration: none">Order</a></h3>
    <h3><a href="feedbacklist.jsp" style="color:black;
        text-decoration: none">Feedback</a></h3>
    <h3><a href="profilelist.jsp" style="color:black;
        text-decoration: none">Profile</a></h3>
    <h3><a href="logout.jsp" style="color:black;
        text-decoration: none">Logout</a></h3>
  </div>
  <div class="col-sm-1">

  </div>
</div>
```

Orderlist. Jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="row" style="height: 600px;width:100%; overflow-y: scroll;">
    <div class="col-sm-2" style="background-color: #0062cc">

        <%@include file="navigation.jsp" %>
    </div>
<div class="col-sm-10">
    <h1>Order List</h1>
        <table class="table-striped" width="100%">
            <tr><th>Userid</th><th>Order Date</th><th>Payment Date</th>
            <th>Product Name</th><th>Price</th>
            <th>Qty</th><th>total</th></tr>
<%
   try {
       dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
       Class.forName(ob.DRIVER);
       Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
       Statement st = cn.createStatement();
       String sql1 = "select c.userid,c.orderdate,c.paymentdate,c.id as cartid,p.id as
productid,p.name,p.price,c.qty, p.price*c.qty as total from cart c,product p where
c.pid=p.id;";
       //out.println(sql1);
       ResultSet rs = st.executeQuery(sql1);
       while (rs.next()) {
%>
         <tr><td><%=rs.getString(1)%></td>
            <td><%=rs.getString(2)%></td>
            <td><%=rs.getString(3)%></td>
             <td><%=rs.getString(6)%></td>
              <td><%=rs.getString(7)%></td>
               <td><%=rs.getString(8)%></td>
               <td><%=rs.getString(9)%></td>
         </tr>

<%
       }
   } catch (Exception ex) {
       out.println("error" + ex);
   }
%>
       </table>
</div>
```

```
</div>

%@include file="footer.jsp" %
```

Productedit. Jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Productlist. Jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #0062cc">
    <%@include file="navigation.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: antiquewhite">
    <div class="row">
      <div class="col-sm-3">
        <a href="productnew.jsp">
        <button class="btn-primary">New Product</button>
        </a>
      </div>
      <div class="col-sm-3">
```

```
          <a href="productlist.jsp">
          <button class="btn-primary">List Product</button>
          </a>
        </div>
        <div class="col-sm-3"></div>
        <div class="col-sm-3"></div>
      </div>
      <div class="row">
        <div class="col-sm-12" style="text-align: center;">
          <h3>Product List</h3>
          <table class="table-bordered" style="width: 80%">
            <tr>
<th>ID</th>
<th>Catid</th>
<th>Name</th>
<th>Price</th>
<th>Image</th>
</tr>
            <%
            try
            {
              dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
              Class.forName(ob.DRIVER);
              Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
              Statement st=cn.createStatement();
              String query="select * from product";
              ResultSet rs=st.executeQuery(query);
              while(rs.next())
              {
                out.println("<tr>");
                out.println("<td>"+rs.getString(1)+"</td>");
                out.println("<td>"+rs.getString(2)+"</td>");
```

```
           out.println("<td>"+rs.getString(3)+"</td>");
           out.println("<td>"+rs.getString(4)+"</td>");
           out.println("<td><img src='../SingleImage?id="
                   +rs.getString(1)+"' width='50' height='50'/></td>");
           out.println("</tr>");
         }
        }
        catch(Exception ex)
        {
          out.println("Error "+ex);
        }
        %>
      </table>
      </div>
    </div>
  </div>
</div>
%@include file="footer.jsp" %
```

Productnew.jsp:

```
<%@include file="header.jsp" %>
<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #0062cc">
    <%@include file="navigation.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: antiquewhite">
    <div class="row">
      <div class="col-sm-3">
        <a href="productnew.jsp">
        <button class="btn-primary">New Product</button>
        </a>
      </div>
      <div class="col-sm-3">
        <a href="productlist.jsp">
        <button class="btn-primary">List Product</button>
        </a>
      </div>
      <div class="col-sm-3"></div>
      <div class="col-sm-3"></div>
    </div>
    <div class="row">
      <div class="col-sm-12" style="text-align: center;">
        <h3>New Product</h3>
        <form method="post" style="text-align: left;" enctype="multipart/form-data"
             action="../SaveProduct" >
          <div class="form-control">
            <label>Category ID</label>
            <input type="text" name="txtcatid" class="form-control"/>
            <label>Product Name</label>
            <input type="text" name="txtpname" class="form-control"/>
             <label>Product Price</label>
            <input type="text" name="txtprice" class="form-control"/>
             <label>Product Detail</label>
             <textarea  name="txtdetail" class="form-control"></textarea>
             <label>Product Image</label>
             <input type="file" name="txtimage" class="form-control"/>
             <input type="submit" name="submit" value=" SAVE " class="btn-
primary"/>
          </div>
        </form>

      </div>
    </div>
  </div>
</div>
%@include file="footer.jsp" %
```

Profilelist.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<div class="row" style="height: 600px;width:100%; overflow-y: scroll;">
    <div class="col-sm-2" style="background-color: #0062cc">
<%@include file="navigation.jsp" %>
    </div>
<div class="col-sm-10">
        <h1>Change Password</h1>
        <div class="row">
            <div class="col-sm-12">
                <form method="post">
                    <%
                        if (request.getParameter("submit") != null) {
                            try {
                                dbcon.DatabaseConnection ob = new
dbcon.DatabaseConnection();
                                Class.forName(ob.DRIVER);
                                Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
                                Statement st = cn.createStatement();
                                String userid = request.getParameter("userid");
                                String password = request.getParameter("password");
                                String npassword = request.getParameter("npassword");
                                String query = "update admin_login set
password='"+npassword+"' where userid='" + userid
                                        + "' and password='" + password + "' ";
                               int re = st.executeUpdate(query);
                                if(re>0)
                                {
                                    out.println("Password change success<br>");
                                }

                            } catch (Exception e) {
```

```
                out.println("Error" + e);
              }
            }

          %>
          <label>User ID</label>
          <input type="text" name="userid" class="form-control"/>
          <label>Old Password</label>
          <input type="password" name="password" class="form-control"/>
          <label>New Password</label>
          <input type="password" name="npassword" class="form-control"/>
          <input type="submit" name="submit" value="Change Password"
class="btn-primary"/>
          <%
          if(request.getParameter("error")!=null)
          {
            out.println("Userid or passwor is wrong!");
          }



          %>
        </form>
      </div>
</div>
</div>
</div>
```

%@include file="footer.jsp" %

Customer:

singleImage.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

about.jsp:

```
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row" height:600px>
  <div class="col-sm-2">
    <%@include file="sidebar.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: white;width: 100%;"  >

    <h1 style="text-align:center; color: #dc3545">Hi,user!!</h1>
    <p style="color: #491217" > This is e-commerce website.
      In this website, you can shop each and every eletronic item that are used is today's era.
      This is mainly provide user convenience so, that user never let get visit from their home to shop.
      Each and every item that are shown in this website are very affordable to user without any worry!
    </p>
  </div>
</div>
<%@include file="footer.jsp" %>
```

cart. Jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>

<div class="row" style="height: 600px;width: 100%; overflow-y: scroll;">
  <div class="col-sm-2">
    <%@include file="sidebar.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: #ffffff;">
  <h2>Cart Detail</h2>
  <table class="table-striped" width="100%" style="line-height: 60px">
   <tr>
<th>Remove</th>
<th>ID</th>
<th>Name</th>
<th>Image</th>
<th>Price</th>
<th>Qty</th>
<th>Total</th>
   </tr>
   <%
    try {
     dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
     Class.forName(ob.DRIVER);
     Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
     Statement st = cn.createStatement();
     String sql = "select c.id,c.pid,c.qty , p.name,p.price from cart c, product p
where c.pid=p.id and paymentstatus='0'";
     ResultSet rs = st.executeQuery(sql);
     while (rs.next()) {
        %>
        <tr>
          <td>
            <a href="RemoveFromCart?id=<%=rs.getString(1)%>">
            <button class="btn-danger">Remove</button>
            </a>
          </td>
          <td><%=rs.getString(1)%> </td>
          <td><%=rs.getString(4)%> </td>
          <td><img src="SingleImage?id=<%=rs.getString(2)%>" width="50"
height="60" /></td>
          <td><%=rs.getString(5)%> </td>
          <td><%=rs.getString(3)%> </td>
          <td><%
            String price = rs.getString(5);
            String qty = rs.getString(3);
            int total = Integer.parseInt(price) * Integer.parseInt(qty);
            out.println("" + total);
            %> </td>
        </tr>
        <%
            }
          } catch (Exception ex) {
            out.println("Error " + ex);
          }
        %>
      </table>
        <div class="row" style="margin-top: 50px">
          <div class="col-sm-6">
            <a href="index.jsp">
            <button class="btn-primary">Continue to Shop</button>
            </a>
          </div>
          <div class="col-sm-6">
```

```
     <a href="login.jsp">
     <button class="btn-primary">Checkout</button>
     </a>
    </div>
   </div>
 </div>
</div>
<%@include file="footer.jsp" %>
```

contact.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row" style="height: 600px; overflow-y: scroll;">
 <div class="col-sm-2" style="background-color: #33ccff; ">
  <%@include file="sidebar.jsp" %>
 </div>
     <div class="col-sm-10" style="border-width: 5px;background-color: #ffffff;">
      <h1 style="text-align:center">E-shopping</h1>
      <p style="text-align: center">Address: H. no 127-A <br><br>
        Block-B vikas nagar extn.<br><br>
      Uttam nagar,<br><br>
      new delhi<br><br>
      Email: mahehacks@gmail.com
    </p>
    </div>
    %@include file="footer.jsp" %
```

directLogin.jsp:

```
<%@include  file="header.jsp" %>
<%@include file="navigation.jsp" %>
```

```
<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2">
    <%@include file="sidebar.jsp" %>
  </div>
  <div class="col-sm-2" style="background-color: white;width: 100%;">
      </div>
  <div class="col-sm-4" style="background-color: white;width: 100%;">
        <form method="post" action="LoginAuthenticat">
            <h3>Login form</h3>
      <label>Userid</label>
      <input type="text" name="userid" class="form-control"
placeholder="sansa@gmail.com"/>
          <label>Password</label>
      <input type="password" name="password" class="form-control"
placeholder="Xyz123@~"/>
      <br>
      <input type="submit" name="submit" value="Login" class="btn-primary"
style="width:200px; height: 40px"/>
      <%
        if (request.getParameter("error") != null) {
          out.println("<br>Userid or password is incorrect!");
        }
      %>
    </form>
  </div>
  <div class="col-sm-4" style="background-color: white;width: 100%;">
  </div>
</div>
 %@include  file="footer.jsp"  %
```

feedback.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #33ccff; ">
    <%@include file="sidebar.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: white;width: 100%;">
    <div class="container"><br>
      <h2 style=" text-align:center">Feedback form</h2>
      <br>
      <%
    String msg="";
   try {
    if (request.getParameter("submit") != null) {
     dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
     Class.forName(ob.DRIVER);
     Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
     Statement st = cn.createStatement();
     String u = request.getParameter("userid");
     String f = request.getParameter("feedback");
     String query = "insert into feedback (userid,feedback) values('" + u + "','" + f
+ "')";
     int re = st.executeUpdate(query);
     if (re > 0) {
      msg = "<span style='color:green'>Save successfully!</span>";
     }
    }
   } catch (Exception ex) {
    msg = "<span style='color:red'>Error " + ex + "</span>";
```

```
     }
   %>
         <form>
            <label>Userid</label>
            <input type="text" name="userid" class="form-control"/>
            <label>Feed Back</label>
         <textarea name="feedback" class="form-control" rows="5" ></textarea>
                 <br>
         <input type="submit" name="submit" value="submit" class="btn-primary"
style="width:200px; height: 40px"/>
          </form>
   <%=msg%>
      </div>
   </div>
</div>
</div>
```

footer.jsp:

```
<div class="row" style="width: 100%; ">
   <div class="col-sm-12" style="width:50px;">
      <h1>You can find me at:</h1>
   </div>
   <div class="col-sm-4" >
      <a href="https://www.facebook.com/appletreeinfotech" target="_blank">
      <img src="images/face.png" alt="" style="height: 50px"/>
      </a>
   </div>
   <div class="col-sm-4" >
      <img src="images/face.png" alt="" style="height: 50px"/>
   </div>
   <div class="col-sm-4" >
      <img src="images/face.png" alt="" style="height: 50px"/>
   </div>
```

```
</div>
</div>
</body>
</html>
```

header.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>electronics Shop</title>
    <link href="bootstrap/bootstrap.min.css" rel="stylesheet" type="text/css"/>
  </head>
  <body style="background-color:#33ccff">
    <div class="container-fluid">
        <div class="row">
            <div class="col-sm-12">
                <img src='images/electronics-sale-banner-870x350.png' height=100
width=1320/ >
            </div>
        </div>
```

index.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>

<div class="row" style="height: 600px; overflow-y: scroll;">
  <div class="col-sm-2" style="background-color: #33ccff; ">
    <%@include file="sidebar.jsp" %>
  </div>
  <div class="col-sm-10" style="background-color: white;width: 100%;">
```

```
   <%
    try
    {
    dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();

      String sql="select * from product";
      if(request.getParameter("catid")!=null)
      {
          String catid=request.getParameter("catid");
          sql+=" where catid='"+catid+"' ";
      }
     // out.println(sql);
      ResultSet rs=st.executeQuery(sql);
      while(rs.next())
      {
        %>
        <div style=" width: 200px; height: 250px;
           margin: 10px;float:left; text-align: center;">
          <%=rs.getString(3)%>
          <br>
          <a href="productdetail.jsp?id=<%=rs.getString(1)%>">
          <img src="SingleImage?id=<%=rs.getString(1)%>" width="180"
height="200"/>
          </a>
          <br>
          Price:Rs. <%=rs.getString(4)%>
        </div>
        <%
      }
```

```
  }
  catch(Exception ex)
  {
   out.println("error"+ex);
  }
   %>

 </div>
</div>
  %@include file="footer.jsp" %
```

login.jsp:

```
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row">
  <div class="col-sm-4">
    <h3>Login form</h3>
    (if already registered user)
    <br>
    <form method="post" action="LoginAuthenticat">
      <label>Userid</label>
      <input type="text" name="userid" class="form-control"/>

      <label>Password</label>
      <input type="password" name="password" class="form-control"/>

      <br><br>
      <input type="submit" name="submit" value="Login" class="btn-primary"/>
      <%
        if (request.getParameter("error") != null) {
          out.println("<br>Userid or password is incorrect!");
        }
```

```
            %>

        </form>
    </div>
            <div class="col-sm-2"></div>
    <div class="col-sm-6">
        <h3>Registration form</h3>
        (if new user)
        <br>
        <form method="post" action="Register">
            <label>Userid</label>
            <input type="text" name="userid" class="form-control"
placeholder="sansa123"/>
            <label>Password</label>
            <input type="password" name="password" class="form-control"
placeholder="123456@~"/>
            <label>Confirm</label>
            <input type="password" name="cpassword" class="form-control"
placeholder="123456@~"/>
            <label>Name</label>
            <input type="text" name="name" class="form-control"
placeholder="sansa"/>
            <label>Address</label>
            <input type="text" name="address" class="form-control"
placeholder="annapuri, bangluru"/>
            <label>City</label>
            <input type="text" name="city" class="form-control"
placeholder="karnatka"/>
            <label>State</label>
            <input type="text" name="state" class="form-control"
placeholder="karnatka"/>
            <label>Pin</label>
            <input type="text" name="pin" class="form-control"
placeholder="560001"/>
```

```
            <label>Country</label>
            <input type="text" name="country" class="form-control"
placeholder="india"/>
            <label>Mobile</label>
            <input type="text" name="mobile" class="form-control"
placeholder="8210045845"/>
            <label>Email</label>
            <input type="text" name="email" class="form-control"
placeholder="xyz@gmail.com"/>
            <br><br>
            <input type="submit" name="submit" value="Register" class="btn-
primary"/>

        </form>
    </div>
</div>
            <hr>
%@include file="footer.jsp" %
```

navigation.jsp:

```
<div class="row">
    <div class="col-sm-12" style="background-color:#e83e8c; ">
        <table width="100%">
            <tr><td><a href="index.jsp" style="text-decoration: none;color:
black;margin: 10px">HOME</a></td>
                <td><a href="about.jsp" style="text-decoration: none;color:
black">ABOUT</a></td>
                <td><a href="cart.jsp" style="text-decoration: none;color:
black">CART</a></td>
                <td><a href="directlogin.jsp" style="text-decoration: none;color:
black">LOGIN</a></td>
                <td><a href="contact.jsp" style="text-decoration: none;color:
black">CONTACT</a></td>
```

```
            <td><a href="feedback.jsp" style="text-decoration: none;color: black">FEEDBACK</a></td>

         </tr>
       </table>
   </div>
</div>
```

payment.jsp:

```
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row">
   <div class="col-sm-3"></div>
   <div class="col-sm-6">
      <h3>Payment Form</h3>
      <br>
      <script>
         function validate(){
         var cardnumber = document.getElementById("cardnumber").value;
         if (cardnumber === ""){
         alert("please enter cardnumber.");
         return false;
         var cardholder = document.getElementById("cardholder").value;
         if (cardholder === "")
         {
         alert("Please enter cardholder name.");
         return false;
         }
         var expirydate = document.getElementById("expirydate").value;
         if (expirydate === "")
         {
         alert("Please enter Expiry date.");
         return false;
```

```
          }
         var cvv = document.getElementById("cvv").value;
          if (cvv == "")
          {
          alert("Please enter cvv number");
          return false;
          }
          }
      </script>
      <form method="post" action="Payment" onsubmit="return validate()">
          <label> Total Amount </label>
          <%
             cart.TotalAmount obt = new cart.TotalAmount();
             int total = obt.gettotal();
          %>
          <input type="text" name="amount" readonly class="form-control" 
value="<%=total%>"/>
          <label> Card Number</label>
          <input type="text" name="cardnumber" id="cardnumber" class="form-
control"/>
          <label> Card Holder </label>
          <input type="text" name="cardholder" id="cardholder" class="form-
control"/>
          <label> Expiry Date </label>
          <input type="text" name="expire"  id="expirydate" class="form-control"/>
          <label> CVV </label>
          <input type="password" name="cvv" id="cvv" class="form-control"/>
          <br><br>
          <input type="submit" name="submit" value="Pay Now" class="btn-
primary"/>
      </form>
      <br><br>
   </div>
   <div class="col-sm-3"></div>
```

```
</div>
```

%@include file="footer.jsp" %

productdetail.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row" style="height: 600px;width: 100%; overflow-y: scroll;">
    <div class="col-sm-12" style="background-color: #33ccff;">
        <style>
            h2{
                border-style:solid;
                border-width: 1px;
                border-color: #721c24;
                padding:30px;
            }
        </style>
        <h2>Product Detail</h2>

        <%
            try {
                String id = request.getParameter("id");
                dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
                Class.forName(ob.DRIVER);
                Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
                Statement st = cn.createStatement();
                String sql = "select * from product where id='" + id + "'";
                ResultSet rs = st.executeQuery(sql);
                while (rs.next()) {
        %>
        <div class="row">
            <div class="col-sm-1"></div>
            <div class="col-sm-5">
                <img src="SingleImage?id=<%=rs.getString(1)%>" width="300"
height="400"/>
            </div>
            <div class="col-sm-5">
                <form method="post" action="AddtoCart">
                    Product ID:
                    <input type="text" name="id" readonly
value="<%=rs.getString(1)%>"/>
                    <br><br>
                    Name:<%=rs.getString(3)%>
                    <br><br>
                    Price:Rs. <%=rs.getString(4)%><br><br>
                    Qty
                    <input type="text" name="qty"/>
                    <br><br>
                    <input type="submit" name="addtocart" class="btn-primary"
value="Add to Cart"/>
                </form>
            </div>
            <div class="col-sm-1"></div>
        </div>
        <div class="row">
            <div class="col-sm-12">
                <%=rs.getString(5)%>
            </div>
        </div>
        <%
                }
            } catch (Exception ex) {
                out.println("error" + ex);
            }
        %>

    </div>
</div>
%@include file="footer.jsp" %
```

register.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

sidebar.jsp

```
<%@page import="java.sql.*"%>

<div class="row" >
  <div class="col-sm-12">
     Category
    <hr>
    <%
    try
    {
      dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();
      String sql="select * from category order by name";
      ResultSet rs=st.executeQuery(sql);
      while(rs.next())
      {
          %>
        <a href="index.jsp?catid=<%=rs.getString(1)%>" style="color:red"
><%=rs.getString(2)%></a> <br><br>
        <%
      }
    }
    catch(Exception ex)
    {
      out.println("error"+ex);
    }

    %
  </div>
</div>
```

success.jsp:

```
<%@page import="java.sql.*"%>
<%@include file="header.jsp" %>
<%@include file="navigation.jsp" %>
<div class="row" style="height: 600px; overflow-y: scroll;">

  <div class="col-sm-12" style="background-color: #33ccff;">
    <h5>Your order is success</h5>
    <h2>Order Detail</h2>
    <table class="table-striped" width="100%" style="line-height: 60px">
```

```
   <tr><th>ID</th><th>Name</th><th>Image</th><th>Price</th>
    <th>Qty</th><th>Total</th>
   </tr>
   <%
    try {
     dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
     Class.forName(ob.DRIVER);
     Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
     Statement st = cn.createStatement();
     String sql = "select c.id,c.pid,c.qty , p.name,p.price from cart c, product p
where c.pid=p.id";
     ResultSet rs = st.executeQuery(sql);
     while (rs.next()) {
   %>
   <tr>
      <td><%=rs.getString(1)%> </td>
    <td><%=rs.getString(4)%> </td>
    <td><img src="SingleImage?id=<%=rs.getString(2)%>" width="50"
height="60" /></td>
    <td><%=rs.getString(5)%> </td>
    <td><%=rs.getString(3)%> </td>
    <td><%
     String price = rs.getString(5);
     String qty = rs.getString(3);
     int total = Integer.parseInt(price) * Integer.parseInt(qty);
     out.println("" + total);
     %> </td>
   </tr>
   <%
     }
    } catch (Exception ex) {
     out.println("Error " + ex);
```

```
        }
      %>
    </table>
        </div>
</div>
    %@include file="footer.jsp" %
```

JAVA (SERVELET PAGE):

Addtocart.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

@WebServlet(urlPatterns = {"/AddtoCart"})
public class AddtoCart extends HttpServlet {

  protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
      /* TODO output your page here. You may use following sample code. */
      out.println("<!DOCTYPE html>");
      out.println("<html>");
      out.println("<head>");
      out.println("<title>Servlet AddtoCart</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Servlet AddtoCart at " + request.getContextPath() + "</h1>");
      out.println("</body>");
      out.println("</html>");
    }
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
  }

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    try {
      dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();
      String pid=request.getParameter("id");
      String qty=request.getParameter("qty");
      java.util.Date dt=new java.util.Date();
      String orderdate=(dt.getYear()+1900)+"-"+(dt.getMonth()+1)+"-"+dt.getDate()
          +" "+dt.getHours()+":"+dt.getMinutes()+":"+dt.getSeconds();
      String query="insert into cart(pid,qty,orderdate) values('"+pid+"','"
          +qty+"','"+orderdate+"')";
      int re=st.executeUpdate(query);
      if(re>0)
      {
        response.sendRedirect("cart.jsp");
      }

    } catch (Exception e) {
      out.println("Error "+e);
    }
  }
  @Override
  public String getServletInfo() {
    return "Short description";
  }
}
```

<u>loginAutheticate.java:</u>

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet(urlPatterns = {"/LoginAuthenticat"})
public class LoginAuthenticat extends HttpServlet {



 protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
```

```
     throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  try (PrintWriter out = response.getWriter()) {
   /* TODO output your page here. You may use following sample code. */
   out.println("<!DOCTYPE html>");
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Servlet LoginAuthenticat</title>");
   out.println("</head>");
   out.println("<body>");
   out.println("<h1>Servlet LoginAuthenticat at " + request.getContextPath() +
"</h1>");
   out.println("</body>");
   out.println("</html>");
  }
 }

 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
 @Override
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
     throws ServletException, IOException {
  processRequest(request, response);
 }

 @Override
 protected void doPost(HttpServletRequest request, HttpServletResponse response)
     throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  PrintWriter out =response.getWriter();
  try {
   dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
   Class.forName(ob.DRIVER);
```

```
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();
      String userid=request.getParameter("userid");
      String password=request.getParameter("password");
      String query="select * from customer where userid='"+userid
              +"' and password='"+password+"' ";
      ResultSet rs=st.executeQuery(query);
      if(rs.next())
      {
        HttpSession session=request.getSession();
        session.setAttribute("userid", userid);
        response.sendRedirect("payment.jsp");
      }
      else
      {
        response.sendRedirect("login.jsp?error=true");
      }

    } catch (Exception e) {
      out.println("Error"+e);
    }
  }

  @Override
  public String getServletInfo() {
    return "Short description";
  }// </editor-fold>

}
```

payment.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet(urlPatterns = {"/Payment"})
public class Payment extends HttpServlet {

  protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
      /* TODO output your page here. You may use following sample code. */
      out.println("<!DOCTYPE html>");
      out.println("<html>");
      out.println("<head>");
      out.println("<title>Servlet Payment</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Servlet Payment at " + request.getContextPath() + "</h1>");
      out.println("</body>");
      out.println("</html>");
    }
  }
  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
    processRequest(request, response);
  }

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
      Statement st = cn.createStatement();
      String amount = request.getParameter("amount");
      String cardholder = request.getParameter("cardholder");
      HttpSession session = request.getSession();
      String userid = session.getAttribute("userid").toString();
      java.util.Date dt = new java.util.Date();
      String pdate = (dt.getYear() + 1900) + "-" + (dt.getMonth() + 1) + "-"
              + dt.getDate();
      String query = "insert into payment (userid,amount,paydate,cardholder)
values('"
              + userid + "','" + amount + "','" + pdate + "','" + cardholder + "')";
      int re = st.executeUpdate(query);
      if (re > 0) {
        query = "update cart set userid='" + userid + "', paymentdate='"
                + pdate + "', paymentstatus='1' ";
        re = st.executeUpdate(query);
        if (re > 0) {
          response.sendRedirect("success.jsp");
        }
      }

    } catch (Exception e) {
      out.println("Error" + e);
    }
  }
  @Override
  public String getServletInfo() {
    return "Short description";
  }// </editor-fold>

}
```

Register.java:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet(urlPatterns = {"/Register"})
```

```
public class Register extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Register</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet Register at " + request.getContextPath() +
"</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
            Class.forName(ob.DRIVER);
            Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic", "root",
"2658");
            Statement st = cn.createStatement();
            String userid = request.getParameter("userid");
            String password = request.getParameter("password");
            String name = request.getParameter("name");
            String address = request.getParameter("address");
            String city = request.getParameter("city");
            String state = request.getParameter("state");
            String pin = request.getParameter("pin");
            String country = request.getParameter("country");
            String mobile = request.getParameter("mobile");
            String email = request.getParameter("email");

            String query = "insert into customer (userid, password, name, address, city,
state, pin, country, mobile,email) "
          + "values('" + userid + "','" + password + "','" + name + "','" + address
+ "','" + city + "','" + state
          + "','" + pin + "', '" + country + "','" + mobile + "','" + email + "')";
            int re = st.executeUpdate(query);
            if (re > 0) {
              HttpSession session = request.getSession();
              session.setAttribute("userid", userid);
              response.sendRedirect("payment.jsp");
            }

        } catch (Exception e) {
            out.println("Error" + e);
        }
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Removefromcart.java:

```
import dbcon.DatabaseConnection;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
@WebServlet(urlPatterns = {"/RemoveFromCart"})
public class RemoveFromCart extends HttpServlet {
 protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
      throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  try (PrintWriter out = response.getWriter()) {
   /* TODO output your page here. You may use following sample code. */
   out.println("<!DOCTYPE html>");
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Servlet RemoveFromCart</title>");
   out.println("</head>");
   out.println("<body>");
```

```
      out.println("<h1>Servlet RemoveFromCart at " + request.getContextPath() +
"</h1>");
      out.println("</body>");
      out.println("</html>");
    }
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    try
    {
      dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();
      String id=request.getParameter("id");
      String query="delete from cart where id='"+id+"'";
      int re=st.executeUpdate(query);
      if(re>0)
      {
        response.sendRedirect("cart.jsp");
      }
    }
    catch(Exception ex)
    {
      out.println("Error "+ex);
  }
    }

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
    processRequest(request, response);
  }
  @Override
  public String getServletInfo() {
    return "Short description";
  }// </editor-fold>

}
```

saveproduct.java:

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;

@MultipartConfig(fileSizeThreshold =1024*1024*10,
        maxFileSize = 1024*1024*50,
        maxRequestSize =1024*1024*100 )
@WebServlet(urlPatterns = {"/SaveProduct"})
```

```
public class SaveProduct extends HttpServlet {

  protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
      /* TODO output your page here. You may use following sample code. */
      out.println("<!DOCTYPE html>");
      out.println("<html>");
      out.println("<head>");
      out.println("<title>Servlet SaveProduct</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Servlet SaveProduct at " + request.getContextPath() +
"</h1>");
      out.println("</body>");
      out.println("</html>");
    }
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
  }

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    //logic to save data
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      String catid = request.getParameter("txtcatid");
      String name = request.getParameter("txtpname");
      String price = request.getParameter("txtprice");
      String detail = request.getParameter("txtdetail");
      Part image = request.getPart("txtimage");
      InputStream is=image.getInputStream();
      dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
           Connection
     cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root
     ","2658");
      String query="insert into product(catid,name,price,detail,image) values
(?,?,?,?,?)";
      PreparedStatement st=cn.prepareStatement(query);
      st.setString(1, catid);
      st.setString(2, name);
      st.setString(3, price);
      st.setString(4, detail);
      st.setBlob(5, is);
      int re=st.executeUpdate();
      if(re>0)
      {
        out.println("Save successfully");
      }
    } catch (Exception e) {
      out.println("Error " + e);
    }

  }

  @Override
  public String getServletInfo() {
    return "Short description";
  }// </editor-fold>

}
```

singleImage.java:

```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = {"/SingleImage"})
public class SingleImage extends HttpServlet {

  protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
      /* TODO output your page here. You may use following sample code. */
      out.println("<!DOCTYPE html>");
      out.println("<html>");
      out.println("<head>");
      out.println("<title>Servlet SingleImage</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Servlet SingleImage at " + request.getContextPath() +
"</h1>");
```

```
      out.println("</body>");
      out.println("</html>");
    }
  }
  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
      dbcon.DatabaseConnection ob = new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection cn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","265
8");
      Statement st = cn.createStatement();
      String id=request.getParameter("id");
      String query = "select image from product where id='"+id+"'";
      ResultSet rs = st.executeQuery(query);
      if (rs.next()) {
        response.reset();
        Blob blb = rs.getBlob(1);
        byte[] bdata = blb.getBytes(1, (int) blb.length());
        OutputStream output = response.getOutputStream();
        output.write(bdata);
        output.close();
        rs.close();
      }
    } catch (Exception ex) {
    }
  }
  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
    processRequest(request, response);
  }
  @Override
  public String getServletInfo() {
    return "Short description";
  }// </editor-fold>

}
```

Total amount.java:

```
package cart;
import java.sql.*;
public class TotalAmount {
 public int gettotal()
  {
    try {
      dbcon.DatabaseConnection ob=new dbcon.DatabaseConnection();
      Class.forName(ob.DRIVER);
      Connection
cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/electronic","root","
2658");
      Statement st=cn.createStatement();
    String sql="select c.qty,p.price from cart c, product p where c.pid=p.id and
c.paymentstatus=0";
    ResultSet rs=st.executeQuery(sql);
    int total=0;
    while(rs.next())
    {
      int q=Integer.parseInt(rs.getString(1));
      int p=Integer.parseInt(rs.getString(2));
      int t=q*p;
      total=total+t;
    }
    return total;

    } catch (Exception e) {
      return 0;
    }

  }
}
```

Dbconnection:

```
package dbcon;
public class DatabaseConnection {
  public final String DRIVER = "com.mysql.jdbc.Driver";
  public final String URL = "jdbc:mysql://localhost:3306/electronic";
  public final String USER = "root";
  public final String PASSWORD = "2658";
}
```

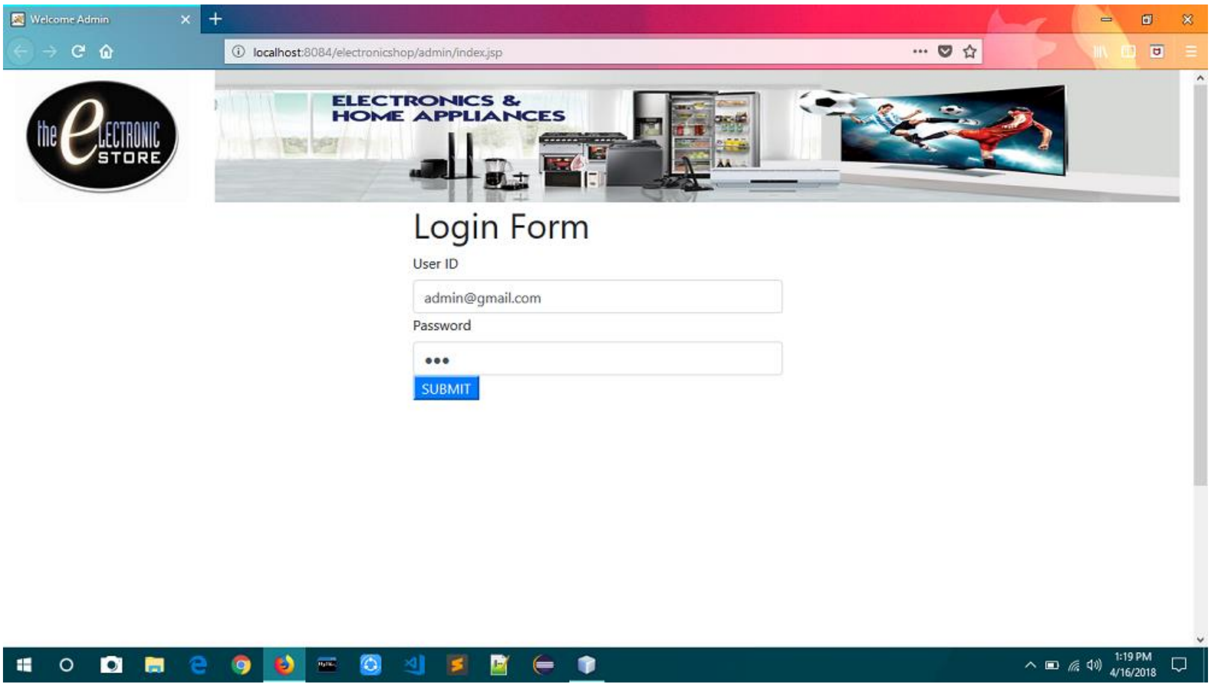
MySQL library:

mysql-connector-java-5.1.23-bin.jar

SNAPSHOT:

ADMIN BROWSER:

ADMIN HOME_PAGE:


Welcome Admin
localhost:8084/electronicshop/admin/index.jsp
the ELECTRONIC STORE
ELECTRONICS & HOME APPLIANCES
SUBMIT
1:19 PM
4/16/2018

Login Form

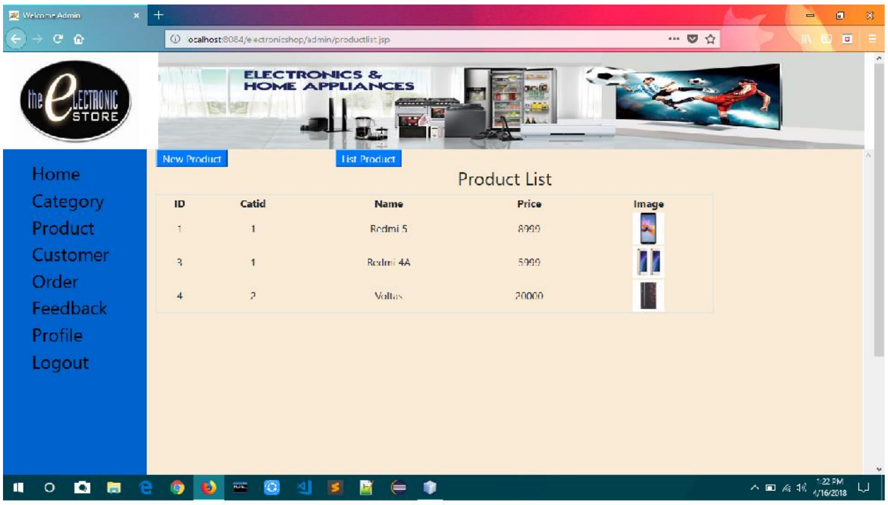
User ID

admin@gmail.com

Password

●●●

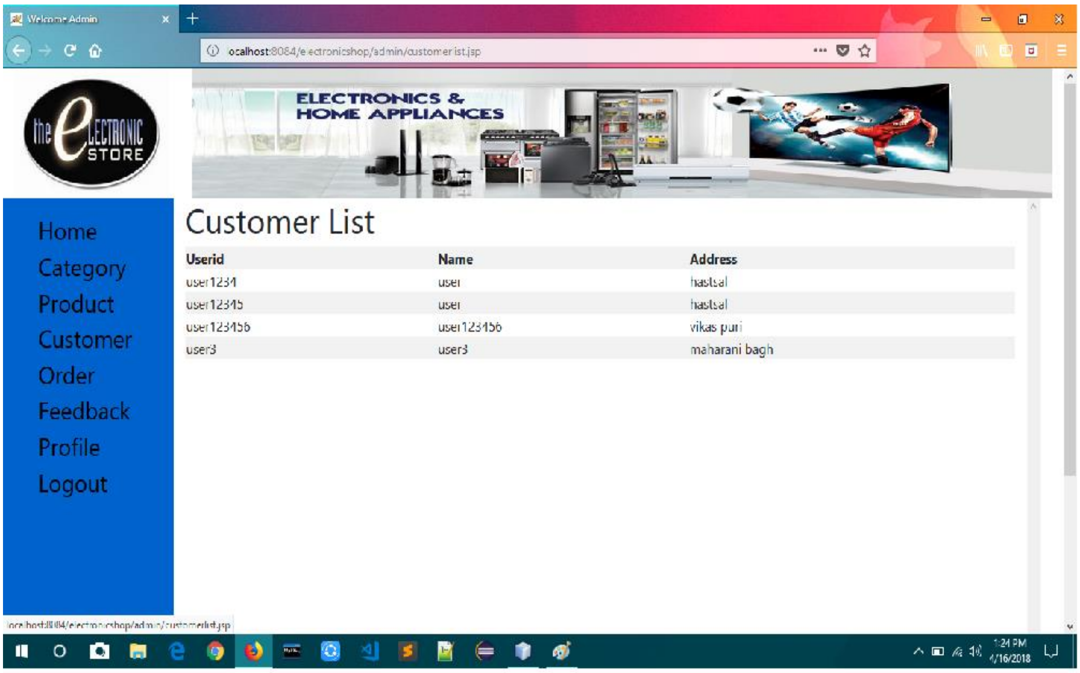
AFTER LOGIN:



Product List

ID	Catid	Name	Price	Image
1	1	Redmi 5	8999	
3	1	Redmi 4A	5999	
4	2	Voltas	20000	

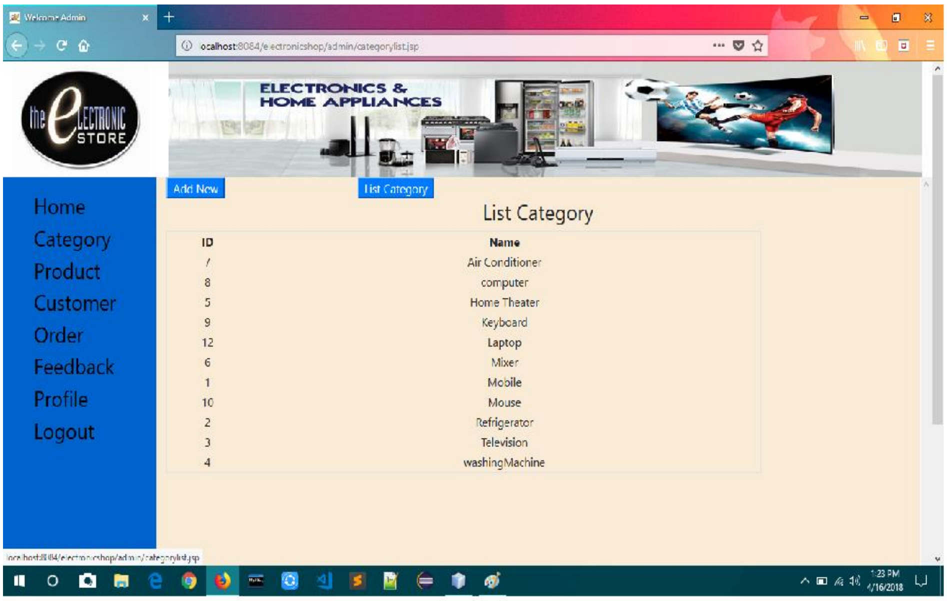
CUSTOMER LIST:



Customer List

Userid	Name	Address
user1234	user	hastsal
user12345	user	hastsal
user123456	user123456	vikas puri
user3	user3	maharani bagh

CATEGORY LIST:

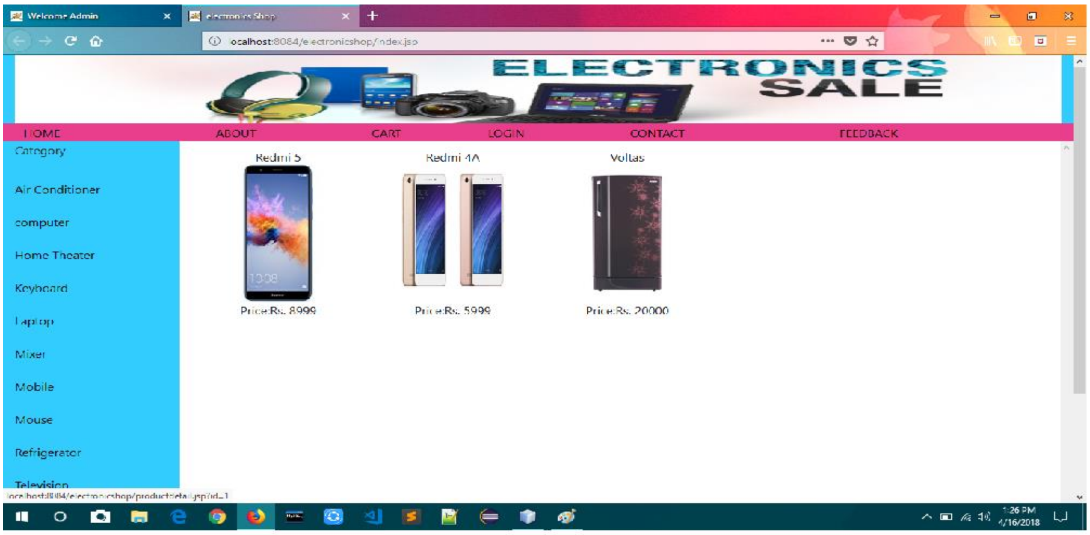


List Category

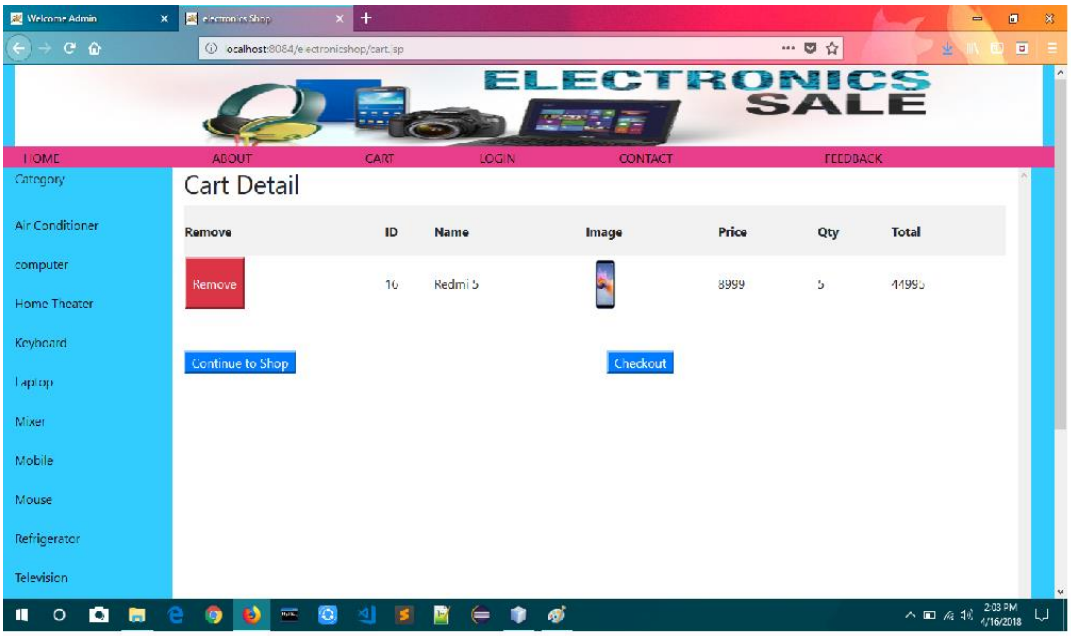
ID	Name
7	Air Conditioner
8	computer
5	Home Theater
9	Keyboard
12	Laptop
6	Mixer
1	Mobile
10	Mouse
2	Refrigerator
3	Television
4	washingMachine

CUSTOMER BROWSER:

HOME_PAGE:



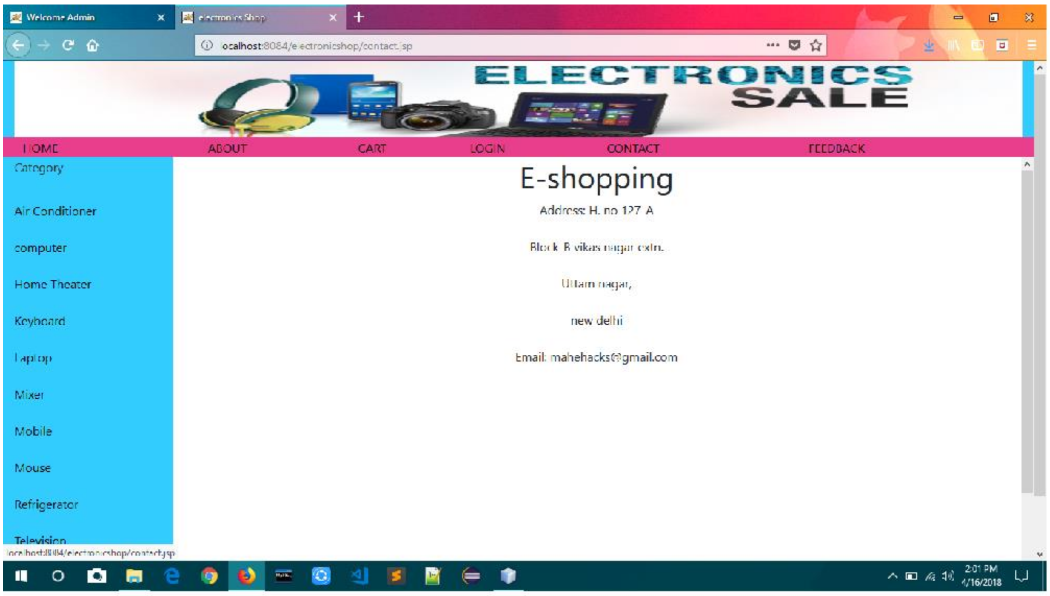
CART:



Cart Detail

Remove	ID	Name	Image	Price	Qty	Total
Remove	16	Redmi 5		8999	5	44995

CONTACT PAGE:



E-shopping

Address: H. no 127 A

Block B vikas nagar extn.

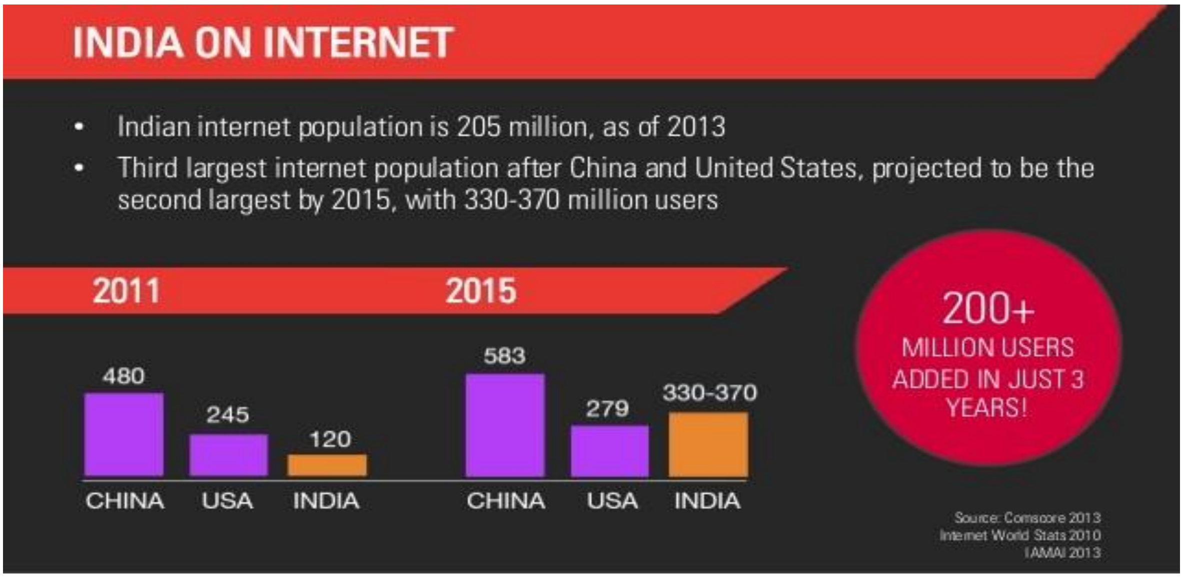
Uttam nagar,

new delhi

Email: mahehacks@gmail.com

Scope

Electronic Commerce is more than just buying and selling products online. It also includes the entire online process of developing, marketing, selling, delivering, servicing and paying for products and services. India has shown tremendous growth in the E-commerce segment. With an **internet user base of over 300 million**, India has third largest internet population after US & China (see info-graphic below).


INDIA ON INTERNET

- Indian internet population is 205 million, as of 2013
- Third largest internet population after China and United States, projected to be the second largest by 2015, with 330-370 million users

Source: Comscore 2013
Internet World Stats 2010
IAMAI 2013

India has witnessed a major breakthrough E-commerce success stories particularly in **e-retail in Consumer Electronics & Fashion Apparel & Home Furnishing segments**. E-commerce creates new opportunities for **entrepreneurial start-ups**. Ease of Internet access, Safe and secure payment modes coupled with aggressive marketing by E-Commerce Giants has revolutionized this segment. Rapid development in mobile technology has given way to **Mobile Commerce** with many E-Commerce companies shifting to App only model.

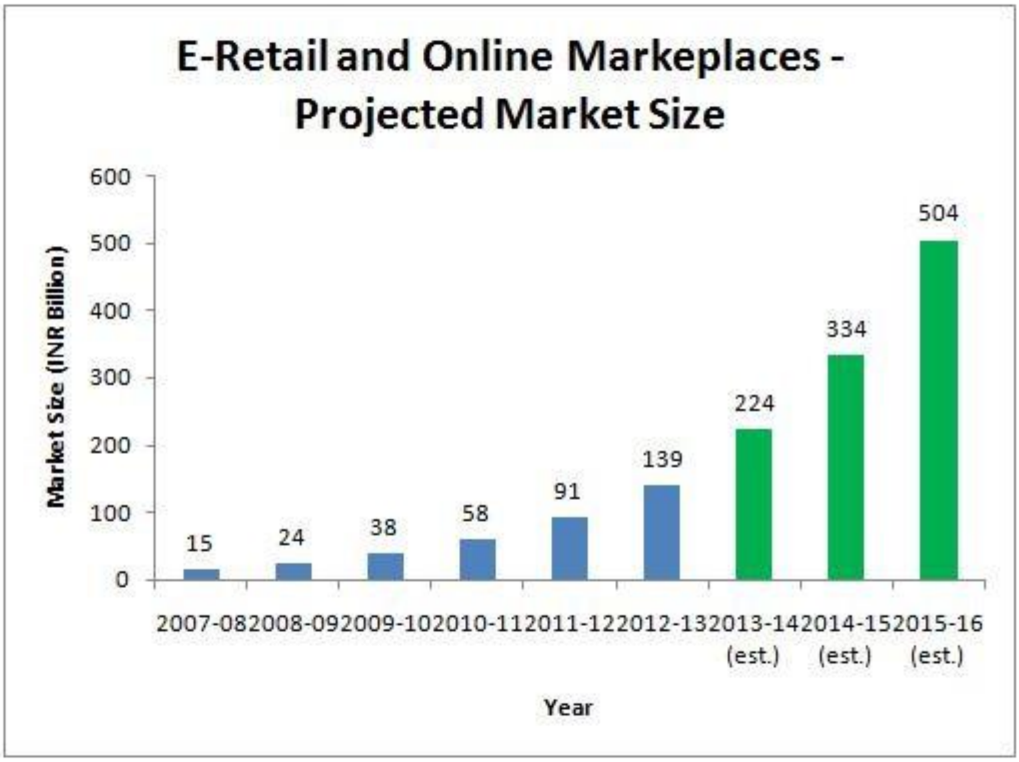
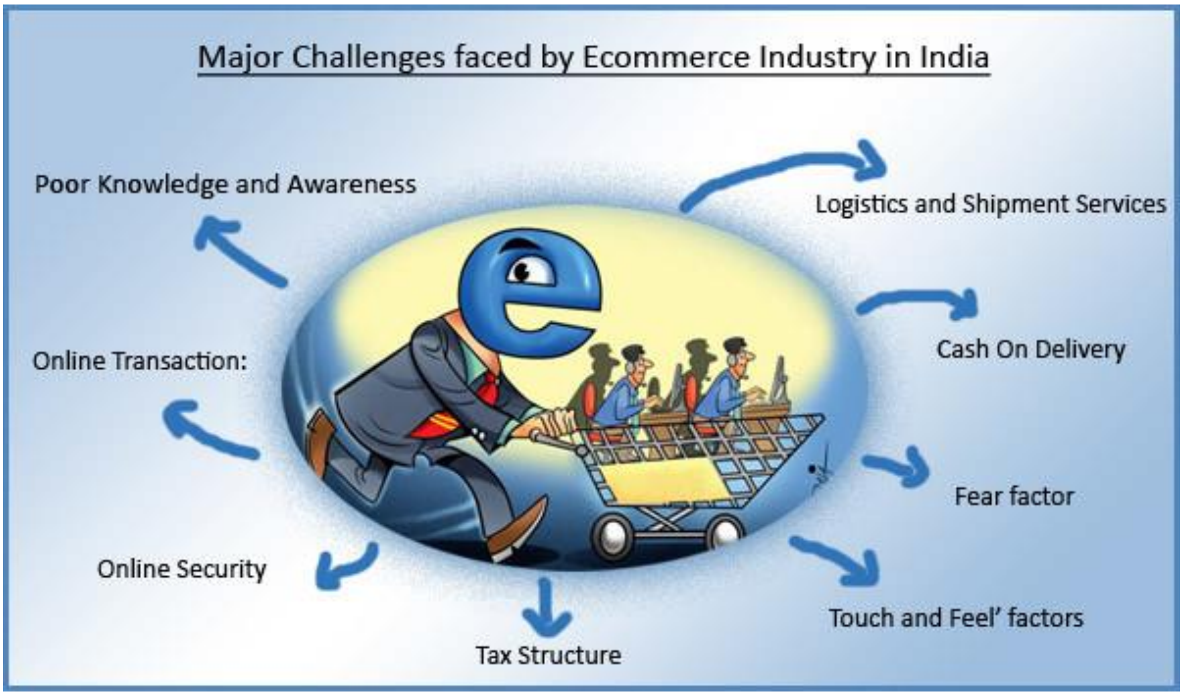


Exhibit 2 Growth Forecast for online retailing and marketplaces in India Source: CRISIL

Challenges

E-Commerce, in-spite of the opportunities it presents also has poses certain challenges which are sometimes too much to handle for start-ups:


Major Challenges faced by Ecommerce Industry in India
Poor Knowledge and Awareness
Logistics and Shipment Services
Cash On Delivery
Online Transaction:
Fear factor
Online Security
Touch and Feel' factors
Tax Structure

- **E-Infrastructural Issues:** Internet is the backbone of e-commerce. Unfortunately, in India internet penetration is so far dismally low at 0.5 per cent of the population, penetration of personal computer (PC) as low as 3.5 per thousand of population and penetration of telephone only 2.1 per cent of population, **e-commerce remains far away from the common man**.

- **Branding & Marketing:** To get people to come on an e-Commerce site and make a purchase involves heavy cost due to branding and marketing. This cost is significant and can be brought down to cost per customer, if the volumes permit to do so. **Experts say that the average figure for this metric in the current e-Commerce ecosystem is between INR 500 – 1000 customer**, which isn't sustainable for even medium sized companies, let alone early stage ones.

- **Declining Margins:** With the introduction of a large number of players in the already competitive e-commerce market, **the customer is pampered by offering huge discounts, offers, taking returns etc. resulting in razor-thin margins**.

- **Logistics & Supply Chain: Logistics failure in any area can mean detrimental damage to a startup's future** and can hurt the brand overall. Add to this the need for a guaranteed return policy. Getting this right is a challenge.

- **Tax related issues:** Tax rate system of Indian market is another factor for lesser growth rate of eCommerce in India in comparison to other developed countries like USA and UK. In those countries, tax rate is uniform for all sectors whereas **tax structure of India varies from sector to sector**. This factor creates accounting problems for the Indian online business companies.

- **Touch and Feel:** Indian customers are more comfortable in buying products physically. Companies dealing with products like apparel, handicrafts, jewelry have to face challenges to sell their products as the **buyers want to see and touch before they buy these stuffs**.

Future

E-Commerce is here to stay !!

- **Social Media: Majority of online buying decisions are made on Social Media**. Social network like Facebook, LinkedIn, Twitter, Google+, Pinterest etc have become a medium for easy log-in and purchase. Moreover, the clients can stay updated via the posts published on this media. Further, the advertising & promotions on these social sites has increased the chances of success of generating transactions to many folds.

- **Drone Delivery:** Companies have been working their way around to innovate the delivery process to shorten human effort as well as time. The answer to these problems is Delivery by Drones. DGCA is now fast tracking the process of issuing guidelines for the use of drones for civil purposes in India. If everything goes as per the plan**, then India might become the first country in the world to allow the use of drones for civil purposes.**

- **App only Approach:** Statistics suggest the future of internet lies in mobiles. Experts say more than 580 million people in India will use the Internet by 2018, and 70-80% of them will access the Web on mobile phones. This will cause all major players to switch to app only model. About two-thirds of its online traffic of Flipkart comes from users in small cities and towns. Flipkart's app-only approach assumes larger significance in these places where most people don't own desktop computers and have limited access to broadband.

REFERENCE:

- www.oracle.com
- www.netBeans.com
- www.w3school.com
- www.wikipedia.com
- www.google.com
- www.tutorialpoint.com
- www.linkedin.com
- www.stackoverflow.com
- www.quora.com
- The complete java reference(TMH publication)
- Head first java by Kathy sierra and bert bates
- Let us java by yashawant kanetkar