ONLINE SHOPPING SHOES
SNEAKER SHOP

CLASS: SE1401

GROUP 4: Lê Đức Hòa(Leader): CE140469

Nguyễn Thị Diễm Hương: CE140435

Võ Anh Nhiều: CE14037

Nguyễn Tân Đại Phát: CE140539

TABLE CONTENT

INTRODUCTION

Today the Internet has become a popular and essential service and has a profound impact on the habits, activities and entertainment of many people. With the rapid development of the Internet, the forms of buying and selling goods to people have become more diverse and developed. Web applications are becoming more and more popular. Facing that demand, along with the course requirements, our team decided to select the topic of Building an Online Sales Website, namely shoes.

However, due to many limitations the topic of the construction group cannot avoid shortcomings. We hope to have teachers and friends contribute ideas to make the program better and bertter.

PART I: SURVEY OF THE SITUATION OF THE SYSTEM

- Shoes Shop is a store specializing in supplying shoes from other countries to Vietnam with a variety of designs, reasonable prices, good quality, and prestige. Currently the store is the main distributor for major shoe brands in the world. Store structure includes: Store owner, manager, salesperson, warehouse department and shipping department.

- Customers of the store have 2 types: 1 is a regular customer (customers who want to enter a large number of resale or other small retail stores), 2 are irregular customers (retail customers). Retail customers can come to the store to buy directly. For regular customers, big customers will pre-order the shoes as ordered and have their own preferential policies.

- The need to expand the market as well as promote the brand and increase customers is a concern of the store. Therefore, it is necessary to set up a product introduction website as well as sell products to remote customers who want to buy products, those who want to buy online at home as well as be able to support customers with the information they need. More specific design and advice for customers.

PART II: OBJECTIVES OF BUILDING WEBSITE

1. The object of the website

Website is built to serve two main objects: Admin (administrator) and Customers with the following functions:

➢ **Admin:**
- Login Website
- View, update, delete product information
- Manage orders
- View, delete customer information but not the right to change it.

➢ **Customer:**

★ *Guest:* Visitor
- See product information
- Member registration

★ *User:*
- Already have an account
- Have the right to log in, log out, change passwords
- Order products
- Enjoy priority rights like the latest product announcements

2. Characteristics:

Built a simple, user-friendly, easy-to-use online shoe sale system that allows customers to view information and place orders online, and administrators manage product and user information.

Website designed with:

-The interface is harmonious, friendly and easy to use.

- The home page will display a list of the latest and best-selling products to make it easier for users to search.

- Customers can easily find detailed information about the type of shoes they are interested in.

-Customer can choose to buy the shoes they need based on their financial ability and functionality needed by adding to the shopping cart

-With registration and login function

Modules:

- **Product module:**

Display information and product classification in virtual store. Products displayed on the website will be displayed with full information about such products such as pictures, product names, outstanding features of products, prices, ...

- **Shopping cart module:**

  When referring to the full information about the product, customers can order products right on the Website through the shopping cart function without having to go to the transaction location, the cart is simulated as a shopping cart in reality. Products purchased. When choosing a customer cart payment Support [College Program Proposal] must contain all personal information, which is stored and processed by the system.

- **Module member registration and system login:**

Each customer dealing at the Website will be entitled to register for a separate account. This account will be used when required by the system. An account registered by the customer will store the personal information of the customer.

- **Product search module:**

Customers will be provided with the search function on the Website.

- **Product management module, orders:**

Administrators can update information of items, types of goods, manage order information.

3. Hardware and Software Requirement

a) Hardware

Servers

Hardware requirement (minimal configuration)	Hardware requirement (recommended configuration)
• **CPU:** 500 MHz processor • **RAM:** 2GB RAM • **HDD:** 500MB of free disk space	• **CPU:** 4GHZ Dual Core or 3GHZ processor • **RAM:** 4GB RAM • **HDD:** 1GB of free disk space

Client

Hardware requirement (minimal configuration)	Hardware requirement (recommended configuration)
• **CPU:** 500 MHz processor • **RAM:** 2GB RAM • **HDD:** 500MB of free disk space	• **CPU:** 4GHZ Dual Core or 3GHZ processor • **RAM:** 4GB RAM • **HDD:** 1GB of free disk space

b) Software

Server

Software requirement
• Window 7 or higher

• JRE 1.7 or higher

- **Client**

Software requirement
• Window 7 or higher • JRE 1.7 or higher

III. THEORY

1. Introduce to JSP

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. This tutorial will teach you how to use Java Server Pages to develop your web applications in simple and easy steps.

Advantages

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including **JDBC, JNDI, EJB, JAXP,** etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

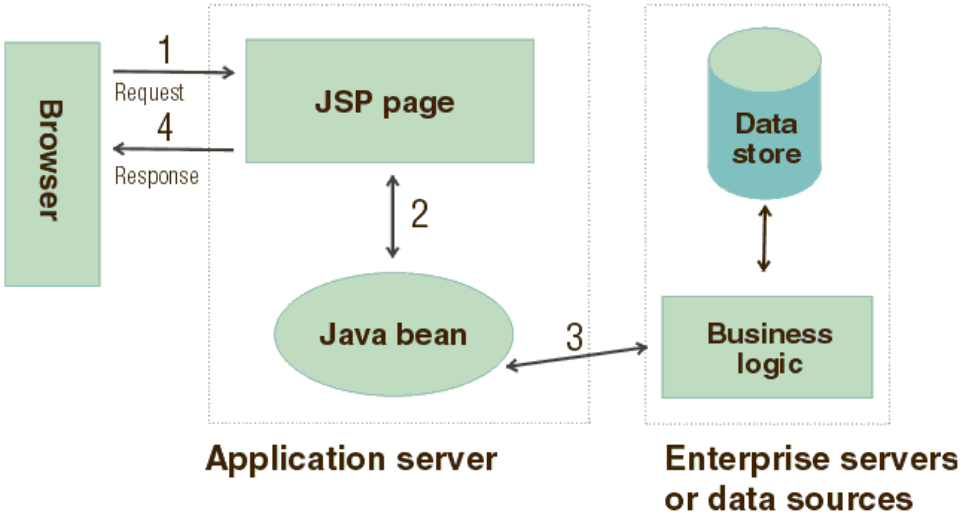


Figure 2: *Example JSP connect browser and data store*

2. Introduction to MVC in JSP

MVC stands for Model View and Controller. It is a **design pattern** that separates the business logic, presentation logic and data.

- M stands for Model
- V stands for View
- C stands for controller.

Model Layer

- This is the data layer which consists of the business logic of the system.
- It consists of all the data of the application
- It also represents the state of the application.
- It consists of classes which have the connection to the database.
- The controller connects with model and fetches the data and sends to the view layer.
- The model connects with the database as well and stores the data into a database which is connected to it.

View Layer

- This is a presentation layer.
- It consists of HTML, JSP, etc. into it.
- It normally presents the UI of the application.
- It is used to display the data which is fetched from the controller which in turn fetching data from model layer classes.
- This view layer shows the data on UI of the application.

Controller Layer

- It acts as an interface between View and Model.
- It intercepts all the requests which are coming from the view layer.
- It receives the requests from the view layer and processes the requests and does the necessary validation for the request.
- This requests is further sent to model layer for data processing, and once the request is processed, it sends back to the controller with required information and displayed accordingly by the view.

Advantages

- Easy to maintain.
- Easy to extend.
- Easy to test.
- Navigation control is centralized.

3. Introduction to Servlet

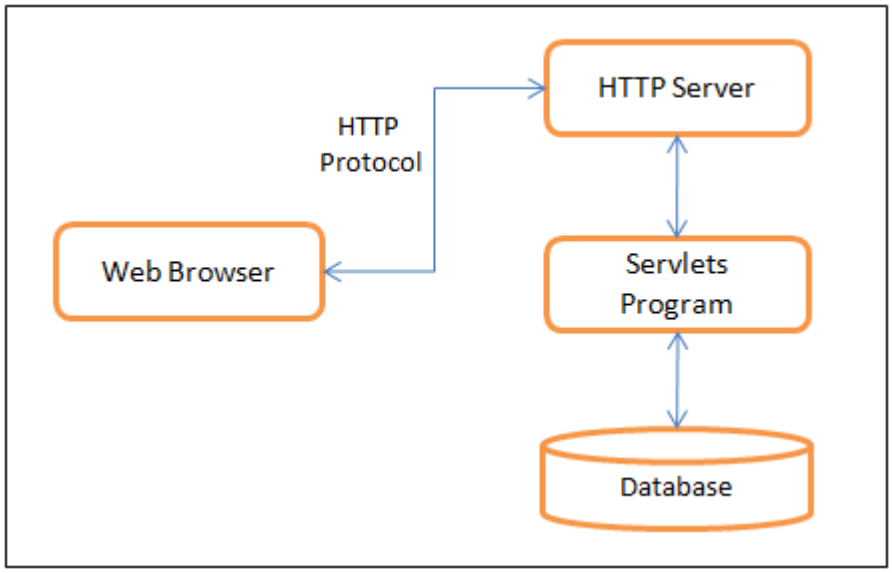
Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.



PART IV: SYSTEM ANALYSIS

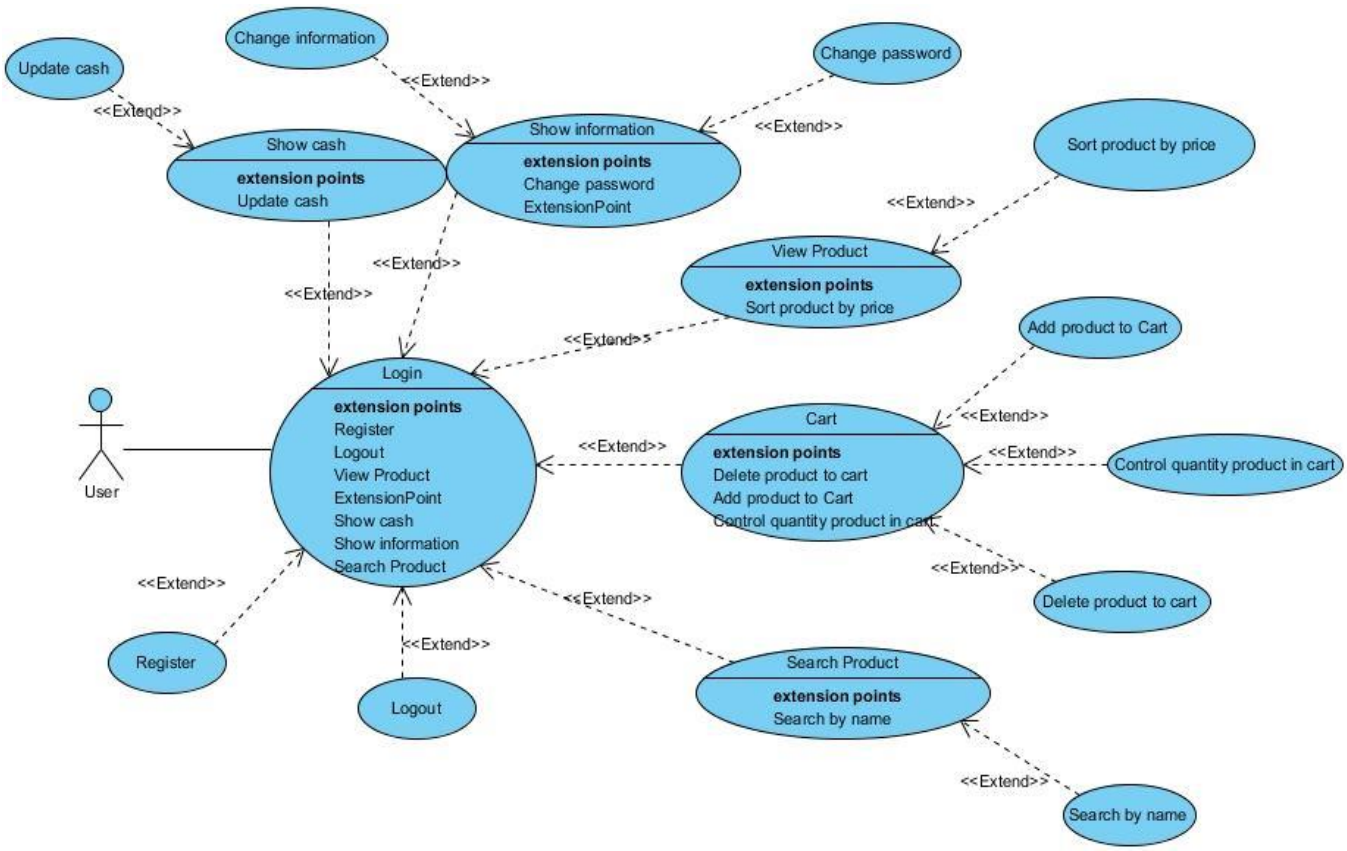
1. Client - Server architecture

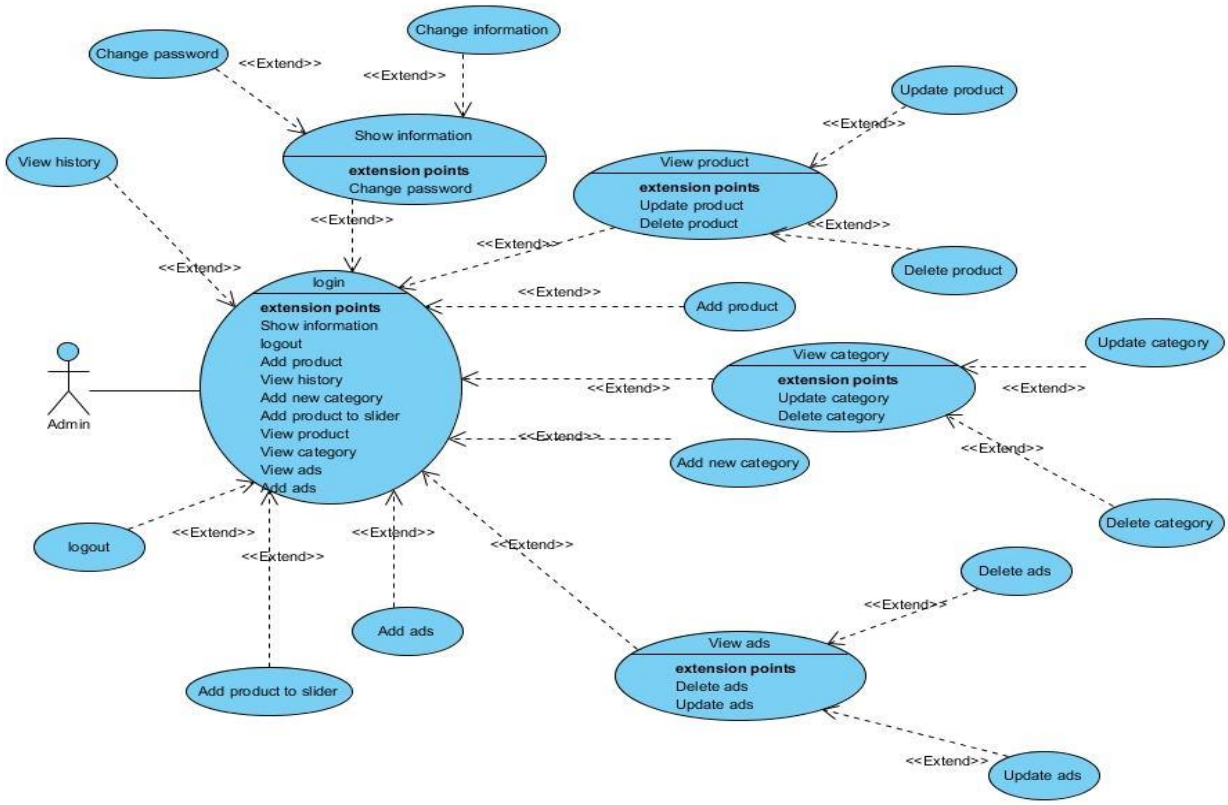
A Client-Server Architecture consists of two types of components: clients and servers. A server component perpetually listens for requests from client components. When a request is received, the server processes the request, and then sends a response back to the client. Servers may be further classified as stateless or stateful. Clients of a stateful server may make composite requests that consist of multiple atomic requests. This enables more conversational or transactional interactions between client and server. To accomplish this, a stateful server keeps a record of the requests from each current client. This record is called a session.

In order to simultaneously process requests from multiple clients, a server often uses the Master-Slave Pattern. In this case, the Master perpetually listens

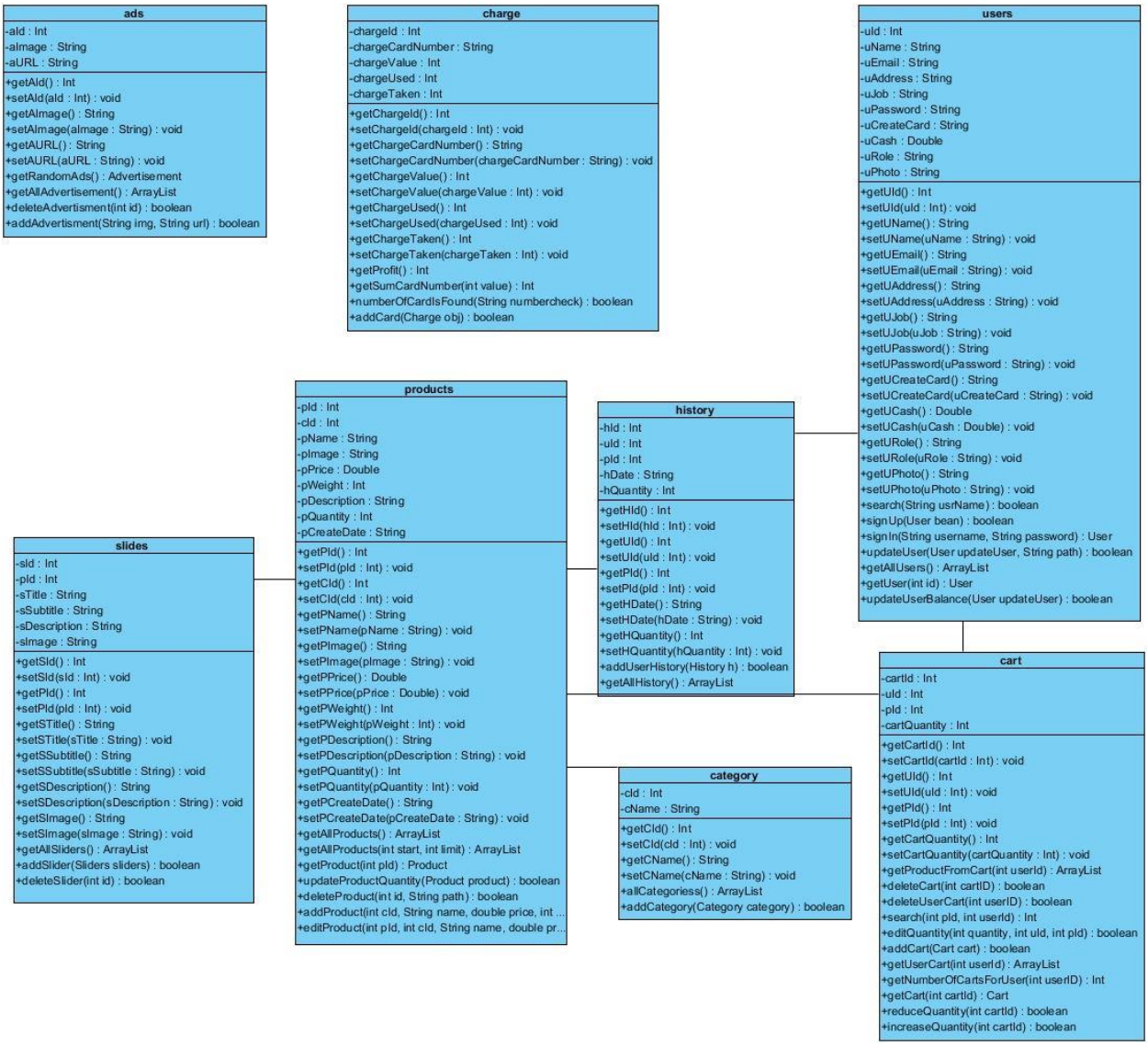
for client requests. When a request is received, the master creates a slave to processes the request and then resumes listening. Meanwhile, the slave performs all subsequent communication with the client.

2. Use-case diagram





3.Class diagram


ads
-aId : Int
-aImage : String
-aURL : String
+getAId() : Int
+setAId(aId : Int) : void
+getAImage() : String
+setAImage(aImage : String) : void
+getAURL() : String
+setAURL(aURL : String) : void
+getRandomAds() : Advertisement
+getAllAdvertisement() : ArrayList
+deleteAdvertisment(int id) : boolean
+addAdvertisment(String img, String url) : boolean
charge
-chargeId : Int
-chargeCardNumber : String
-chargeValue : Int
-chargeUsed : Int
-chargeTaken : Int
+getChargeId() : Int
+setChargeId(chargeId : Int) : void
+getChargeCardNumber() : String
+setChargeCardNumber(chargeCardNumber : String) : void
+getChargeValue() : Int
+setChargeValue(chargeValue : Int) : void
+getChargeUsed() : Int
+setChargeUsed(chargeUsed : Int) : void
+getChargeTaken() : Int
+setChargeTaken(chargeTaken : Int) : void
+getProfit() : Int
+getSumCardNumber(int value) : Int
+numberOfCardIsFound(String numbercheck) : boolean
+addCard(Charge obj) : boolean
users
-uId : Int
-uName : String
-uEmail : String
-uAddress : String
-uJob : String
-uPassword : String
-uCreateCard : String
-uCash : Double
-uRole : String
-uPhoto : String
+getUId() : Int
+setUId(uId : Int) : void
+getUName() : String
+setUName(uName : String) : void
+getUEmail() : String
+setUEmail(uEmail : String) : void
+getUAddress() : String
+setUAddress(uAddress : String) : void
+getUJob() : String
+setUJob(uJob : String) : void
+getUPassword() : String
+setUPassword(uPassword : String) : void
+getUCreateCard() : String
+setUCreateCard(uCreateCard : String) : void
+getUCash() : Double
+setUCash(uCash : Double) : void
+getURole() : String
+setURole(uRole : String) : void
+getUPhoto() : String
+setUPhoto(uPhoto : String) : void
+search(String usrName) : boolean
+signUp(User bean) : boolean
+signIn(String username, String password) : User
+updateUser(User updateUser, String path) : boolean
+getAllUsers() : ArrayList
+getUser(int id) : User
+updateUserBalance(User updateUser) : boolean
products
-pId : Int
-cId : Int
-pName : String
-pImage : String
-pPrice : Double
-pWeight : Int
-pDescription : String
-pQuantity : Int
-pCreateDate : String
+getPId() : Int
+setPId(pId : Int) : void
+getCId() : Int
+setCId(cId : Int) : void
+getPName() : String
+setPName(pName : String) : void
+getPImage() : String
+setPImage(pImage : String) : void
+getPPrice() : Double
+setPPrice(pPrice : Double) : void
+getPWeight() : Int
+setPWeight(pWeight : Int) : void
+getPDescription() : String
+setPDescription(pDescription : String) : void
+getPQuantity() : Int
+setPQuantity(pQuantity : Int) : void
+getPCreateDate() : String
+setPCreateDate(pCreateDate : String) : void
+getAllProducts() : ArrayList
+getAllProducts(int start, int limit) : ArrayList
+getProduct(int pId) : Product
+updateProductQuantity(Product product) : boolean
+deleteProduct(int id, String path) : boolean
+addProduct(int cId, String name, double price, int ...
+editProduct(int pId, int cId, String name, double pr...
history
-hId : Int
-uId : Int
-pId : Int
-hDate : String
-hQuantity : Int
+getHId() : Int
+setHId(hId : Int) : void
+getUId() : Int
+setUId(uId : Int) : void
+getPId() : Int
+setPId(pId : Int) : void
+getHDate() : String
+setHDate(hDate : String) : void
+getHQuantity() : Int
+setHQuantity(hQuantity : Int) : void
+addUserHistory(History h) : boolean
+getAllHistory() : ArrayList
slides
-sId : Int
-pId : Int
-sTitle : String
-sSubtitle : String
-sDescription : String
-sImage : String
+getSId() : Int
+setSId(sId : Int) : void
+getPId() : Int
+setPId(pId : Int) : void
+getSTitle() : String
+setSTitle(sTitle : String) : void
+getSSubtitle() : String
+setSSubtitle(sSubtitle : String) : void
+getSDescription() : String
+setSDescription(sDescription : String) : void
+getSImage() : String
+setSImage(sImage : String) : void
+getAllSliders() : ArrayList
+addSlider(Sliders sliders) : boolean
+deleteSlider(int id) : boolean
category
-cId : Int
-cName : String
+getCId() : Int
+setCId(cId : Int) : void
+getCName() : String
+setCName(cName : String) : void
+allCategoriess() : ArrayList
+addCategory(Category category) : boolean
cart
-cartId : Int
-uId : Int
-pId : Int
-cartQuantity : Int
+getCartId() : Int
+setCartId(cartId : Int) : void
+getUId() : Int
+setUId(uId : Int) : void
+getPId() : Int
+setPId(pId : Int) : void
+getCartQuantity() : Int
+setCartQuantity(cartQuantity : Int) : void
+getProductFromCart(int userId) : ArrayList
+deleteCart(int cartID) : boolean
+deleteUserCart(int userID) : boolean
+search(int pId, int userId) : Int
+editQuantity(int quantity, int uId, int pId) : boolean
+addCart(Cart cart) : boolean
+getUserCart(int userId) : ArrayList
+getNumberOfCartsForUser(int userID) : Int
+getCart(int cartId) : Cart
+reduceQuantity(int cartId) : boolean
+increaseQuantity(int cartId) : boolean

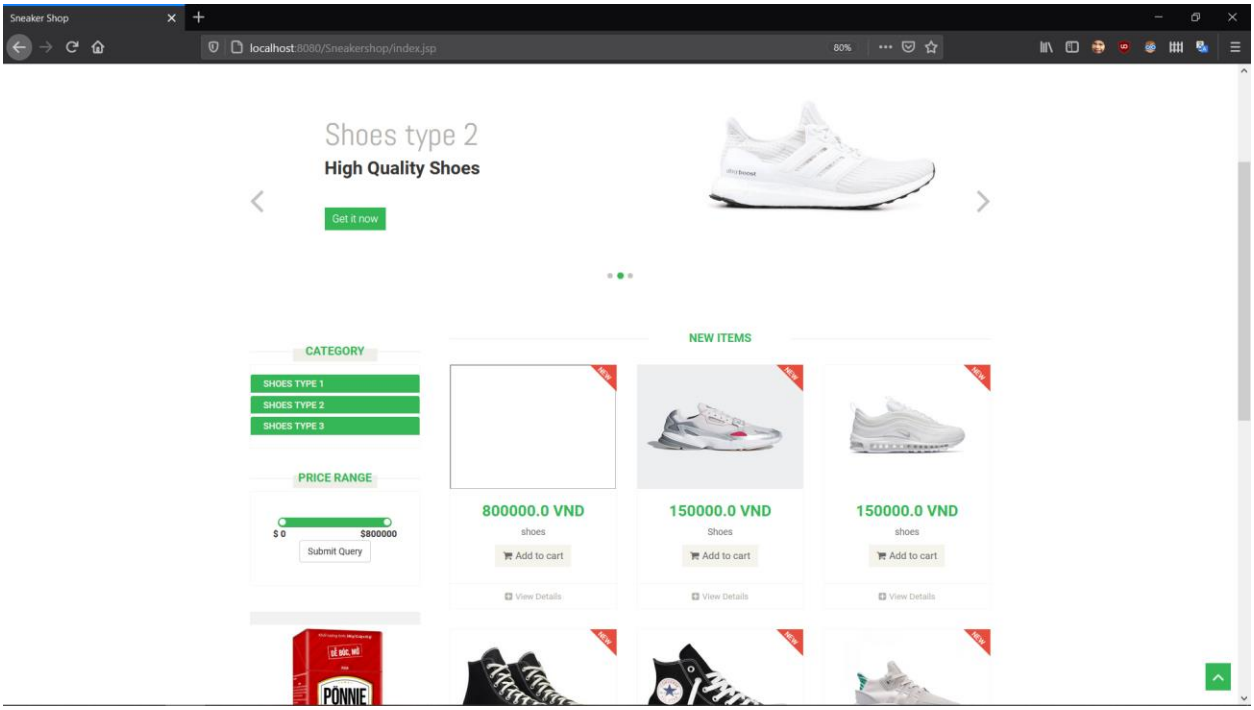
PART V: CONCLUSION DEVELOPMENT DIRECTIONS

This is to conclude that the project that I undertook was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirement which have been remaining, can be completed with a short extension.

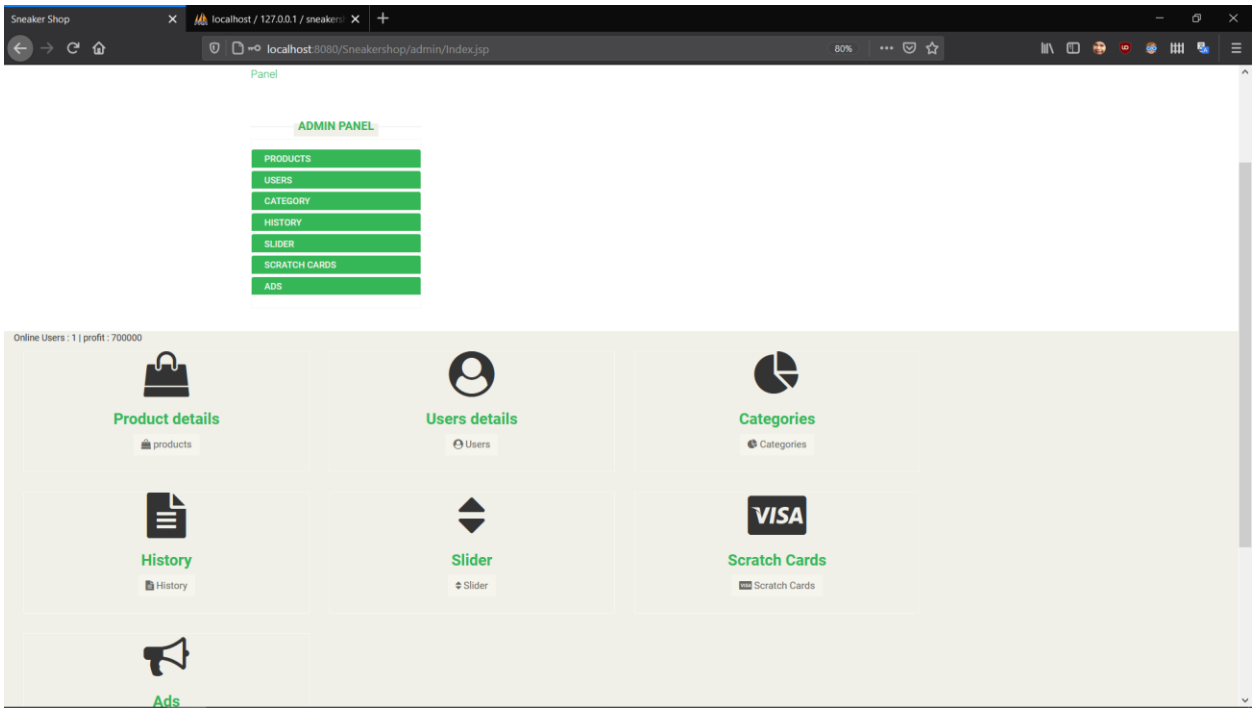
The project made here is just to ensure that this product could be valid in today real challenging world. Here all the facilities are made and tested.

In near future it will be extended for many types of features so that efficiency can be improved.

1. Interface design



2. Admin Page


Sneaker Shop
localhost / 127.0.0.1 / sneaker
localhost:8080/Sneakershop/admin/Index.jsp
80%
Panel
ADMIN PANEL
PRODUCTS
USERS
CATEGORY
HISTORY
SLIDER
SCRATCH CARDS
ADS
Online Users : 1 | profit : 700000
Product details
products
Users details
Users
Categories
Categories
History
History
Slider
Slider
VISA
Scratch Cards
Scratch Cards
Ads