

E-commerce Engine with GUI Based Product Design Platform

Submitted By

S. M Asif Al – Samy
ID: 2013-2-60-002

Md. Lutfullah Hill Kabir
ID: 2013-2-60-019

Monim Hossian
ID: 2013-2-60-050

Supervised By

Md. Shamsujjoha
Senior Lecturer
Department of Computer Science and Engineering
East West University

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of
Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
EAST WEST UNIVERSITY

December 2017

ABSTRACT

Many customers want to buy their own designed product, this project helps those customers to add custom design (add images, text, change color etc.) to the desired product. This project provides the user with a category of different clothing products available and also provides a user interface to design a clothing product according to his demand and order these. There is a facility of **add to cart** to design as well as order multiple products at a time and also an online payment system to make a successful transaction. In addition, features of searching product from other websites and compare those according to price are also available in our project. These all things are done at the user site of the project, there is also an admin site of the project where all the products, its orders, all the customers are control by the admin. Finally In our project a database system is used to handle all the data. This is a fully dynamic system which can be easily operated by the users.

DECLARATION

The project has been submitted to the Department of the Computer Science & Engineering. East West University in the partial fulfillment of the requirement for the degree of Bachelor of Science in CSE performed by us under supervision of **Dr. Md. Shamsujjoha, Senior Lecturer, Department of Computer Science and Engineering** at East West University. We hereby, declare that this project has not been submitted elsewhere for the requirement of any degree or diploma or any other purposes.

Submitted By:

S. M. Asif Al – Samy

ID: 2013-2-60-002

Department of Computer Science and Engineering
East West University

Md. Lutfullah Hill Kabir

ID: 2013-2-60-019

Department of Computer Science and Engineering
East West University

Monim Hossain

ID: 2013-2-60-050

Department of Computer Science and Engineering
East West University

Supervised By:

Md. Shamsujjoha

Senior Lecturer

Department of Computer Science and Engineering
East West University

LETTER OF ACCEPTANCE

This Project entitled “**E-commerce Engine with GUI Based Product Design Platform**” submitted By S. M. Asif Al - Samy, Id: 2013-2-60-002, Monim Hossain, Id:2013-2-60-050, Md. Lutfullah Hill Kabir, Id: 2013-2-60-019 to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh has been accepted as satisfactory for the partial fulfillment of the of the requirement for the degree of Bachelor of Science in Computer Science & Engineering and approved as to its style and contents. The presentation has been held on December 07, 2017.

Supervisor

Chairperson

Md. Shamsujjoha
Senior Lecturer
Dept. of Computer Science & Engineering
East West University, Dhaka, Bangladesh

Dr. Ahmed Wasif Reza
Associate Professor & Chairperson
Dept. of Computer Science & Engineering
East West University, Dhaka, Bangladesh

ACKNOWLEDGEMENT

First, we are thankful and expressing our gratefulness to Almighty who offers us divine blessings, patience, mental and physical strength to complete this project.

Working on this project **E-commerce Engine with GUI Based Product Design Platform** has been immense knowledge to us. We would like to express our sincere gratitude to our honorable supervisor Md. Shamsujjoha, Senior Lecturer, Dept. of Computer Science & Engineering at East West University for his guidance and valuable support for this project. We acknowledge with a deep sense of gratitude, the encouragement and inspiration from our faculty members and colleagues.

Finally, we would like to convey our Special thanks to our parents whom have always given us tremendous Support. Without their love and encouragement, we would not had achieved this far.

TABLE OF CONTENTS

Title	Page no
Abstract	i
Declaration	ii
Letter of Acceptance	iii
Acknowledgement	iv
Table of Contents	v-vi
Chapter 01 Introduction	01-03
1.1 Motivation	01
1.2 Objectives	02
1.3 Scope	02-03
Chapter 02 Analysis and Designs	04-09
2.1 Requirement Analysis	04
2.2 Functional Requirement	04-05
2.3 Non-Functional Requirements	05
2.4 Class Diagram	06
2.5 Use Case Diagram	07
2.6 Sequence Diagram	08-09
Chapter 03 Database Design	10-15
3.1 E-R Diagram	10
3.2 Database Table	11
3.3 Admin Table	12
3.4 Customer Table	13
3.5 Product Table	13
3.6 Category Table	14
3.7 Contact Table	14
3.8 Cart Table	15
3.9 Customer Order Table	15

Chapter 04 Application Functionalities	16-21
4.1 Product Design Platform	16
4.2 User Registration and Verification	16-17
4.3 Cart Functionalities	17
4.4 Checkout & Payments	17-18
4.5 Order Approval & Cancellation	18
4.6 Other Functionalities	19
4.7 Search and Compare Functionalities	19-20
4.8 Updating Automatically the Database	20
4.9 Digital Marketing	20-21
Chapter 05 Implementation Plan	22
5.1 Project co-ordination	22
5.2 Project Merging	22
Chapter 06 Security and System	23-25
6.1 Security for User	23
6.2 Security for Admin	23
6.3 Secured with Session Cookie	23
6.4 Security of Database	24-25
Chapter 07 Conclusion and Future Scope	26
Reference	27
Appendix	

Chapter 1

Introduction

E-commerce stand for Electronic commerce is platform where buying and selling of product is done over electronic system such as internet or any other computer network.

To develop an **E-commerce Engine with Product Design Feature** which will provide buying and selling of product through the main target of this project is to create a platform to design a product efficiently for the users. Moreover, the system has been developed in such a way which makes it reliable, secure and user friendly. This system keeps a list of Stocks of pre-designed products so that customer can purchase it. Another major operation is to search and compare products among different websites using this website.

For product design feature we have used **Fabric.js** a powerful and simple **JavaScript HTML5** canvas library.

1.1 Motivation

We found some motives which encourage doing this project eagerly.

- i. Online retail is expanding at an astounding pace. In the past few years, global ecommerce sales have eclipsed \$1 trillion, and it is well on its way to reaching \$2 trillion per year. Doing business online has become part of daily life for many of us- we're now living in a world where people order their groceries online and have them delivered right to their doorstep.
- ii. Nowadays people are using any product which is customized as their own choices and tests.
- iii. **Fabric.js** is a powerful and simple **JavaScript HTML5** canvas library provides interactive object model on top of canvas element. Fabric also has SVG-to-canvas (and canvas-to-SVG) parser and many other features (will be found at <http://fabricjs.com/>).

1.2 Objectives

- General Objectives:
 - i. To develop our country's first ecommerce site where an user can design a product, order and get that product from this site.
 - ii. To increase efficiency and services to customers through better application of technology in daily operation.
 - iii. To able to stand out from competition in the ecommerce sites.

- Specific Objectives:
 - i. To enable customer to view, search existing product and also design new product without any authentication.
 - ii. To make a comparison among products of different websites and show those compared product information.
 - iii. To purchase products (pre-designed as well as post-designed by customer) by making online payment.

1.3 Scope

This system provides the services from the admin to the customer with all necessary steps.

Users (Not Registered):

- View existing/pre-designed products
- Design a new product
- Add to cart
- Update and Delete product from cart
- Search & Compare Products from different sites
- Contact Staff (send message)

Users (Registered):

- View existing/pre-designed products
- Design a new product
- Add to cart
- Update & Delete product from cart
- Search & Compare Products from different sites
- Checkout by online payment

- View order status (Pending/Approved/Cancelled)
- View & Update Profile
- Contact Staff (send message)
- Upload designed product

Administrator:

- Create/update products & category
- Check Order and Take action on Order (Approve/Cancel)
- View Sales details according to date
- View Messages

Chapter 2

Analysis and Designs

2.1 Requirement Analysis

Requirement Analysis also called Requirement Engineering is the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed. That means it's a process of determining the requirements for developing a software or system. For our system we found functional and non-functional requirements.

2.1.1 Functional Requirement

- Users must be able to view the details of existing/pre-designed products.
- Users must be able to compare cost by searching products from different websites.
- Users must be able to design (change color, add text, add image) a new product.
- Users must be able to add any product to cart.
- Users must be able to update or delete any product from his cart.
- Users must be able to contact with admin.
- Users must be log in to the system for purchasing a product.
- On registration, users must verify their email.
- The system must be set online payment (using credit/debit card) system for purchasing any product.
- Users should be able to upload their designed product to this website.
- Users should be able to view order status (Pending/Approves/Cancelled).
- Registered users must be able to update their profile.
- Registered Users should be able to give feedback for a product.
- Admin must be able check all the orders
- Admin must be able to take action over the orders (Approve/Cancel Orders)
- Admin should be able to check the messages received from the users.
- Admin must be able to add, update, and delete product & category.
- Admin should be able to generate a receipt for ordered product for delivery purpose.

- Admin must be able to view the entire sale information (Income Report).

2.1.2 Non-functional Requirements

- Email Verification
- Server-side validation
- Block a user after 3 unsuccessful login attempt
- Ease-of-use
- Payment validation
- Redundancy free

2.2 Class Diagram

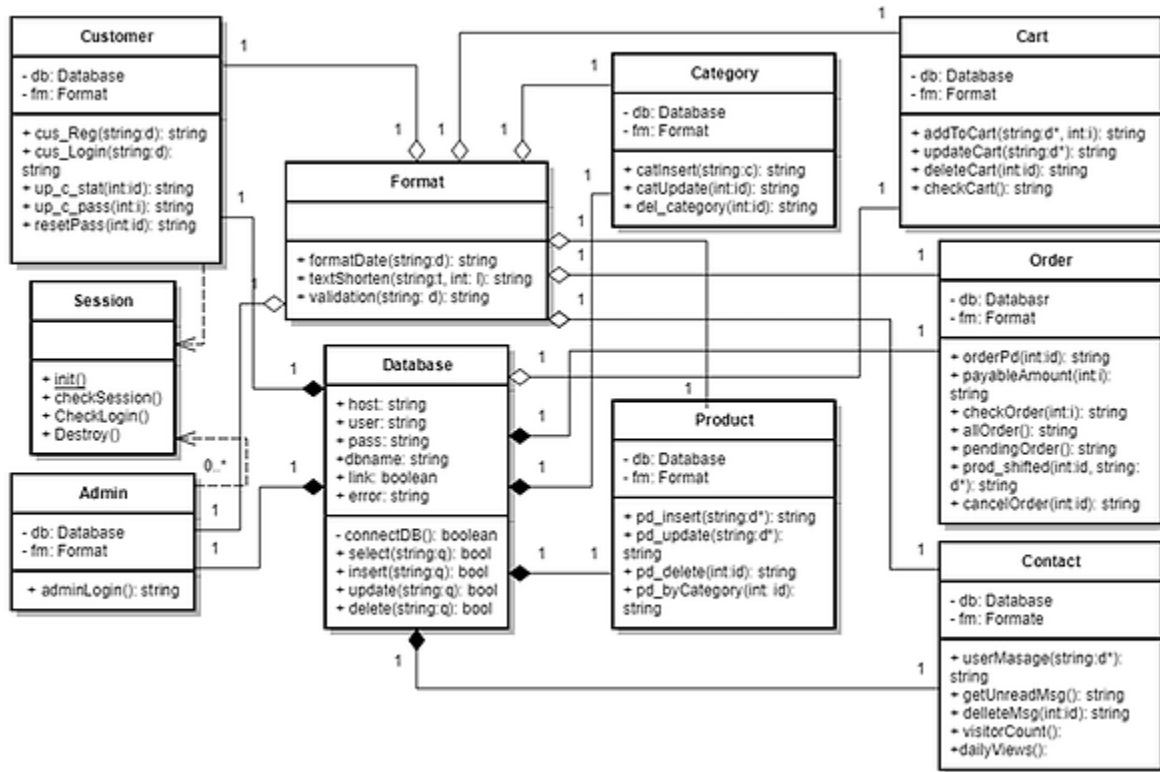


Figure 2.1: Class Diagram of the system.

2.3 Use Case Diagram

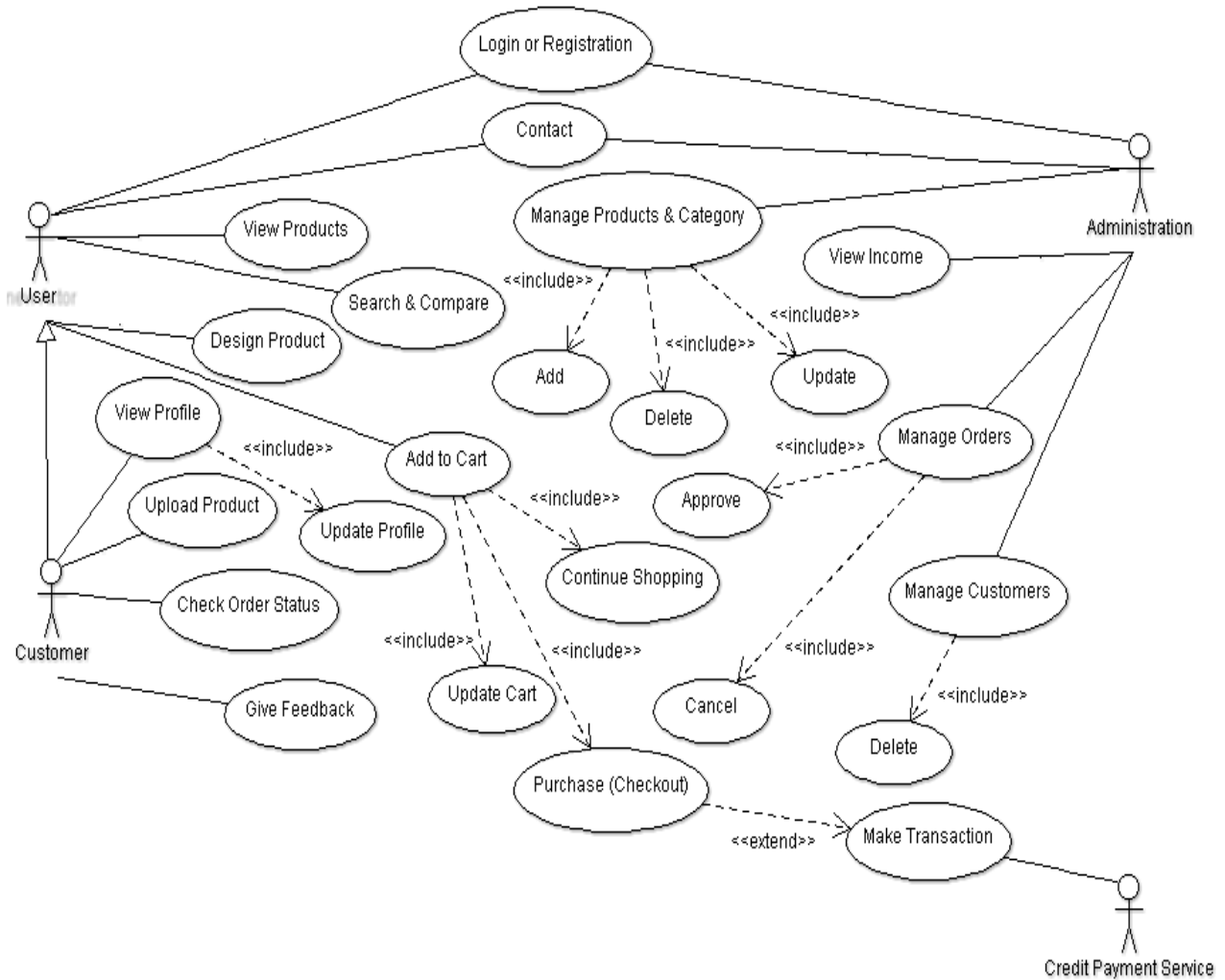


Figure 2.2: Use Case Diagram of the System

2.4 Sequence Diagram

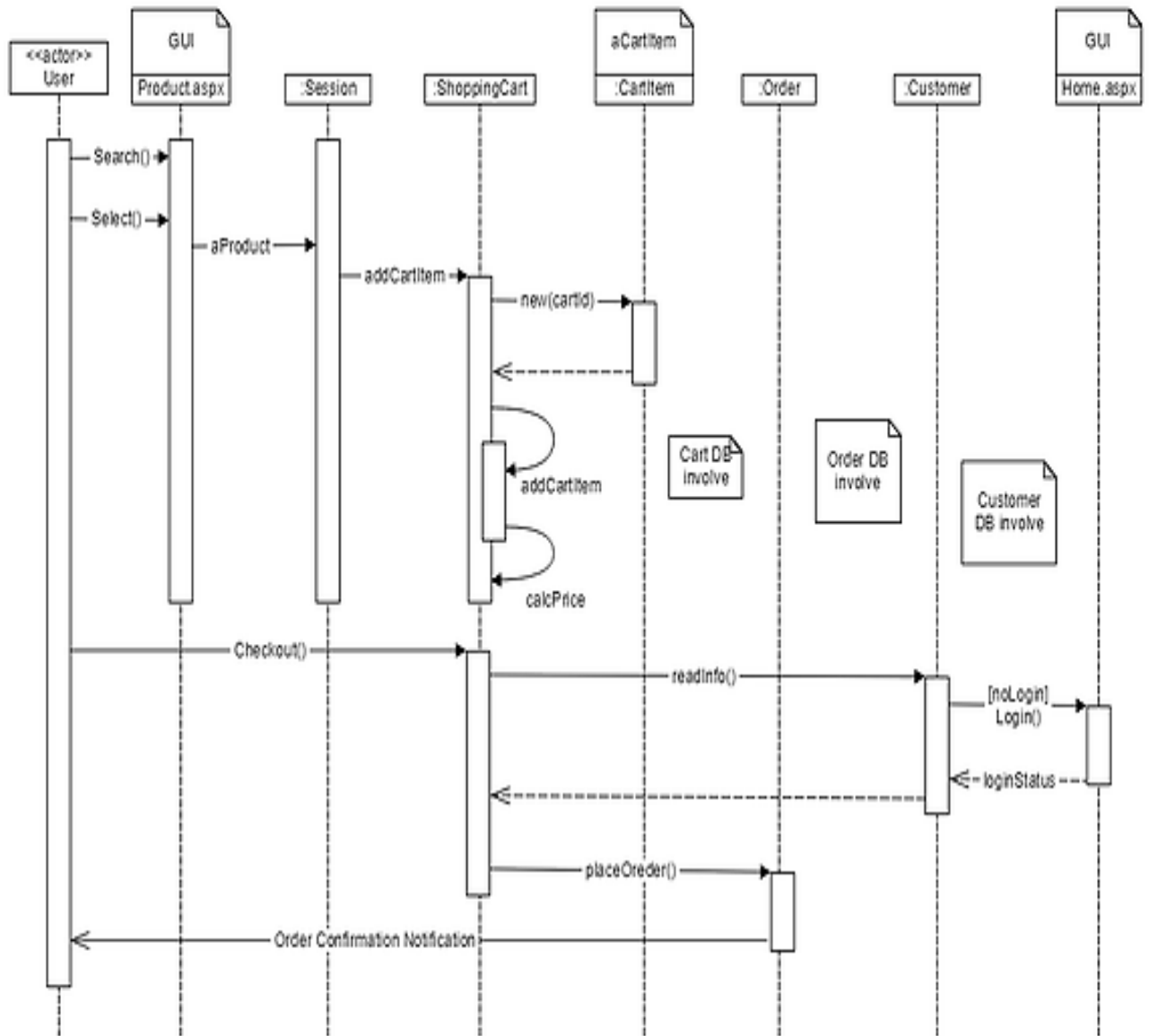


Figure 2.3: Sequence Diagram for User.

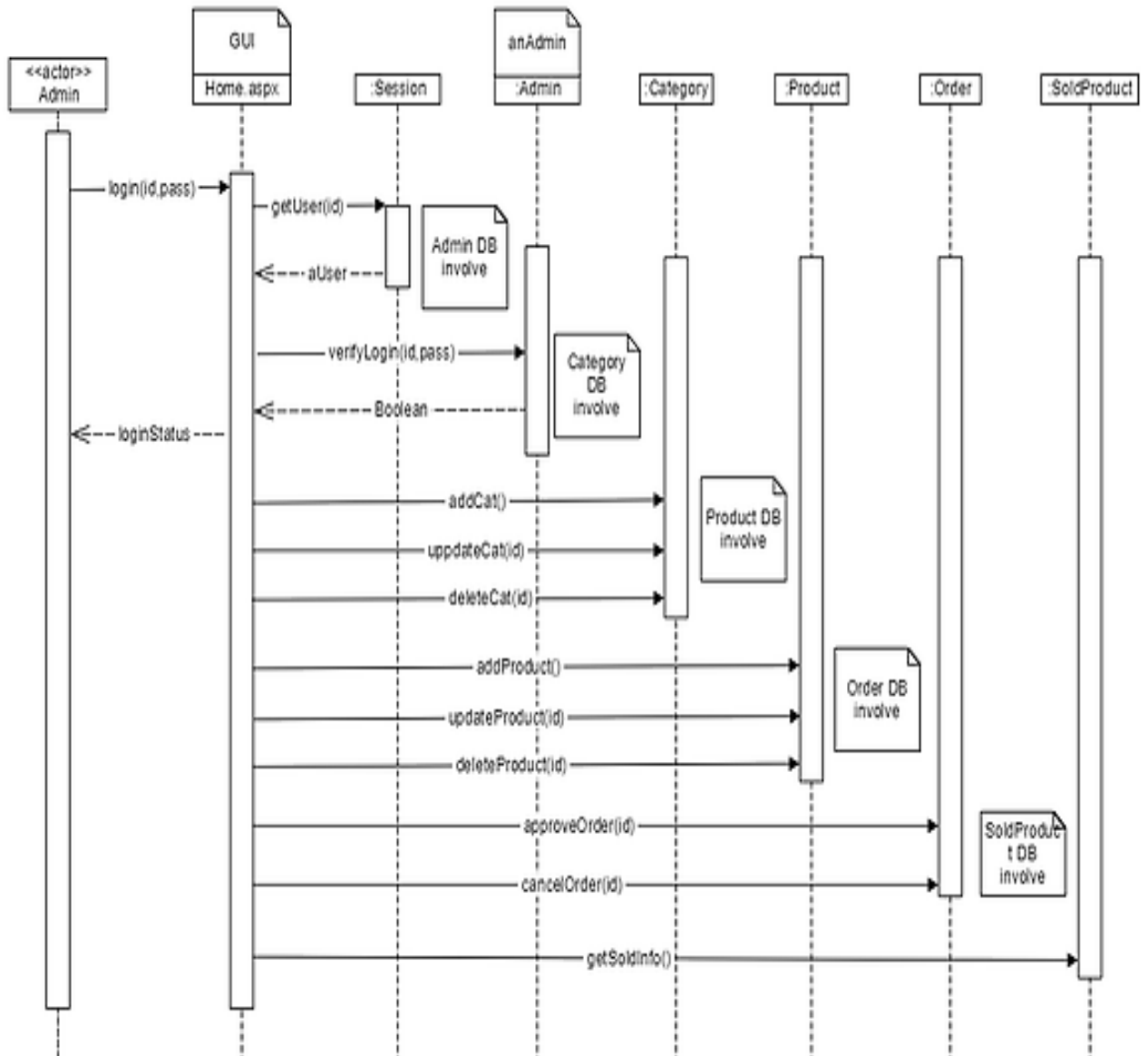


Figure 2.4: Sequence Diagram for Administration.

Chapter 3

Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

3.1 E-R Diagram

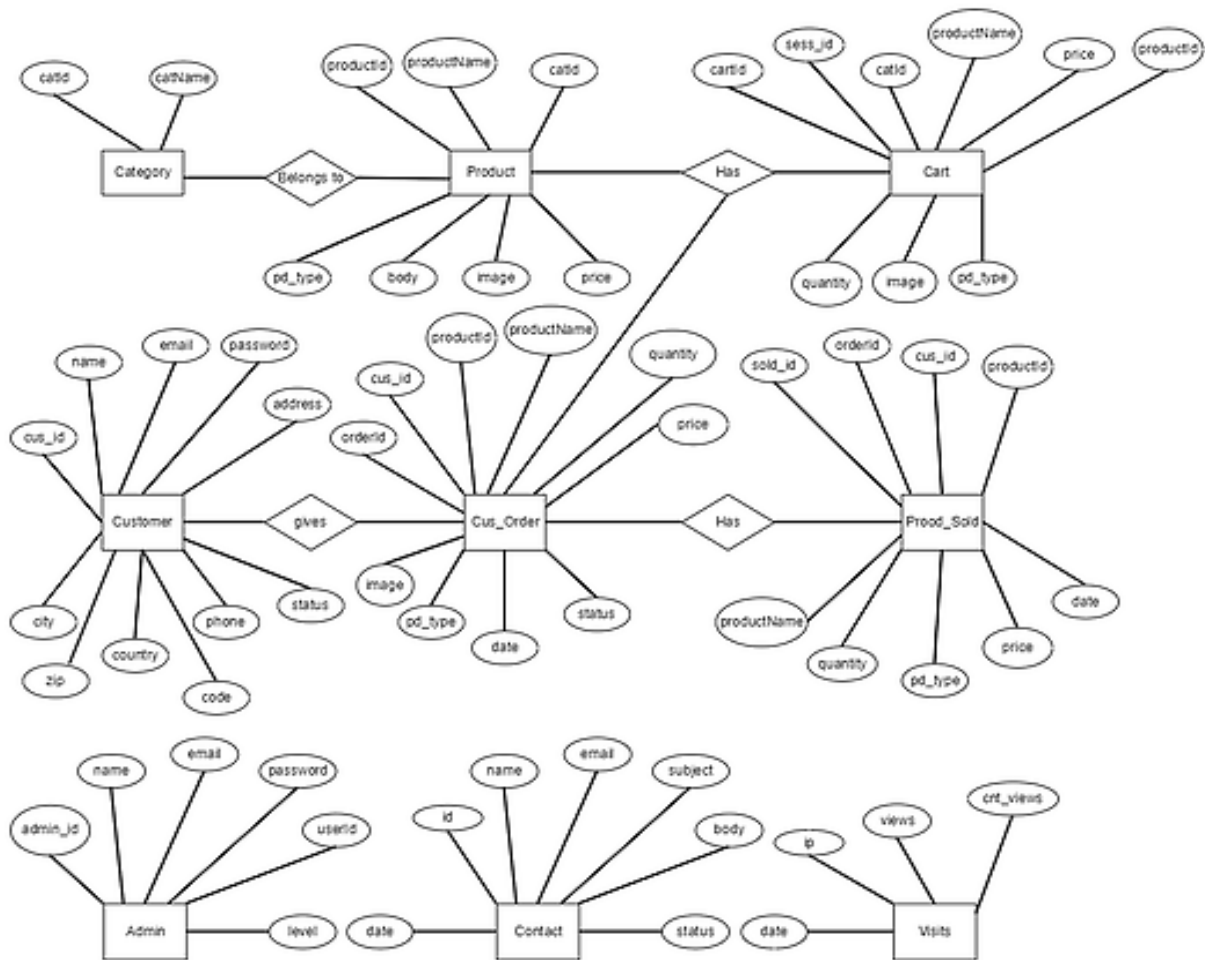


Figure 3.1: E-R Diagram of the database.

3.2 Database Table

Server: 127.0.0.1 » Database: design_club

Structure SQL Search Query Export Import Operations Privileges Routines

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
cart	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
category	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
contact	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
customer	★ Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
cus_order	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
pd_sold	★ Browse Structure Search Insert Empty Drop	33	InnoDB	latin1_swedish_ci	16 KiB	-
product	★ Browse Structure Search Insert Empty Drop	24	InnoDB	latin1_swedish_ci	16 KiB	-
visits	★ Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	32 KiB	-
9 tables	Sum	80	InnoDB	latin1_swedish_ci	160 KiB	0 B

Figure 3.2: Entire database tables.

3.3 Admin Table

Server: 127.0.0.1 » Database: design_club » Table: admin

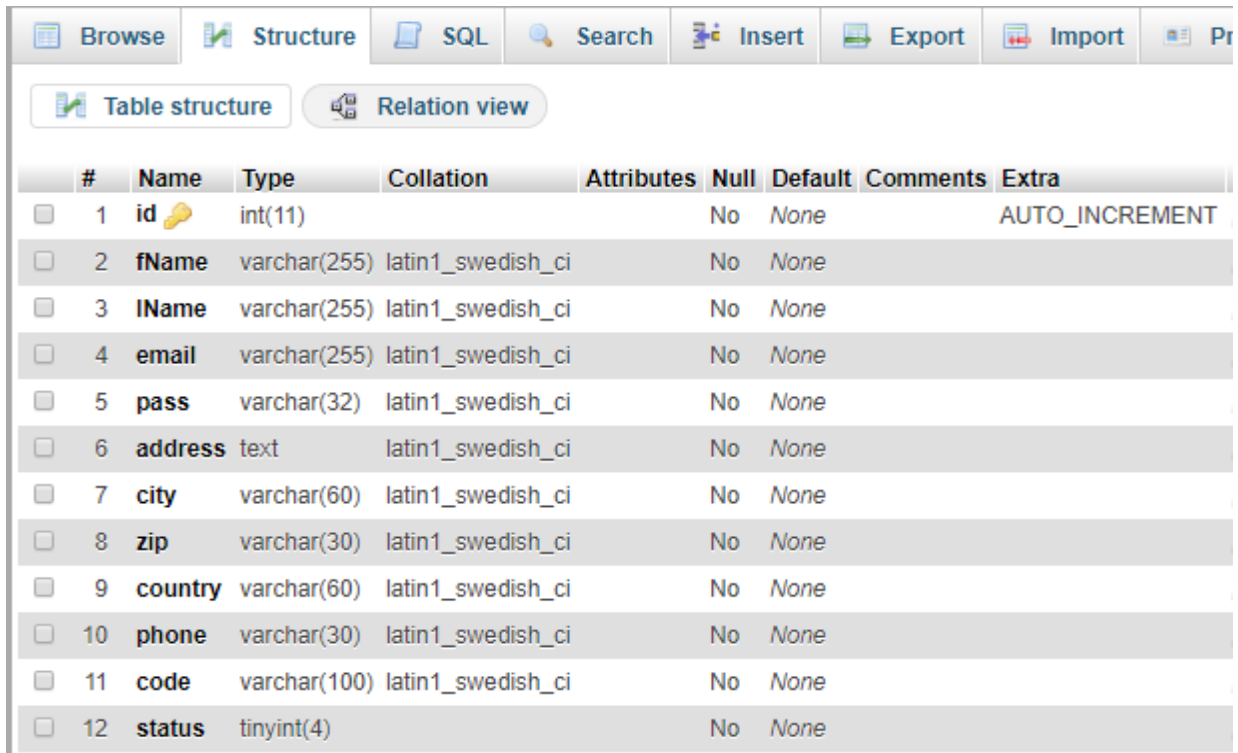
Browse Structure SQL Search Insert Export Import Priv

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	adminId	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	adminName	varchar(255)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 3	adminUser	varchar(255)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 4	adminEmail	varchar(255)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 5	adminPass	varchar(32)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 6	level	tinyint(4)			No	None		

Figure 3.3: Admin table of the database.

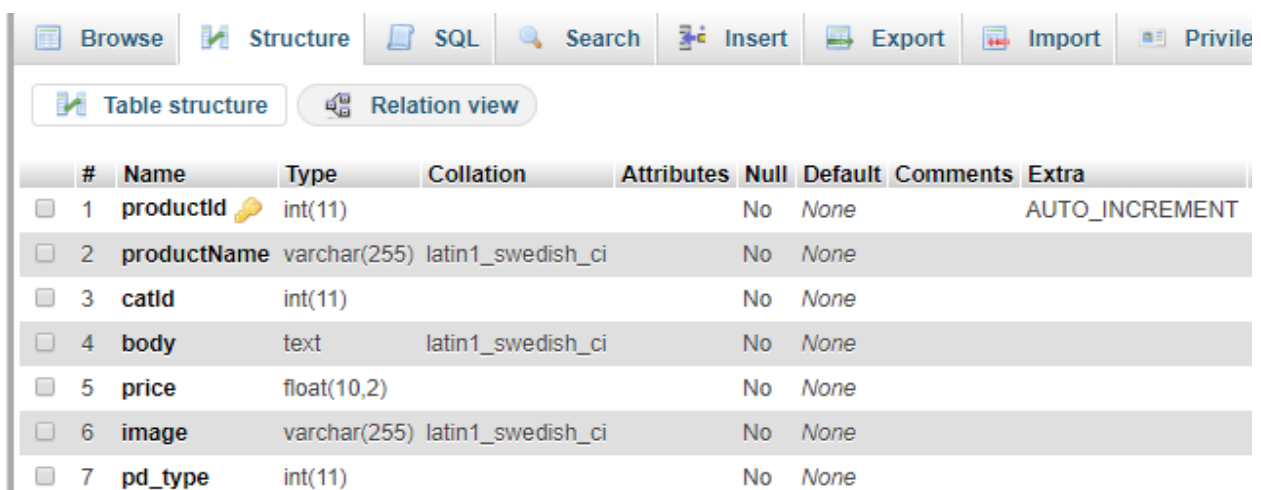
3.4 Customer Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	fName	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	3	lName	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	4	email	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	5	pass	varchar(32) latin1_swedish_ci		No	None		
<input type="checkbox"/>	6	address	text latin1_swedish_ci		No	None		
<input type="checkbox"/>	7	city	varchar(60) latin1_swedish_ci		No	None		
<input type="checkbox"/>	8	zip	varchar(30) latin1_swedish_ci		No	None		
<input type="checkbox"/>	9	country	varchar(60) latin1_swedish_ci		No	None		
<input type="checkbox"/>	10	phone	varchar(30) latin1_swedish_ci		No	None		
<input type="checkbox"/>	11	code	varchar(100) latin1_swedish_ci		No	None		
<input type="checkbox"/>	12	status	tinyint(4)		No	None		

Figure 3.4: Customer table of the database.

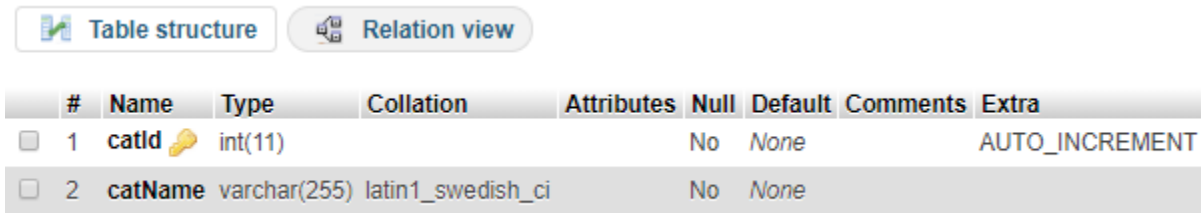
3.5 Product Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	productId	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	productName	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	3	catId	int(11)		No	None		
<input type="checkbox"/>	4	body	text latin1_swedish_ci		No	None		
<input type="checkbox"/>	5	price	float(10,2)		No	None		
<input type="checkbox"/>	6	image	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	7	pd_type	int(11)		No	None		

Figure 3.5: Product table of the database.

3.6 Category Table

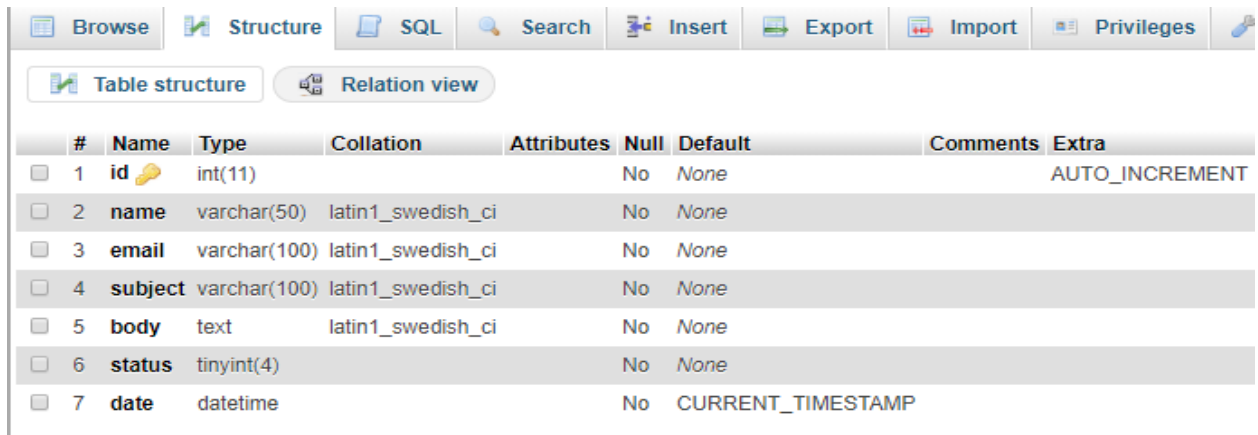


The screenshot shows a database interface with two tabs: 'Table structure' (selected) and 'Relation view'. Below the tabs is a table with the following columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, and Extra. The table contains two rows of data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	catid	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	catName	varchar(255)	latin1_swedish_ci		No	None		

Figure 3.6: Category table of the database.

3.7 Contact Table

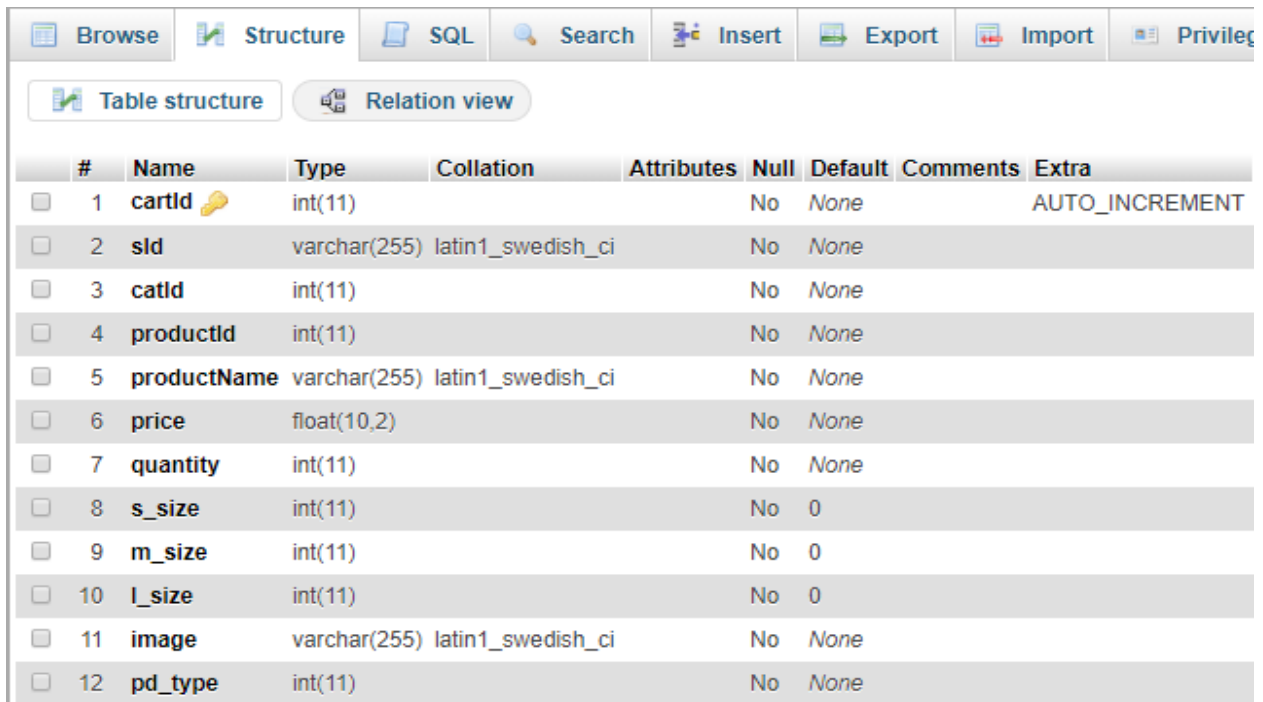


The screenshot shows a database interface with a menu bar containing 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', and 'Privileges'. Below the menu bar are two tabs: 'Table structure' (selected) and 'Relation view'. Below the tabs is a table with the following columns: #, Name, Type, Collation, Attributes, Null, Default, Comments, and Extra. The table contains seven rows of data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(50)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 3	email	varchar(100)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 4	subject	varchar(100)	latin1_swedish_ci		No	None		
<input type="checkbox"/> 5	body	text	latin1_swedish_ci		No	None		
<input type="checkbox"/> 6	status	tinyint(4)			No	None		
<input type="checkbox"/> 7	date	datetime			No	CURRENT_TIMESTAMP		

Figure 3.7: Contact table of the database.

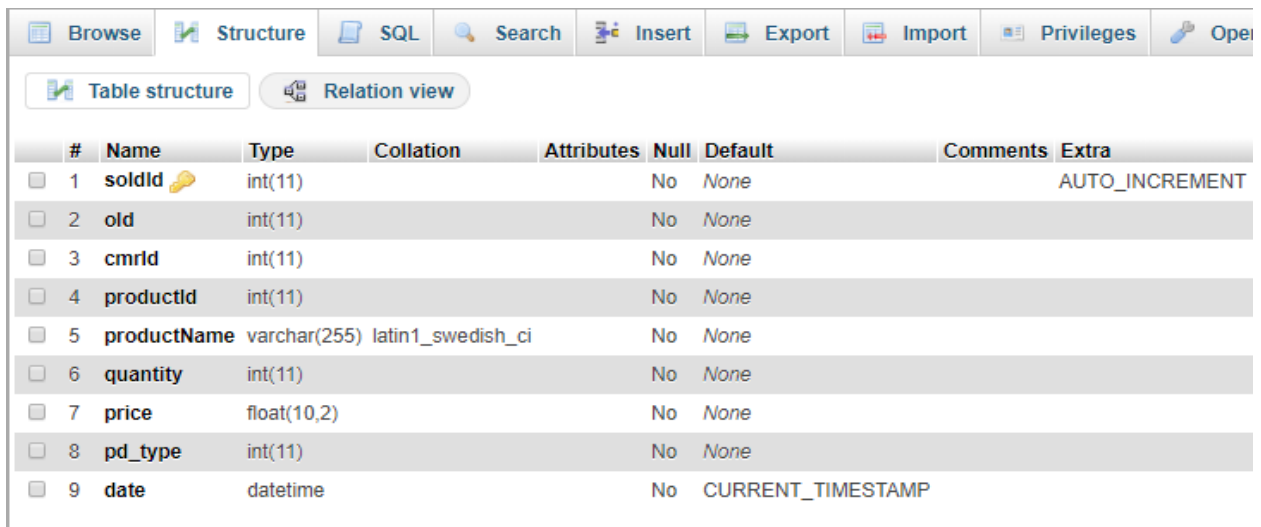
3.8 Cart Table:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	cartId			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	sId	latin1_swedish_ci		No	None		
<input type="checkbox"/>	3	catId			No	None		
<input type="checkbox"/>	4	productId			No	None		
<input type="checkbox"/>	5	productName	latin1_swedish_ci		No	None		
<input type="checkbox"/>	6	price			No	None		
<input type="checkbox"/>	7	quantity			No	None		
<input type="checkbox"/>	8	s_size			No	0		
<input type="checkbox"/>	9	m_size			No	0		
<input type="checkbox"/>	10	l_size			No	0		
<input type="checkbox"/>	11	image	latin1_swedish_ci		No	None		
<input type="checkbox"/>	12	pd_type			No	None		

Figure 3.8: Cart table of the database.

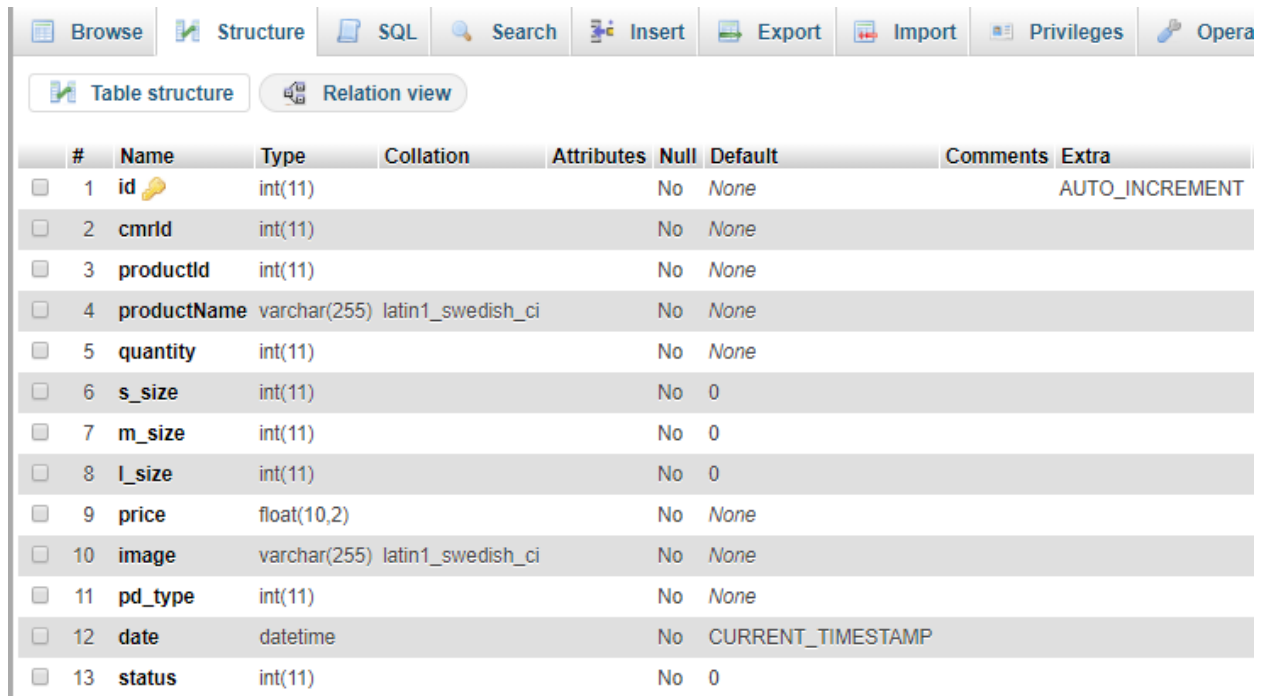
3.9 Customer Order Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	soldId			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	old			No	None		
<input type="checkbox"/>	3	cmrId			No	None		
<input type="checkbox"/>	4	productId			No	None		
<input type="checkbox"/>	5	productName	latin1_swedish_ci		No	None		
<input type="checkbox"/>	6	quantity			No	None		
<input type="checkbox"/>	7	price			No	None		
<input type="checkbox"/>	8	pd_type			No	None		
<input type="checkbox"/>	9	date			No	CURRENT_TIMESTAMP		

Figure 3.9: Customer table of the database.

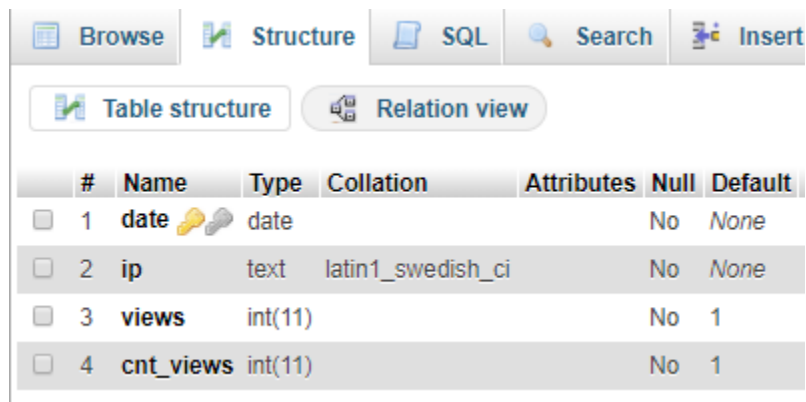
3.10 Product Sold Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	cmrId			No	None		
<input type="checkbox"/>	3	productId			No	None		
<input type="checkbox"/>	4	productName	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	5	quantity			No	None		
<input type="checkbox"/>	6	s_size			No	0		
<input type="checkbox"/>	7	m_size			No	0		
<input type="checkbox"/>	8	l_size			No	0		
<input type="checkbox"/>	9	price	float(10,2)		No	None		
<input type="checkbox"/>	10	image	varchar(255) latin1_swedish_ci		No	None		
<input type="checkbox"/>	11	pd_type			No	None		
<input type="checkbox"/>	12	date			No	CURRENT_TIMESTAMP		
<input type="checkbox"/>	13	status			No	0		

Figure 3.10: Product table of the database.

3.11 Visit Table



#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	date			No	None
<input type="checkbox"/>	2	ip	text latin1_swedish_ci		No	None
<input type="checkbox"/>	3	views	int(11)		No	1
<input type="checkbox"/>	4	cnt_views	int(11)		No	1

Figure 3.11: Visit table of the database.

Chapter 4

Application Functionalities

This application offers miscellaneous tasks for different users. From designing or searching to purchasing product, from approve orders to sell products- a huge number of tasks are performed by users, customers and administrator. In this section we have mentioned the primary functionalities of our web application.

4.1 Product Design Platform

In our web application when a user goes to product design page he finds an UI where he can color, add text and also add image on the product. After that he can add the product into the cart and by paying online he can order it. And when the admin approve his order he will wait for the delivery.

Customize Half-Sleeve T-Shirt



Figure 4.1: UI of Product design platform.



Figure 4.2: UI of add to cart of product.

4.2 User Registration and Verification

For purchasing a product, editing profile a user must be logged in to the system. He also must be registered to log into the system. While registering a confirmation e-mail is sent to the user-given e-mail

address with a randomly generated 40character unique confirmation code and by confirming it a user can log in successfully.

Limiting too many login attempts at once and hence negating the chance of getting hacked by Any kind of brute-force attacks is a primary security feature an application should have. So after 3 unsuccessful login a user will be blocked and for farther log in he must reset his password through email.

4.3 Cart Functionalities

After adding to cart it will take to the cart page and there a customer can update the amount of the product, continue shopping as previous product is in the cart, delete the product from cart and also checkout for purchasing the product.

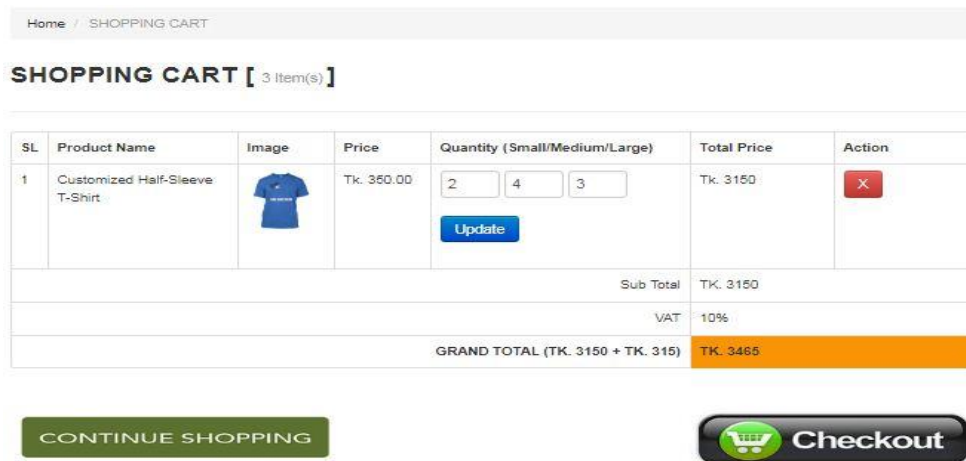


Figure 4.3: UI of Cart Page.

4.4Checkout & Payments

Customers can checkout only using valid payment cards. The validity and authentication is managed By **Stripe**- a well renowned, accredited API for handling payment procedures globally. During payment customer can check their profile to edit their address if any changes occurred.

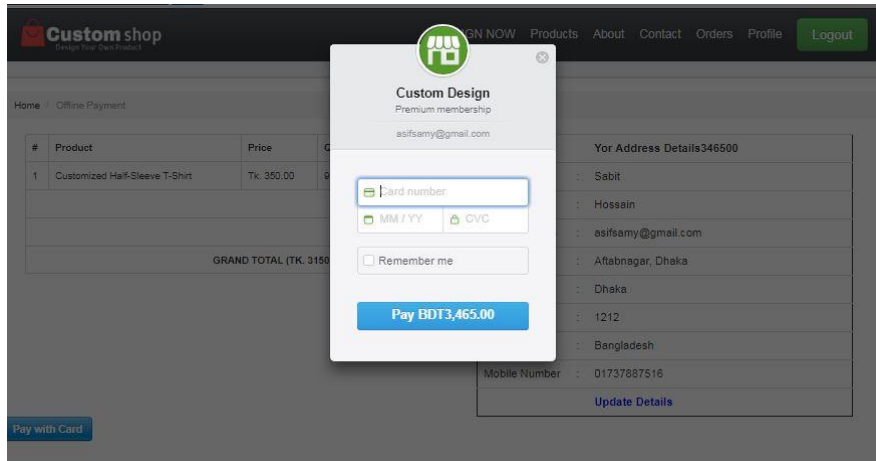


Figure 4.4: UI of Online Payment with Stripe.

4.5 Order Approval & Cancellation

Customers can check order page to check whether his order is approved or not. On the other side, admin must get orders and also have the authority to approve or cancel the order.

Home / Order Details

Your Ordered Details



#	Product Name	Image	Quantity (S, M, L)	Price	Date	Status
1	Customized Hoody		6 (2, 2, 2)	Tk. 550.00	November 27, 2017, 12:41 am	Approved
2	Customized Half-Sleeve T-Shirt		7 (3, 2, 2)	Tk. 350.00	November 26, 2017, 4:45 pm	Pending

Figure 4.5: UI of Order details page for Customer.

OID	Order Time	P_ID	Product Name	Quantity (S, M, L)	Price	Cust.ID	Address	Action
18	November 27, 2017, 12:45 am	76	Customized Hoody	6 (2, 2, 2)	TK. 550.00	11	View Details	Approve / Cancel
17	November 27, 2017, 12:41 am	76	Customized Hoody	6 (2, 2, 2)	TK. 550.00	12	View Details	Remove Print
16	November 27, 2017, 12:37 am	45	Customized Half-Sleeve T-Shirt	6 (2, 2, 2)	TK. 350.00	14	View Details	Approve / Cancel
13	November 26, 2017, 10:32 pm	73	Customized Full-Sleeve T-Shirt	8 (2, 2, 4)	TK. 450.00	14	View Details	Remove Print
12	November 26, 2017, 4:45 pm	71	Customized Half-Sleeve T-Shirt	7 (3, 2, 2)	TK. 350.00	12	View Details	Approve / Cancel

Figure 4.6: UI of Order details page for Administrator.

4.6 Other Functionalities

Without these there are also many functionalities for users such as edit profile, view all products according to category, make contact with admin and for admin such as product and category management, view and delete customer, generate income report according to date, check mailbox to reply customer's query, generate receipt etc.

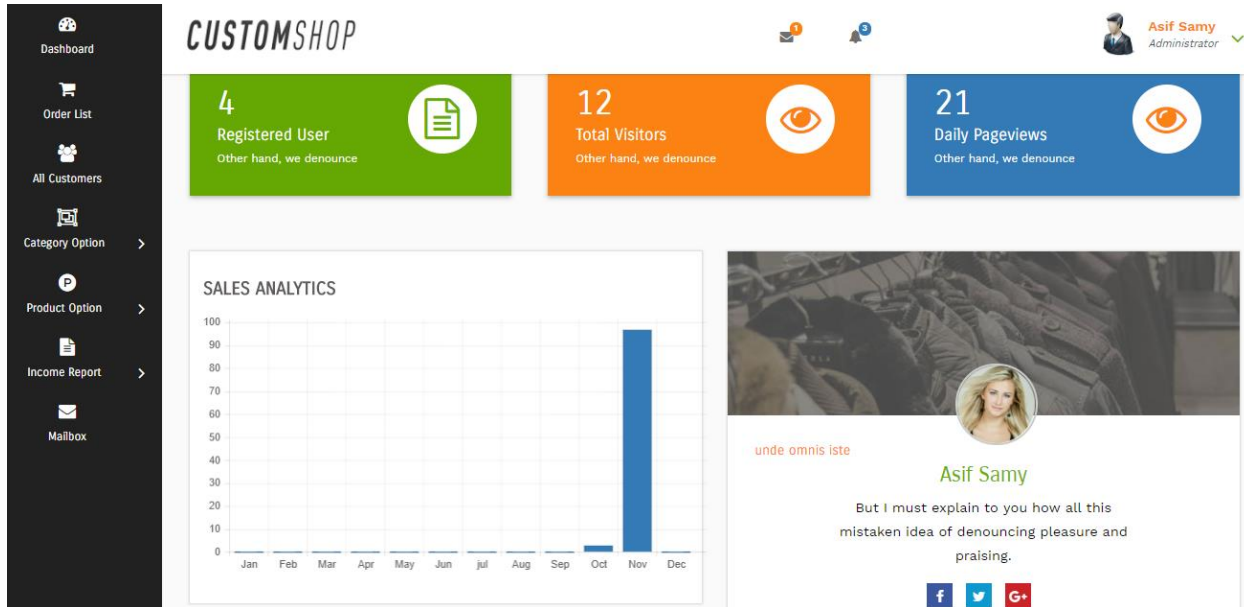


Figure 4.7: UI of Administrator's dashboard.

4.7 Search and Compare Functionalities

One of the functionality is to comparing websites. We focus the price of the various products with similar category and show the price in ascending order. Here we only compare three websites including ours. We have crawl the data from two websites and store it to our own database. We can only focus on the product price. We create a lot of scope here for further studies or plan. We use web crawler data not any kind of API for this job. The focus is to save time for consumer to determine the desired product in a single website. Which is a feature and we will work on this in future. The number is 3 but it can be 50 in near future. Actually, we want to introduce affiliate marketing through our feature. Here is some of the screenshot of this comparing feature. We call it comparison feature. That's why the project is called shop engine which obviously compare the products based on price.



Figure 4.8: Searching for compare.



Figure 4.9: Compared product.

4.8 Updating Automatically the Database

One of our features is auto update the database. Which is important as their prices can be change every now and then so we have the update the price as well. So, we are here to use a function called cron.php which is update automatically the prices of the two websites. Why we say it automatic process? This process will run after every ten minutes in the backend of the Operating System. Every ten minutes a black screen appear and it is just a blink depends on the internet speed and the machine's memory. It is so fast that we can't take any screenshot of the process.

4.9 Digital Marketing

Merging the product into our web application can be a way of affiliate marketing. But we thought, why not going to a next level of marketing. Why not use our web site name to other two e-commerce sites? For making the dream into reality we just think out of the box and it clicked finally. We create an extension which can be used in the web browser. First, we have to load the directory to the and then we enable the E-Commerce Expert which is an extension and then a link will be going to appear in the

particular websites and on this link the address will be our website. It is a unique feature like comparing e-commerce sites product through search engine. Below we name the link as “Go to our comparing website” in here the link will be our website.

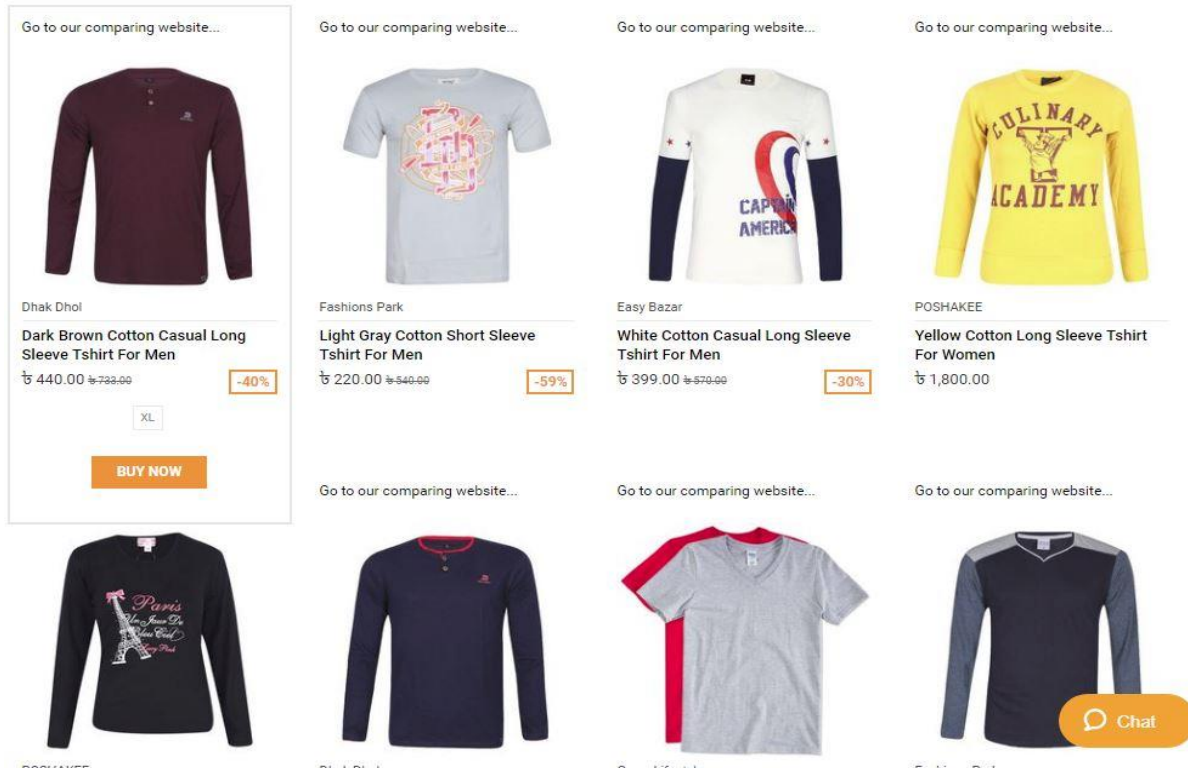


Figure 4.10: Comparing Website.

Chapter 5

Implementation Plan

Implementation plan is one of the most important part for implementing the project. In this chapter we will try to discuss more about key term to implement the process. Implementing the process we use both **PHP** and **JavaScript (Fabric.js)** as an open source. We mainly used so that our project can be most secured and more reliable as well.

We implemented our project along with **HTML, CSS, and Bootstrap** for front-end designing,

Apache server as a **local host**,

MySQL as a **Database Management System**,

PHP for **Server Side Language**,

Lastly a package manager **Fabric.js** is used to manage product designing platform.

5.1 Project co-ordination

We managed coordination for our project mainly by keeping each other up to date with current event. We separated the implemented task of our project in different module among us. The coordination is mainly done by online chatting and meeting each other and also sharing our opinions.

5.2 Project Merging

We merged our project with great care and sharing our files face to face mainly but on the other hand we merged our project basically online based site (Google Drive).

Chapter 6

Security and System

System Security is responsible for controlling access to system resources, which will include sensitive data. The system must therefore include a certain amount of protection for such data, and must in turn control access to those parts of the system that administer this protection. System security is concerned with all aspects of these arrangements. The purpose of security of the system is to prevent unauthorized access to the system as well as to store the data in such a way that it doesn't get lost. That is why, in this project we have added the different functionalities for both the Admin and the Users. The general Users can only view the data and shop that. But only the Admin has the privileges to modify the data which includes inserting, updating and deleting the data whenever it is necessary.

6.1 Security for User

In this system, the Users have a unique id and password. So no unauthorized persons get the permission to access any other's account. And so it is secured with a id and password and so the personal information of the users are secured and nobody can access that until knowing the id and password.

6.2 Security for Admin

In this system, the Admin have also a unique id and password and it is fully secured. So unauthorized persons don't have the access to the Admin area and without entering into the Admin area, nobody can change or edit and delete the system information.

6.3 Secured with Session Cookie

WebPages have no memories. A user going from page to page will be treated by the website as a completely new visitor. Session cookies enable the website you are visiting to keep track of your movement from page to page so you don't get asked for the same information you've already given to the site. Cookies allow you to proceed through many pages of a site quickly and easily without having to authenticate or reprocess each new area you visit.

Session cookies allow users to be recognized within a website so any page changes or item or data selection we do is remembered from page to page. The most common example of this functionality is the shopping cart feature of any e-commerce site. When we visit one page of a catalog and select some items, the session cookie remembers our selection so our shopping cart will have the items selected when we are ready to check out. Without session cookies, if we click CHECKOUT, the new page does not recognize our past activities on prior pages and our shopping cart will always be empty.

We can adjust your session cookies through the settings feature of your browser.

Without cookies, websites and their servers have no memory. A cookie, like a key, enables swift passage from one place to the next. Without a cookie every time we open a new web page the server where that page is stored will treat us like a completely new visitor.

Websites typically use session cookies to ensure that we are recognized when we move from page to page within one site and that any information we have entered is remembered. For example, if an e-commerce site did not use session cookies then items placed in a shopping basket would disappear by the time we reach the checkout. We can choose to accept session cookies by changing the settings in your browser.

6.4 Security of Database

Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks.

It is a broad term that includes a multitude of processes, tools and methodologies that ensure security within a database environment. Database security covers and enforces security on all aspects and components of databases. This includes:

- Data stored in database
- Database server
- Database management system (DBMS)
- Other database workflow applications

Database security is generally planned, implemented and maintained by a database administrator and or other information security professional.

Some of the ways database security is analyzed and implemented include:

- Restricting unauthorized access and use by implementing strong and multifactor access and data management controls.
- Load/stress testing and capacity testing of a database to ensure it does not crash in a distributed denial of service (DDoS) attack or user overload.

- Physical security of the database server and backup equipment from theft and natural disasters.
- Reviewing existing system for any known or unknown vulnerabilities and defining and implementing a road map/plan to mitigate them.

Chapter 7

Conclusion & Future Scope

In the age of technology, we are surrounded with many things which make our life faster and easier. In business, technology has added a new dimension. E-commerce is one of the most progressing and is becoming more and more important to businesses as technology continues to advance and is something that should be taken advantage of and implemented.

From the inception of the Internet and e-commerce, the possibilities have become endless for both businesses and consumers. Create more opportunities for profit and advancements for businesses, while creating more options for consumers.

However, just like anything else, e-commerce has its disadvantages including consumer uncertainties, but nothing that cannot be resolved or avoided by good decision-making and business practices.

In our web application we make compare the several e-commerce website based on the similar product and their prices. Consumer will find ease not to compare manually but to automatic process which we provide in our e-commerce web application. There is our product and others product and we can compare both in a single platform. There were a also a feature which is the extension thing which can be very helpful for the consumer to find the similar product in other website as well.

In this era the world is becoming so small as the world is in everyone's pocket. So the e-commerce business is worth for this era. Technology makes us idle so sooner we won't go for shopping physically. And the trend graph of e-commerce says 2004 to 2017. Both in 2004 and 2016 hipe was in the peak but there is huge difference between them. As in 2004 there was a hipe created by the western or developed country but in the 2010 in gradually down as people were used to it but now it goes up because the it is now creating a hipe in a developing country. In world map if we see the most hipe created in the Indian subcontinent. In individual index the India has 64 point and Bangladesh has 63. So there is a lot of scope in this market. Unique feature has already established in our website and if we collaborate more and more e-commerce sites through marketing, there is a lot of good opportunity in our country and the world as well.

Reference

1. Google Trends. (2017). Google Trends. [online] Available at: <https://trends.google.com/trends/explore?q=%2Fm%2F02m96> [Accessed 5 Dec. 2017].
2. <https://www.stackoverflow.com>
3. <https://www.tutorialspoint.com>
4. <https://www.trainingwithliveproject.com>

Appendix

Code Snippets:

Admin login:

```
<?php

    $filepath = realpath(dirname(__FILE__));

    include_once ($filepath.'../lib/Session.php');

    Session::checkLogin();

    include_once ($filepath.'../lib/Database.php');

    include_once ($filepath.'../helpers/Format.php');

?>
```

```
<?php

class Adminlogin

{

    private $db;

    private $fm;

    public function __construct()

    {

        $this->db = new Database();

        $this->fm = new Format();

    }

    public function adminLogin($adminUser, $adminPass)
```

```

{
    $adminUser = $this->fm->validation($adminUser);
    $adminPass = $this->fm->validation($adminPass);

    $adminUser = mysqli_real_escape_string($this->db->link, $adminUser);
    $adminPass = mysqli_real_escape_string($this->db->link, $adminPass);

    if (empty($adminUser) || empty($adminPass)) {
        $loginmsg = "<div class='alert alert-warning'><strong>Username or Password
must not be empty !</strong></div>";
        return $loginmsg;
    } else {
        $query = "SELECT * FROM admin WHERE adminUser = '$adminUser' AND
adminPass = '$adminPass'";
        $result = $this->db->select($query);

        if ($result != false) {
            $value = $result->fetch_assoc();
            Session::set("adminlogin", true);
            Session::set("adminId", $value['adminId']);
            Session::set("adminUser", $value['adminUser']);
            Session::set("adminName", $value['adminName']);

            header("Location:dashbord.php");
        } else {
            $loginmsg = "<div class='alert alert-danger'><strong>Username or
Password not match !</strong></div>";
            return $loginmsg;
        }
    }
}

```

```
        }
    }
}

?>
```

Cart:

```
<?php
    $filepath = realpath(dirname(__FILE__));
    include_once ($filepath.'../lib/Database.php');
    include_once ($filepath.'../helpers/Format.php');
?>
```

```
<?php
class Cart
{
    private $db;
    private $fm;

    public function __construct()
    {
        $this->db = new Database();
        $this->fm = new Format();
    }
}
```

```

public function addToCart($data, $id)
{
    $productId = mysqli_real_escape_string($this->db->link, $id);
    $sId = session_id();

    $squery = "SELECT * FROM product WHERE productId = '$productId'";
    $result = $this->db->select($squery)->fetch_assoc();

    $productName = $result['productName'];
    $price = $result['price'];
    $image = $result['image'];
    $pd_type = $result['pd_type'];
    $catId = $result['catId'];

    $schquery = "SELECT * FROM cart WHERE productId = '$productId' AND sId = '$sId'";
    $getCpd = $this->db->select($schquery);
    if ($getCpd) {
        $msg = "<div class='alert alert-warning'><strong>Product already added!</strong> You should <a href='cart.php' class='alert-link'>Check Your Cart</a>.</div>";
        return $msg;
    } else {
        if ($catId == 1 || $catId == 2 || $catId == 3) {
            $s_size = $this->fm->validation($data['s_size']);
            $s_size = mysqli_real_escape_string($this->db->link, $data['s_size']);
            $m_size = $this->fm->validation($data['m_size']);
            $m_size = mysqli_real_escape_string($this->db->link, $data['m_size']);
            $l_size = $this->fm->validation($data['l_size']);

```

```

$l_size = mysqli_real_escape_string($this->db->link, $data['l_size']);

$quantity = $s_size + $m_size + $l_size;

if ($quantity >= 6) {

    $query = "INSERT INTO cart(sId, catId, productId, productName, price, quantity, s_size,
m_size, l_size, image, pd_type) VALUES('$sId', '$catId', '$productId', '$productName', '$price',
'$quantity', '$s_size', '$m_size', '$l_size', '$image', '$pd_type')";

    $pd_inserted = $this->db->insert($query);

    if ($pd_inserted) {

        echo "<script>>window.location = 'cart.php';</script>";

    } else {

        echo "<script>>window.location = '404.php';</script>";

    }

} else {

    $msg = "<div class='alert alert-warning'><strong>No product has been quantified.</strong>
You have to order atleast 6 product</div>";

    return $msg;

}

} else {

    $quantity = $this->fm->validation($data['quantity']);

    $quantity = mysqli_real_escape_string($this->db->link, $data['quantity']);

    $query = "INSERT INTO cart(sId, catId, productId, productName, price, quantity, image,
pd_type) VALUES('$sId', '$catId', '$productId', '$productName', '$price', '$quantity', '$image',
'$pd_type')";

```

```

$pd_inserted = $this->db->insert($query);

if ($pd_inserted) {
    echo "<script>>window.location = 'cart.php';</script>";
} else {
    echo "<script>>window.location = '404.php';</script>";
}
}
}
}

public function getCartProduct()
{
    $sId = session_id();
    $query = "SELECT * FROM cart WHERE sId = '$sId'";
    $result = $this->db->select($query);
    return $result;
}

public function updateCartQuantity($data)
{
    $catId = $this->fm->validation($data['catId']);
    $catId = mysqli_real_escape_string($this->db->link, $data['catId']);
    $cartId = $this->fm->validation($data['cartId']);
    $cartId = mysqli_real_escape_string($this->db->link, $data['cartId']);

```



```

if ($catId == 1 || $catId == 2 || $catId == 3) {
    $s_size = $this->fm->validation($data['s_size']);
    $s_size = mysqli_real_escape_string($this->db->link, $data['s_size']);
    $m_size = $this->fm->validation($data['m_size']);
    $m_size = mysqli_real_escape_string($this->db->link, $data['m_size']);
    $l_size = $this->fm->validation($data['l_size']);
    $l_size = mysqli_real_escape_string($this->db->link, $data['l_size']);
    $quantity = $s_size + $m_size + $l_size;
    if ($quantity >= 6) {
        $query = "UPDATE cart SET quantity = '$quantity', s_size = '$s_size', m_size = '$m_size', l_size
= '$l_size' WHERE cartId = '$cartId'";

        $cartupdated = $this->db->update($query);
        if ($cartupdated) {
            echo "<script>>window.location = 'cart.php';</script>";
        } else {
            $msg = "<div class='alert alert-danger'><strong>Quantity Not Updated !</strong></div>";
            return $msg;
        }
    } else {
        $msg = "<div class='alert alert-warning'><strong>You have to order atleast 6
product</strong></div>";
        return $msg;
    }
}
} else {

```

```

$quantity = $this->fm->validation($data['quantity']);
$quantity = mysqli_real_escape_string($this->db->link, $data['quantity']);

$query = "UPDATE cart SET quantity = '$quantity' WHERE cartId = '$cartId'";

$cartupdated = $this->db->update($query);
if ($cartupdated) {
    echo "<script>>window.location = 'cart.php';</script>";
} else {
    $msg = "<div class='alert alert-danger'><strong>Quantity Not Updated !</strong></div>";
    return $msg;
}
}
}

public function delPdFromCart($delId)
{
    $delId = $this->fm->validation($delId);
    $delId = mysqli_real_escape_string($this->db->link, $delId);

    $query = "DELETE FROM cart WHERE cartId = '$delId'";

    $cartdeleted = $this->db->delete($query);
    if ($cartdeleted) {
        echo "<script>>window.location = 'cart.php';</script>";
    } else {

```

```
$msg = "<div class='alert alert-danger'><strong>Product Not Deleted !</strong></div>";  
return $msg;  
}  
}
```

```
public function checkCartTable()  
{  
    $sId = session_id();  
    $query = "SELECT * FROM cart WHERE sId = '$sId'";  
    $result = $this->db->select($query);  
    return $result;  
}
```

```
public function delCustomerCart()  
{  
    $sId = session_id();  
    $query = "DELETE FROM cart WHERE sId = '$sId'";  
    $this->db->delete($query);  
}
```

```
public function insertOrderedProduct($cmrId)  
{  
    $sId = session_id();  
    $query = "SELECT * FROM cart WHERE sId = '$sId'";  
    $getPd = $this->db->select($query);  
    if ($getPd) {
```

```

while ($result = $getPd->fetch_assoc()) {
    $productId = $result['productId'];
    $productName = $result['productName'];
    $quantity = $result['quantity'];
    $s_size = $result['s_size'];
    $m_size = $result['m_size'];
    $l_size = $result['l_size'];
    $price = $result['price'];
    $image = $result['image'];
    $pd_type = $result['pd_type'];

    $query = "INSERT INTO cus_order(cmrId, productId, productName, quantity, s_size, m_size,
l_size, price, image, pd_type) VALUES('$cmrId', '$productId', '$productName', '$quantity', '$s_size',
'$m_size', '$l_size', '$price', '$image', '$pd_type')";

    $od_inserted = $this->db->insert($query);

    if ($od_inserted) {
        $queryCus = "SELECT email FROM customer WHERE id = '$cmrId'";
        $cus_result = $this->db->select($queryCus);
        return $cus_result;
    }
}

}

}

}

public function payableAmount($cmrId)
{

```

```
$query = "SELECT price FROM cus_order WHERE cmrId = '$cmrId' AND date = now()";  
$result = $this->db->select($query);  
return $result;  
}
```

```
public function getOrderedProduct($cmrId)  
{  
    $query = "SELECT * FROM cus_order WHERE cmrId = '$cmrId' ORDER BY date DESC";  
    $result = $this->db->select($query);  
    return $result;  
}
```

```
public function checkOrder($cmrId)  
{  
    $query = "SELECT * FROM cus_order WHERE cmrId = '$cmrId'";  
    $result = $this->db->select($query);  
    return $result;  
}
```

```
public function getAllOrderedProduct()  
{  
    $query = "SELECT * FROM cus_order ORDER BY date DESC";  
    $result = $this->db->select($query);  
    return $result;  
}
```

```

public function getPendingOrder()
{
    $query = "SELECT * FROM cus_order WHERE status = '0' ORDER BY date DESC";
    $result = $this->db->select($query);
    return $result;
}

public function productShifted($id, $date, $price)
{
    $id = $this->fm->validation($id);
    $id = mysqli_real_escape_string($this->db->link, $id);
    $date = $this->fm->validation($date);
    $date = mysqli_real_escape_string($this->db->link, $date);
    $price = $this->fm->validation($price);
    $price = mysqli_real_escape_string($this->db->link, $price);

    $query = "UPDATE cus_order SET status = '1' WHERE cmrId = '$id' AND date = '$date' AND price
= '$price'";

    $updated_row = $this->db->update($query);
    if ($updated_row) {
        $msg = "<div class='alert alert-success'><strong>Updated Successfully !</strong></div>";
        return $msg;
    } else {
        $msg = "<div class='alert alert-danger'><strong>Not Updated !</strong></div>";
        return $msg;
    }
}

```

```

}

public function cancelOrder($id, $date, $price)
{
    $id = $this->fm->validation($id);
    $id = mysqli_real_escape_string($this->db->link, $id);
    $date = $this->fm->validation($date);
    $date = mysqli_real_escape_string($this->db->link, $date);
    $price = $this->fm->validation($price);
    $price = mysqli_real_escape_string($this->db->link, $price);

    $query = "UPDATE cus_order SET status = '2' WHERE cmrId = '$id' AND date = '$date' AND price
= '$price'";

    $updated_row = $this->db->update($query);
    if ($updated_row) {
        $msg = "<div class='alert alert-success'><strong>Updated Successfully !</strong></div>";
        return $msg;
    } else {
        $msg = "<div class='alert alert-danger'><strong>Not Updated !</strong></div>";
        return $msg;
    }
}

public function delShiftedProduct($id, $time, $price)
{
    $id = $this->fm->validation($id);

```

```

$id = mysqli_real_escape_string($this->db->link, $id);
$date = $this->fm->validation($time);
$date = mysqli_real_escape_string($this->db->link, $time);
$price = $this->fm->validation($price);
$price = mysqli_real_escape_string($this->db->link, $price);

$query = "DELETE FROM cus_order WHERE cmrId = '$id' AND date = '$date' AND price = '$price'";

$deldata = $this->db->delete($query);
if ($deldata) {
    $msg = "<div class='alert alert-success'><strong>Data Deleted Successfully !</strong></div>";
    return $msg;
} else {
    $msg = "<div class='alert alert-danger'><strong>Data Not Deleted !</strong></div>";
    return $msg;
}
}

public function soldPdInserted($id, $time, $price)
{
    $id = $this->fm->validation($id);
    $id = mysqli_real_escape_string($this->db->link, $id);
    $date = $this->fm->validation($time);
    $date = mysqli_real_escape_string($this->db->link, $time);
    $price = $this->fm->validation($price);
    $price = mysqli_real_escape_string($this->db->link, $price);

```



```

    $squery = "SELECT * FROM cus_order WHERE cmrId = '$id' AND date = '$date' AND price = '$price'";

    $getPd = $this->db->select($squery);

    if ($getPd) {
        while ($result = $getPd->fetch_assoc()) {
            $oId = $result['id'];

            $cmrId = $result['cmrId'];

            $productId = $result['productId'];

            $productName = $result['productName'];

            $quantity = $result['quantity'];

            $price = $result['price'] * $quantity;

            $pd_type = $result['pd_type'];
        }

        $query = "INSERT INTO pd_sold(oId, cmrId, productId, productName, quantity, price, pd_type)
VALUES('$oId', '$cmrId', '$productId', '$productName', '$quantity', '$price', '$pd_type')";

        $od_inserted = $this->db->insert($query);
    }
}

public function getSoldProduct($t_date)
{
    $query = "SELECT * FROM pd_sold WHERE date LIKE '%$t_date%'";

    $result = $this->db->select($query);

    return $result;

    //$query = "SELECT * FROM pd_sold WHERE date BETWEEN '$t_date' AND '2017-11-01'";
}

```

```

}

public function getSoldProduct2($t1_date, $t2_date)
{
    $query = "SELECT * FROM pd_sold WHERE date BETWEEN '$t1_date' AND '$t2_date'";
    $result = $this->db->select($query);
    return $result;
}

public function getYear()
{
    $query = "SELECT sum(price) as total FROM pd_sold WHERE YEAR(date) = YEAR(curdate()) ; ";
    $result = $this->db->select($query);
    return $result;
}

public function getMonth($i)
{
    $query = "SELECT sum(price) as total2 FROM pd_sold WHERE MONTH(date) = '$i' ; ";
    $result = $this->db->select($query);
    return $result;
}

}

?>

```