

SUPERCART E-COMMERCE WITH PAYMENT GATEWAY

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

**CHIMAKURTHY CHAITANYASAI PRASANNA
KUMAR**

(37110148)

&

CHUNDRU RAM KUMAR

(37110164)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119
APRIL - 2021**



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafidework **Chimakurthy Chaitanyasai Prasanna Kumar** (37110148) & **Chundru Ram Kumar** (37110164) who carried out the project entitled "**Supercart e-commerce with payment gateway**" under our supervision from November, 2020 to April 2021.

Internal Guide

Dr. A. Mary Posonia, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,
Dr.L.LAKSHMANAN,M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **Chimakurthy Chaitanyasai Prasanna Kumar and Chundru Ram Kumar** hereby declare that the Project Report entitled “ **Supercart e-commerce with payment gateway** ” is done by us under the guidance of Associate Professor **Dr. A. Mary Posonia, M.E., Ph.D.**, of the Department of CSE at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATES

1.

2.

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, Dean, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.**, and **Dr. L. Lakshmanan, M.E., Ph.D.**, Heads of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Mary Posonia, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of who were helpful in many ways for the completion of the project.

ABSTRACT

As a part of the global economy online transactions are rapidly increasing now, it is becoming more important for companies to be able to accept payments online. This paper reviews the literature and offers current knowledge that focuses on electronic commerce for IS researchers and instructors. We inspect the modules of e-credit providers in this paper and explain how a single transaction is processed by each system. Analyze the market leaders in each sector and summaries each company's strengths and weaknesses. The Guidelines for choosing an e-credit provider are presented and the choices that best apply to online companies are highlighted. Example of an online payment provider outline possible areas of future research and offer a basic tutorial on building a business account with distinct Payments Standard websites.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	V
	LIST OF FIGURES	IX
	LIST OF TABLES	X
1.	INTRODUCTION	11
	1.1 OUTLINE OF THE PROJECT	11
	1.2 LITERATURE REVIEW	11
SYSTEM STUDY	12	
OBJECTIVE	12	
2.	AIM AND SCOPE OF PRESENT INVESTIGATION	13
PROPOSED SYSTEM	13	
EXISTING SYSTEMS	14	
3.	ALGORITHMS AND MODERN DESIGN	16
SCHEMATIC DIAGRAM	16	
SEQUENCE DIAGRAM	17	
	ALGORITHM USED	18
	<i>HASHING ALGORITHM</i>	18
SYSTEM REQUIREMENTS	18	
	<i>HARDWARE REQUIREMENTS</i>	18
	<i>SOFTWARE REQUIREMENTS</i>	19

REQUIREMENT ANALYSIS 19

FUNCTIONAL	20	
	REQUIREMENTS	
	<i>NON-FUNCTIONAL</i>	20
	<i>REQUIREMENTS</i>	20
FEASIBILITY STUDY		
	ECONOMICAL FEASIBILITY	21
	TECHNICAL FEASIBILITY	21
	SOCIAL FEASIBILITY	21
4.	RESULTS AND DISCUSSIONS	22
	TESTING	22
	TYPES OF TESTING	22
	<i>UNIT TESTING</i>	22
	<i>INTEGRATION TESTING</i>	22
	<i>FUNCTIONAL TESTING</i>	23
	<i>SYSTEM TESTING</i>	23
	<i>WHITE BOX TESTING</i>	23
	<i>BLACK BOX TESTING</i>	24
	TEST STRATEGY AND	24
	APPROACH	
	<i>INTEGRATION TESTING</i>	25
	<i>ACCEPTANCE TESTING</i>	25
	<i>APLHA TESTING</i>	25
	SURVEY	25
	<i>ANALYSIS OF USER</i>	25
	<i>CHARACTERSTICS</i>	

5.	CONCLUSION	29
	REFERENCES	30
	APPENDIX	
	A. SOURCE CODE	31
	B. SCREEN SHOTS	42

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
2.1	SYSTEM ARCHITECTURE	14
2.2	USE CASE DIAGRAM	15
SCHEMATIC	16 DIAGRAM	
	SEQUENCE FLOW DIAGRAM	17
3.3.1	HASH GENERATION FOR PASSWORD	18
4.2.1	UNIT TESTING	22
WHITE BOX TESTING	24	
BLACK BOX TESTING	24	
ANALYSIS OF USER	26 CHARACTERSTICS	
EXISTING WEBSITE USERS		26
	SURVEY MEAN	
EXISTING WEBSITES	27 SATISFACTION LEVEL	
4.4.4	PROPOSED WEBSITE SATISFACTION LEVEL	27
4.4.5	PROPOSED WEBSITE SATISFACTION MEAN	28

LIST OF TABLES

TABLE No.	TABLE NAME	PAGE No.
4.4.1	ANALYSIS OF USER'S CHARACTERSTICS	26

CHAPTER 1

INTRODUCTION

OUTLINE OF THE PROJECT

The main aim of the project is on the online shopping application. The application is very useful where the buyer can directly buy the products from home using internet on their mobile phones or systems. This application reduces a lot of workload for customers as well as owners. The transaction or payment of money is completed in real time system. Online shopping is the process where the customers can buy these products in internet. Online shopping is a type of electronic commerce used for business-to-business and also business-to-consumer transactions. The term web shop also refers to a place of business where web development, web hosting and other types of same related activities take place.

LITERATURE REVIEW

The fast development of web has changed our very life, and it also no doubt made a deep influence towards the progress in civilization. Against this big background, a new thing named online payment got up huge waves in the ocean of internet. It has changed big and efficient ways towards the e-commerce. However, these ways are not as safe as we thought, and they are risky and pitted with peril. To get rid of the growing hacking concerns and change the way for this growth of online payment on the internet. RUPAY, MasterCard, VISA etc and some other leading companies created this Electronic Transaction (SET). "On February 1st, 1996 these companies announced a technical standard for safeguarding the payment card purchases made over open networks. This standard is called the Secure Electronic Transaction (SET) specification."

The most popular definition of E-Commerce is based on the online perspective of the

conducted business. E-commerce provides the capability of buying and selling products, information and services on the Internet and other online environments. As for any trading activity, the issue of safe and reliable money exchange between transacting parties is essential. In an e-commerce environment, payments take the form of money exchange in an electronic form and are therefore called Electronic Payment (Abrazhevich D & Markopoulos,2009).

The merchant sells the goods to customer and customer pay the price with the help of E-Payment system. In offline world the payment are made with cash or through cheque .In online sales accepting payment is a curial aspect of the transaction (Kaur M,2012).

The spread of e-payment usage varies unevenly between countries partly due to differences in factors such as quality of regulatory framework and readiness of telecommunication infrastructure. New payment services based on the Internet and mobile phones proliferate in the advanced economies. (Izhar A, khan A, Sikandar M.

E-Payment system is secure. There should be no threat to the user credit card number, smart card or other personal detail, payment can be carried out without involvement of third party, It makes E payment at any time through the internet directly to the transfer settlement and form E-business environment (Hossein B,2002).

SYSTEM STUDY

In day-to-day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, household items etc. Now days, it is really hard to get some time to go out and get them by ourselves due to busy life style or lots of works. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online.

We went to rural areas and taken survey from the people about the usage of e-commerce in rural areas, due to lack of knowledge and idea about new technology to them they are facing issues and the performance of e-commerce in rural areas are not effective taking all their considerations into the mind we have developed this

project

OBJECTIVE

- (a)** To identify the area of quality customer service with personal attention.
- (b)** To improve the e-commerce usage in rural areas.
- (c)** To establish strong relationship between bank and customer.
- (d)** It identifies how online payment system work.
- (e)** To fulfill the economical requirements of the business

CHAPTER 2

AIM AND SCOPE OF PRESENT INVESTIGATION

PROPOSED SYSTEM

To remove all the disadvantages of conventional methods, a system is proposed which is an payment gateway. The purpose of online shopping is to save time, save money. Through online shopping one can save his valuable time. One can watch and select things he wants to buy. Through online shopping we can save our money because prices are less than market prices and we receive our bought things at our home. No need to go anywhere and do shopping. We can get different varieties of things online and we can choose which one we want.

Through Online Shopping, a person who wants to buy is a lifestyle e-commerce web application, which retails various fashion and lifestyle products. This project allows viewing various products available enables registered users to purchase desired products instantly using PayPal payment processor (Instant Pay) and also can place order by using Cash on Delivery (Pay Later) option. This project provides an easy access to Administrators and Managers to view orders placed using Pay Later and Instant Pay options. In order to develop an e-commerce website, a number of Technologies must be studied and understood.

MERITS OF PROPOSED SYSTEM

Convenience: This is one of the main reasons that online shopping has become so popular, as it allows you to switch stores and products by clicking a button rather than traveling to a new store.

Selection: Of course, a large selection means that your decision-making process may be a bit more difficult, but it also makes it more likely that you will find a high quality product that truly pleases you.

Immediacy: When you purchase a new product, whether for yourself or for another person, it is always nice to have that product in your possession immediately.

Quality: Needless to say, the quality of a product is also very important. And, while most online shopping offers you the ability to return faulty or imperfect products.

Saving Money: Another very important aspect of any shopping experience is trying to save as much money as possible. One reason that people enjoy online shopping is that you can often find a product more cheaply online than you can in stores.

ARCHITECTURE:

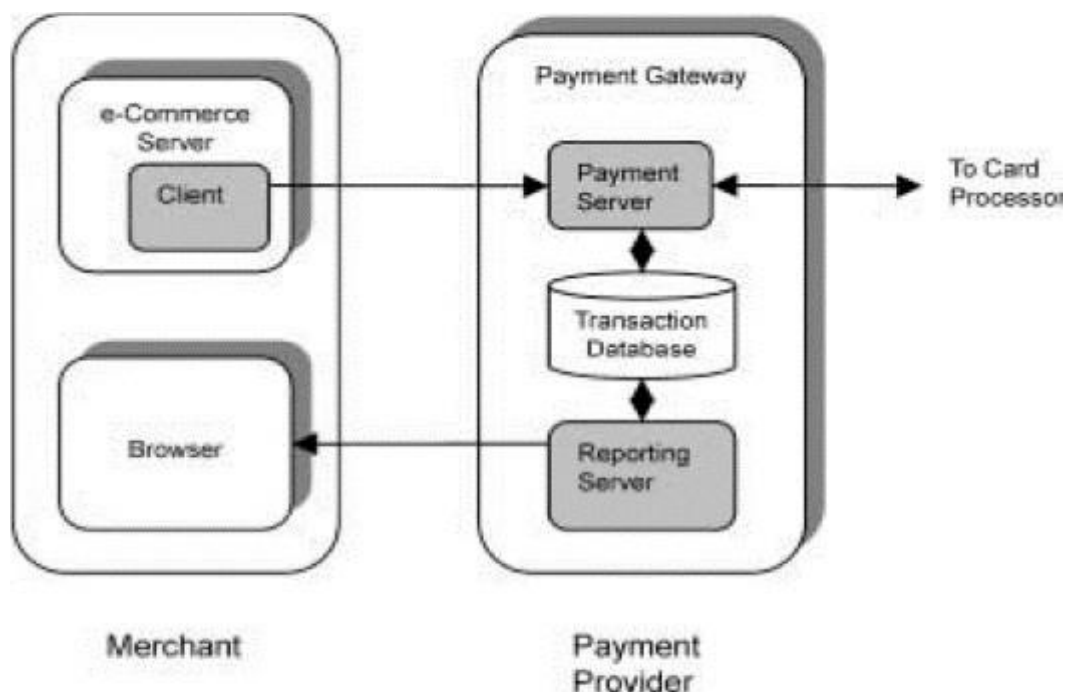


Fig 2.1 : System Architecture.

MODULES

Administrators Administrators are the ones who add or administer the categories for the products, and administers the Vendor. Administrator can see the daily sell. He can also see the feedback given by the customer.

End Users/Customers: The end user will be viewing and buying the products and see the product, their prices and quantity available .Customer can also write feedbacks for products and services.

Welcome/Login Page: The main page, neatly designed page, here login option is provided to login. Also new signup option is provided to for new user to signup.

Product Catcory Display Page: This screen display all the main categories from this point user can select any category and they can proceed to select the products in this category.

View Cart Screen : View cart screen shows all the selected items, here quantity as to be entered. And also option is provided to deselect the products.

Order Details Screen : This screen shows the final products in the view cart. This gives product list, quantity, Total Amount etc, so user will get the clear information that how much is the total amount of this selection.

Payment Gateway

It makes E payment at any time through the internet directly to the transfer settlement and form E-business environment. Studied have been carried out on E-Payment system .E-Payment system an integral part of electronic commerce.An efficient payments system reduces the cost of exchanging goods and services, and is indispensable to the functioning of the interbank, money, and capital markets. Questions are related to E-Payment system in which given options are Agree, Disagree, Strongly disagree, Strongly agree, Neutral.

EXISTING SYSTEM

In existing system we are having a wide variety of e-commerce platforms with various technologies to reduce the steps of placing orders we came up with a

proposed project. Mainly in rural area the existing e-commerce sites are not using in a proper manner because of lack of knowledge on that flow sequence.

Use Case Diagram

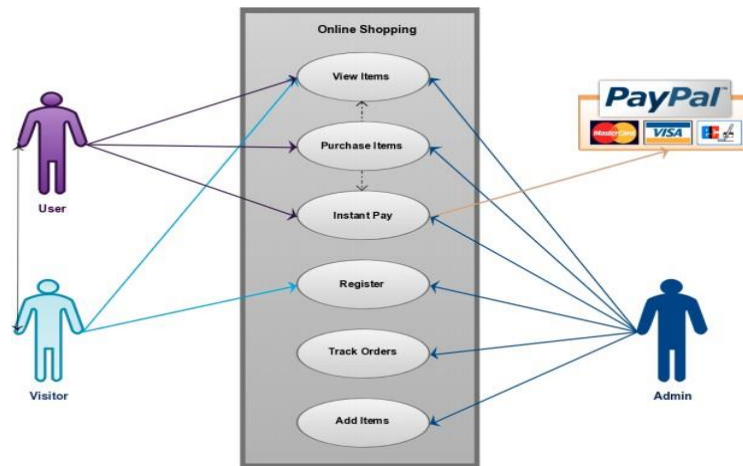


Fig 2.2 Use Case Diagram

CHAPTER 3

ALGORITHMS AND MODEL DESIGN

SCHEMATIC DIAGRAM

To represent the proposed system technically we have drawn a system diagram it gives the complete flow of the proposed project. How it works, what are the modules present and in detail of those. We are having local api in this project everything is controlled and managed by that api.

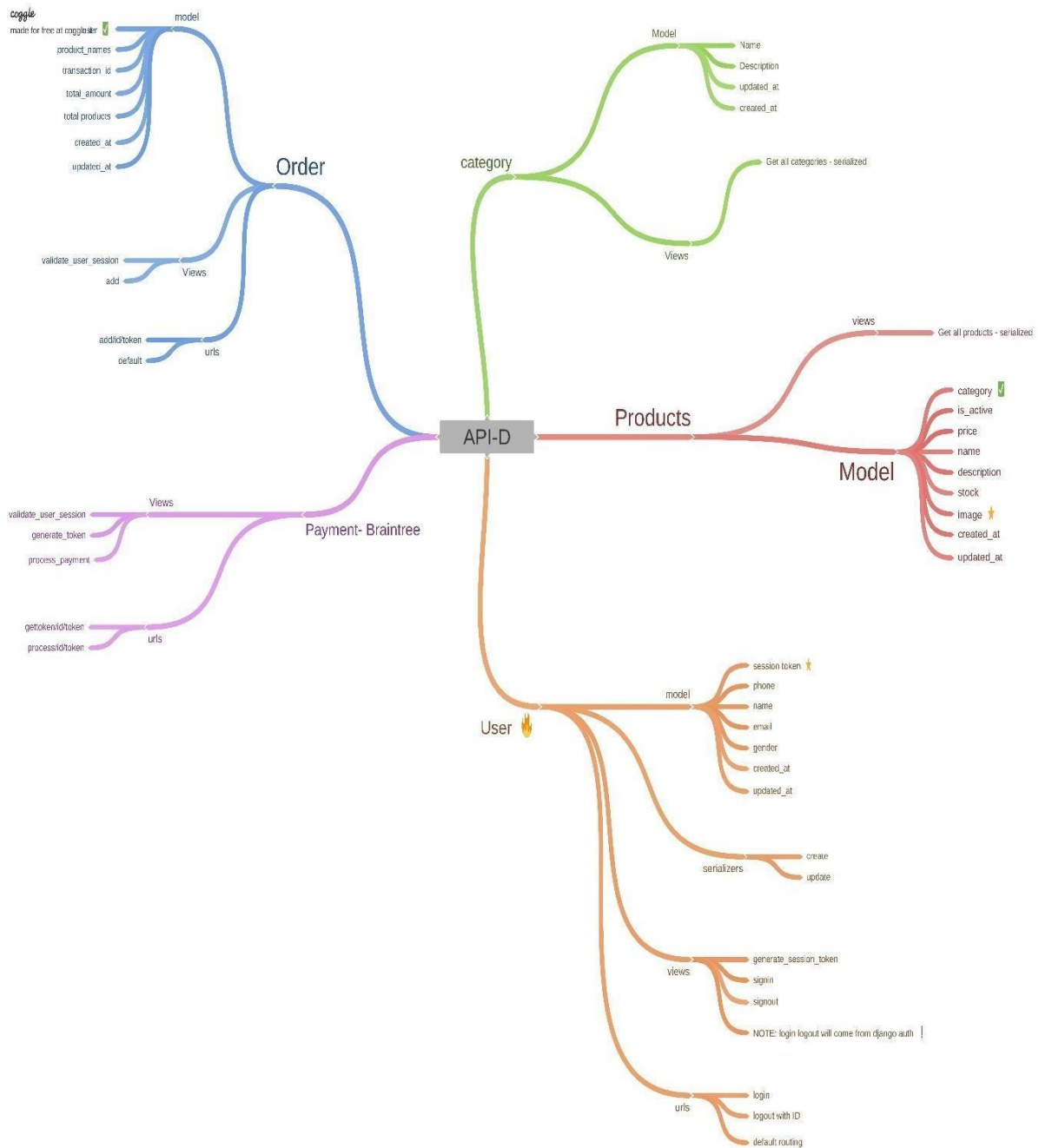


Fig 3.1 Schematic Diagram

SEQUENCE DIAGRAM

In Sequence diagram it shows the complete step by step sequence of working. For suppose I want to place an order first I have to sign-in to the webpage and create an order to be placed, then it will be added with order-id and sent the sequence flow to payment method. In payment method after successful transaction it will give the status of transaction and it will be stored in database for further purpose.

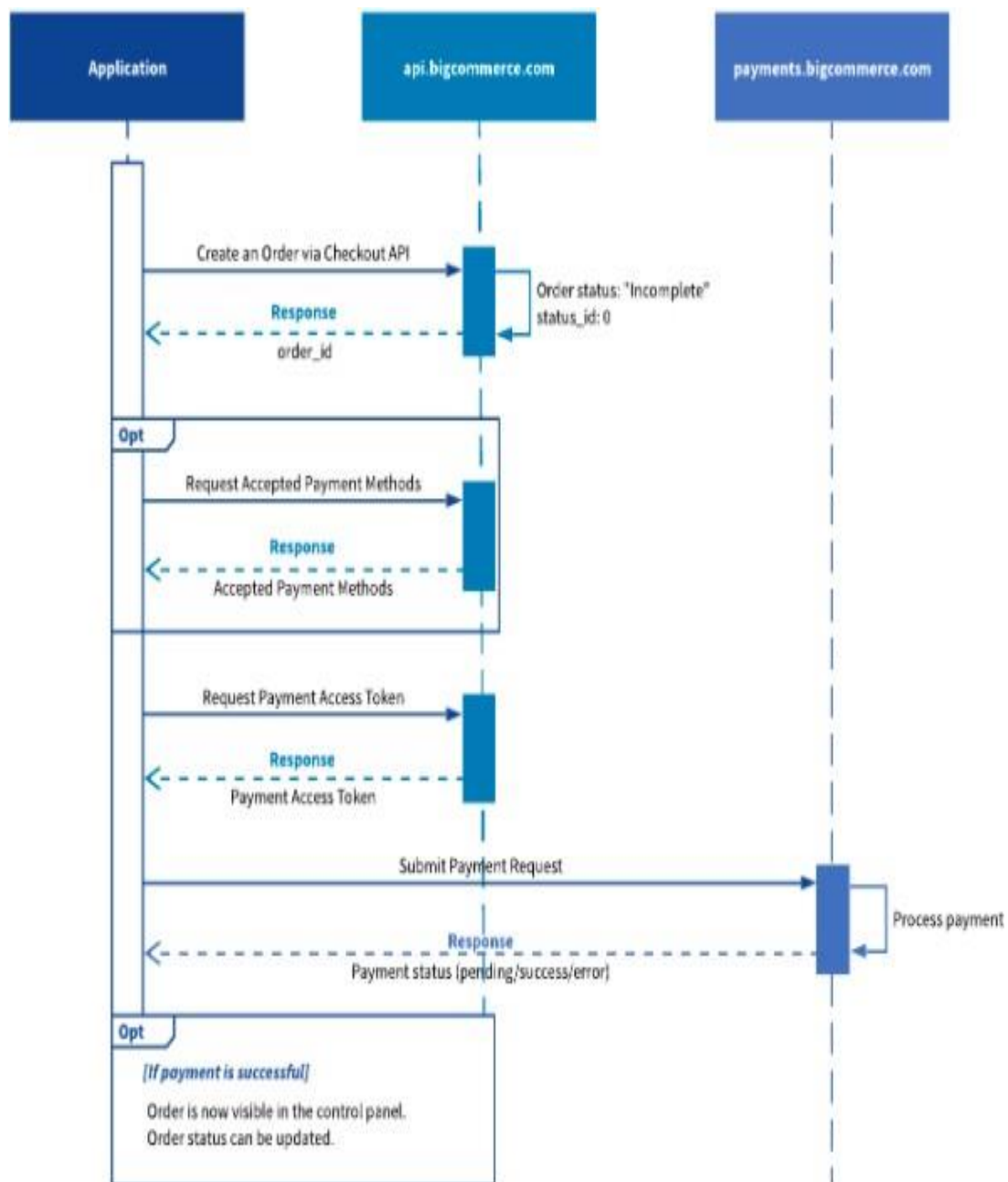


Fig 3.2 Sequence Flow diagram

ALGORITHM USED

HASHING ALGORITHM

It is used to store the customers password in our database. Hash function that takes random size input and yields a fixed-size output. It is easy to calculate but challenging to retrieve original data. It is strong and difficult to duplicate

the same hash with unique inputs and is a one-way function so revert is not possible. Hashing is also known by different names such as Digest, Message Digest, Checksum, etc.

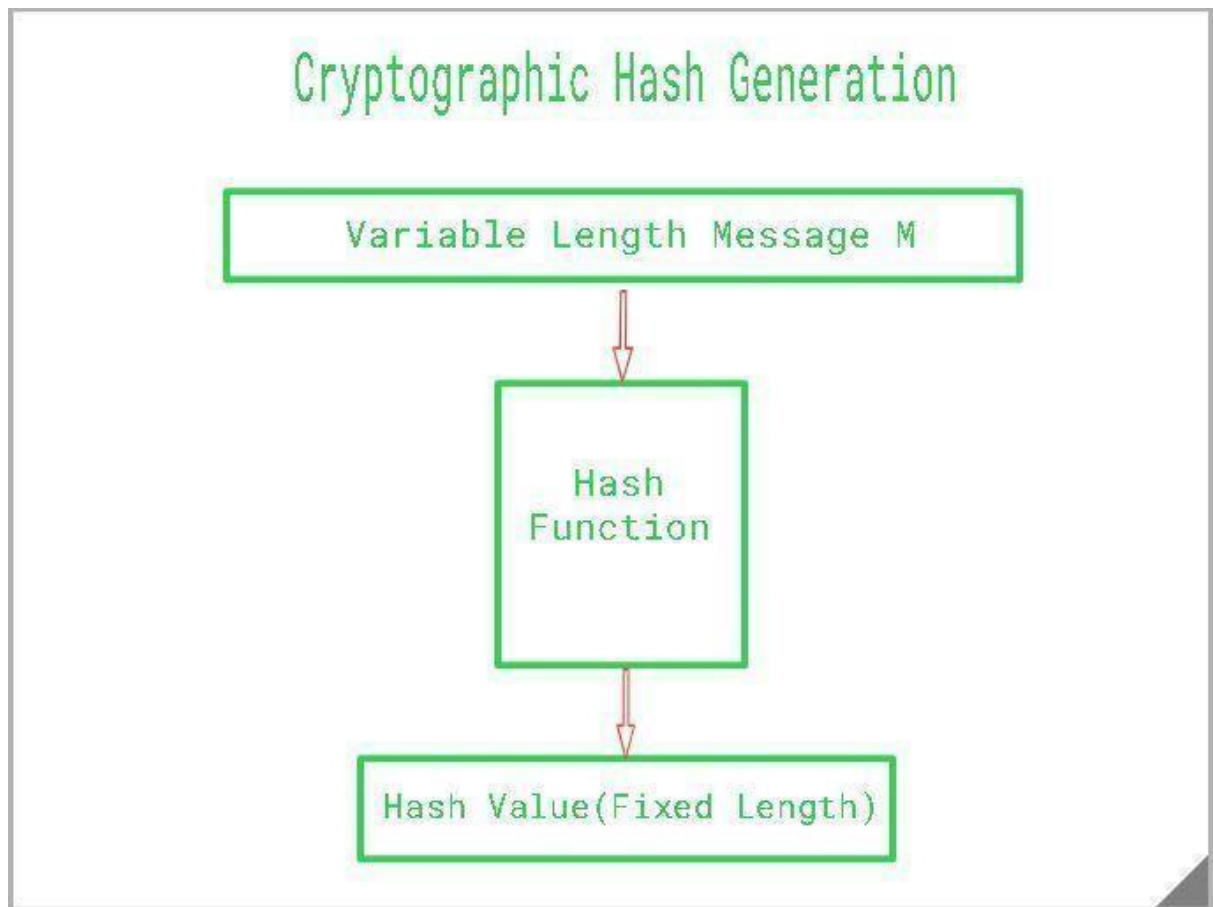


Fig 3.3.1 Hash Generation for password.

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- System : i3 Processor
- Hard Disk : 500 GB.
- Monitor : 15"LED
- Input Devices : Keyboard, Mouse
- Ram : 4GB.

SOFTWARE REQUIREMENTS

- Operating system : Windows

- Coding Language : Python, Java, Html.

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working

around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put

forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. 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 goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system

by the user. This includes the process of training the user to use the system efficiently.

CHAPTER 4

RESULTS AND DISCUSSION

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

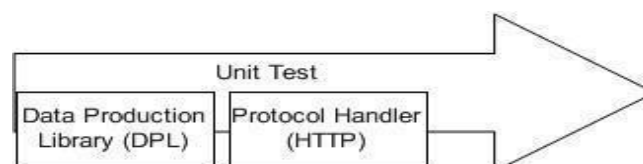


Fig 4.2.1 Unit Testing .

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

Functional testing is centered on the following items:

Valid Input is used to identified classes of valid input must be accepted.

Invalid Input is used to identified classes of invalid input must be rejected.

Functions is used to identified functions must be exercised.

Output is used to identify classes of application outputs

Systems/Procedures is used to interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive Processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

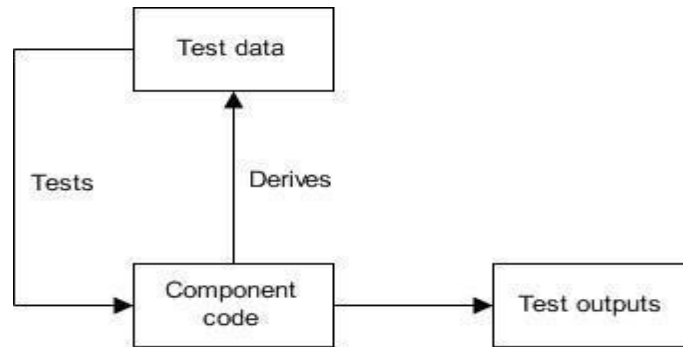


Fig 4.2.5 White Box Testing

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirement Document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without Considering how the software works.

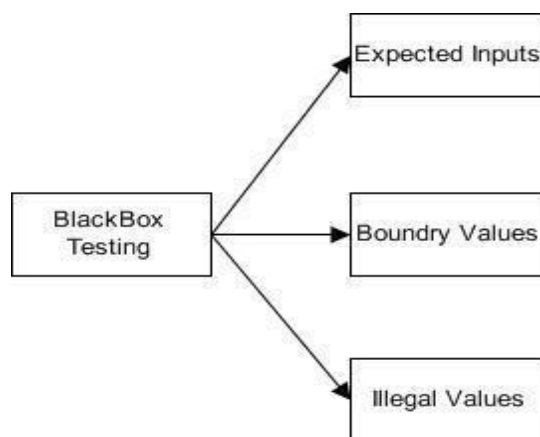


Fig 4.2.6 Black Box Testing

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- ❖ All field entries must work properly.
- ❖ Pages must be activated from the identified link.
- ❖ The entry screen, messages and responses must not be delayed.
- ❖ Features to be tested
- ❖ Verify that the entries are of the correct format
- ❖ No duplicate entries should be allowed
- ❖ All links should take the user to the correct

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

ALPHA TESTING

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are

acceptable and preloads work.

SURVEY

Analysis of user's characteristics

We had taken a survey in rural area like how much they are aware about e-commerce and observed that they can place orders if the process is reduced. Their needs are in 3 steps here First step is sign-in form including address and basic required data second one is categorized wise products and one click to add to cart option and third one is payment gateway with cash-on delivery option as checkout. The survey figure was shown below.

Sr. No	Characteristics	Range	Beginners	Intermediates	Experts
1.	Age	18-25	11	88	37
		26-35	23	44	22
		36-45	24	38	10
		46 & above	8	9	4
2.	Education	Below XII	11	--	--
		XII	25	63	1
		Graduate	20	78	42
		Postgraduate & Above	10	38	30
3.	Frequency of using Computer & Internet	Daily	--	--	73
		Weekly	--	148	--
		Monthly	--	31	--
		Yearly	66	--	--
4.	Frequency of using shopping site	Weekly	--	--	7
		Monthly	--	52	45
		Yearly	66	127	21
5.	Name of shopping site using frequently	Flipkart	31	66	32
		Amazon	24	91	23
		Snapdeal	11	22	18

Fig 4.4.1 Analysis of user's characteristics

From the above figure the survey outputs are classified according to age groups of citizens and differentiated with three types of levels they are beginners, intermediates, experts.

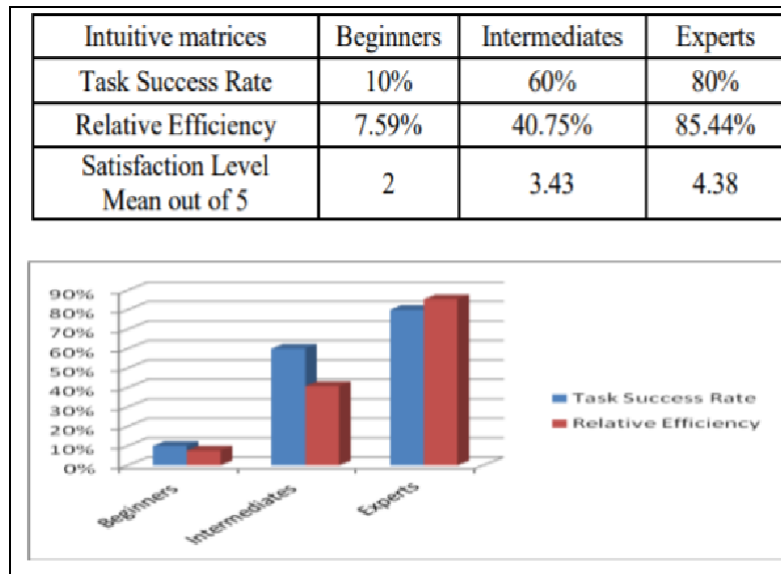


Fig 4.4.2 Existing Website's users survey mean.

The figure 4.4.2 says the mean percentage of the data calculated by using figure 4.4.1 by calculating the mean we can identify the exact point of their view. By using the figure 4.4.2 we have designed a project and considering their needs we have processed further steps in project and makes the process easier.

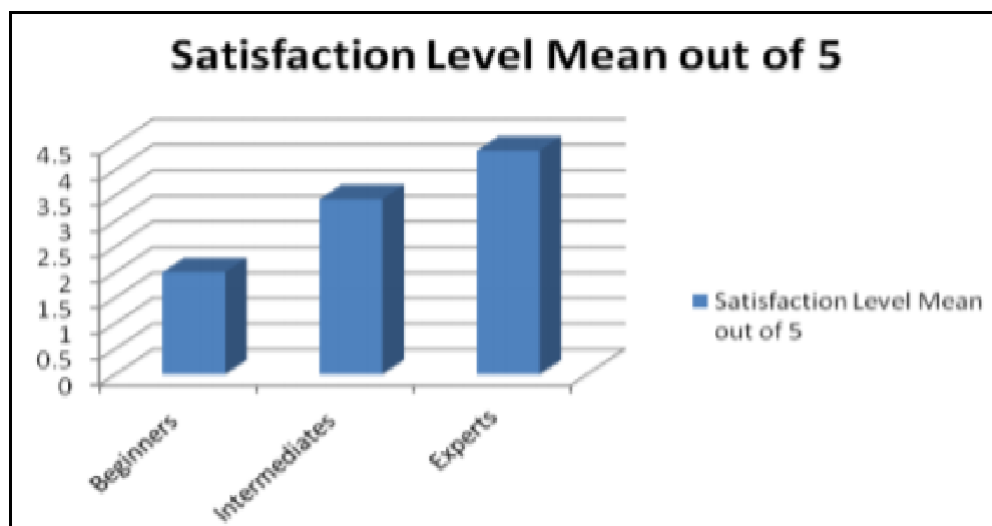


Fig 4.4.3 Existing websites satisfaction level

According to survey the satisfaction level of the people has been shown in the above figure 4.4.3 and taking this into consideration we improved their satisfaction level.

Intuitive matrices	Beginners	Intermediates	Experts
Task Success Rate	70%	90%	100%
Relative Efficiency	79.80%	96.31%	100%
Satisfaction Level Mean out of 5	3.76	4.38	4.76

Fig 4.4.4 Proposed website satisfaction level

In Figure 4.4.4 proposed website satisfaction level has been increased and shown good outcome in survey.

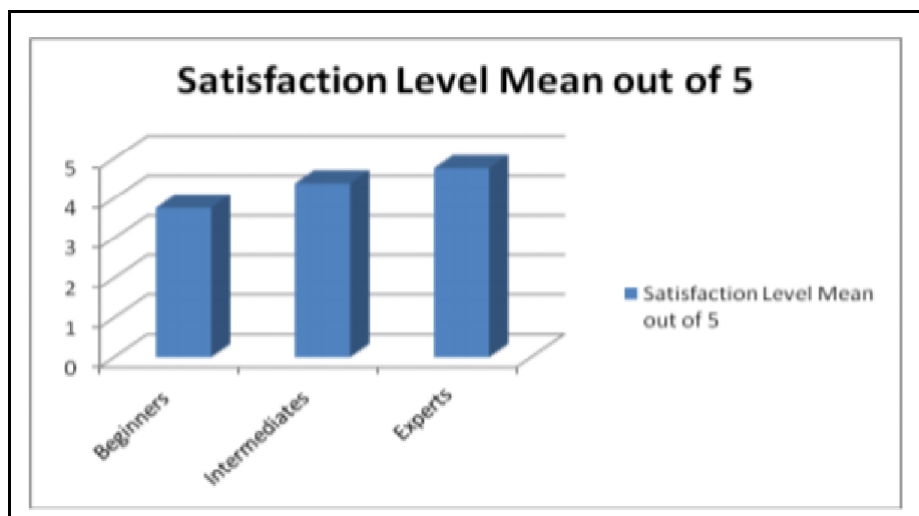


Fig 4.4.5 : Existing websites satisfaction level

The satisfaction level has been increased and steps are reduced in the project as we want to makes the process easier to purchase products through online in rural areas.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

Payment Gateway is a form of Electronic Commerce. E-commerce or Electronic commerce, is a subset of E-Business (a company that does all or most of its transactions through the Internet), is the purchasing, selling, and exchanging of goods and services over computer networks (such as the Internet) through which transactions or terms of sale are performed electronically. Increase in sales. Increase in customers. Ability to be open 24/7. Instant processing of transactions. Increased business reach. Use electronic payment refers to paperless monetary transactions.

REFERENCES

- [1] J. Serrano-Guerrero et al., "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0," *Information Sciences*, Vol. 181, no.9, pp. 1503-1516, May. 2011.
- [2] Z. Zhang et al., "A hybrid fuzzy-based personalized recommender system for telecom products/services," *Information Sciences*, Vol. 235, pp. 117-129, Jun. 2013
- [3] Z.B. Sun et al., "Recommender systems based on social networks," *Journal of Systems and Software*, Vol. 99, pp. 109-119, Jan. 2015
- [4] X.L. Zheng et al., "A Hybrid Trust-Based Recommender System for Online Communities of Practice," *IEEE Transactions on Learning Technologies*, Vol. 8 no. 4, pp. 345-356. Apr. 2015.
- [5] L.O. Colombo-Mendoza et al., "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Systems with Applications*, Vol. 42, no. 3, pp. 1202-1222, Feb. 2015.
- [6] G.M.L. Sarne, "A novel hybrid approach improving effectiveness of recommender

systems,” Journal of Intelligent Information Systems, Vol. 44, no. 3, pp. 397-414, Jun. 2015.

APPENDIX

A. SOURCE CODE

-application

```
package com.example.application;
```

```
import java.math.BigDecimal;
```

```
import java.time.LocalDate;
```

```
import java.util.Arrays;
```

```
import org.springframework.boot.CommandLineRunner;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.vaadin.artur.helpers.LaunchUtil;
```

```

import com.example.application.data.entity.Cart;
import com.example.application.data.entity.Category;
import com.example.application.data.entity.OrderDetails;
import com.example.application.data.entity.Product;
import com.example.application.data.entity.Role;
import com.example.application.data.entity.User;
import com.example.application.data.service.CartRepository;
import com.example.application.data.service.CategoryRepository;
import com.example.application.data.service.OrderRepository;
import com.example.application.data.service.ProductRepository;
import com.example.application.data.service.UserRepository;
import com.example.application.views.login.LoginView;
import com.vaadin.flow.component.UI;

```

```
/**
```

```
 * The entry point of the Spring Boot application.
```

```
*/
```

```
@SpringBootApplication
```

```
public class Application extends SpringBootServletInitializer {
```

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

```
        LaunchUtil.launchBrowserInDevelopmentMode(SpringApplication.run(Application.class, args));
```

```
    }
```

```
@Bean
```

```
public CommandLineRunner run(
```

```
    CategoryRepository categoryRepository,
```

```
    UserRepository userRepository,
```

```
    ProductRepository productRepository,
```

```

        CartRepository cartRepository,
        OrderRepository orderRepository
    ) {
return args -> {

    if(true) {
        //cartRepository.deleteAll();
        productRepository.deleteAll();
    }

    if(orderRepository.count() == 0) {
        orderRepository.saveAll(Arrays.asList(
            OrderDetails.builder()
                .withCreatedAt(LocalDate.now())
                .withProductNames("OnePlus 6, T-Shirt")

.withTotalAmount(BigDecimal.valueOf(1040))
                .withTotalProducts(2)
                .withTransactionId("tx1")
                .withUserId(4)
                .build()
        ));
    }

    if(cartRepository.count() == 0) {
        cartRepository.saveAll(Arrays.asList(
            Cart.builder()
                .withUserId(3)
                .withProductName("OnePlus 6")
                .withQuantity(1)
                .build()
        ));
    }
}
}

```

```

        ));
    }

    if(userRepository.count() == 0) {
        userRepository.saveAll(Arrays.asList(
            User.builder()
                .withRole(Role.ADMIN)
                .withUsername("admin")
                .withPassword("admin123")
                .build(),

            User.builder()
                .withRole(Role.USER)
                .withUsername("user")
                .withPassword("user123")
                .build()

        ));
    }

```

```

    if(categoryRepository.count() == 0)
        categoryRepository.saveAll(
            Arrays.asList(

                Category.builder()
                    .withName("Electronics")
                    .withDescription("Mobile
Phones, Laptops, Audio, TV & Home Entertainment")
                    .build(),

```

```

        Category.builder()
            .withName("Fashion")
            .withDescription("Sportswear,
Men Apparat, Women Apparat, Kids Apparat")
            .build()

    ));

    if(productRepository.count() == 0) {
        productRepository.saveAll(
            Arrays.asList(
                Product.builder()
                    .withCategory("Electronics")

                    .withCreatedAt(LocalDate.now())

                    .withDescription("OnePlus 6
A6003 Dual-SIM (64GB Storage | 6GB RAM) "
                                + "(GSM Only,
No CDMA) Factory Unlocked 4G Smartphone "
                                + "(Mirror Black)
- International Version/No Warranty")

                    .withImage("https://static.toiimg.com/thumb/msid-64260433,width-640,resizemode-4/
64260433.jpg")

                    .withIsActive(true)
                    .withName("OnePlus 6")

                    .withPrice(BigDecimal.valueOf(1000))

                    .withStock("12")
                    .build(),

```

```

        Product.builder()
            .withCategory("Fashion")

.withCreatedAt(LocalDate.now())

            .withDescription(" Symbol is
built on the pillars of quality, reliability and affordability to offer you "
                                + "lifestyle
essentials that never go out of fashion. Never out of style, this Symbol classic "
                                + "polo features
contrast inner placket as an accent detail. Styled from breathable pique cotton, "
                                + "this polo is cut
in an easy regular fit with ribbed collar & sleeves to provide a better grip and
high-low"
                                + " hem for extra
comfort. "
                                + "Wear it alone
or under a jacket to make this collar polo look great.")

.withImage("https://images-na.ssl-images-amazon.com/images/I/81eSlq6bYdL._UL1
500_.jpg")

            .withIsActive(true)
            .withName("Symbol Men's
Solid Regular Polo Shirt")

.withPrice(BigDecimal.valueOf(40))

            .withStock("10")
            .build()

        ));
    }
};
}

```

```
}
```

-payment gateway

```
package com.example.application.paytm;
```

```
import static com.paytm.pg.merchant.PaytmChecksum.generateSignature;
```

```
import static com.paytm.pg.merchant.PaytmChecksum.verifySignature;
```

```
import java.util.Map;
```

```
import java.util.TreeMap;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.core.env.Environment;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
import com.example.application.data.entity.OrderDetails;
```

```
import com.example.application.data.service.OrderRepository;
```

```
import com.vaadin.flow.component.UI;
```

```
@Controller
```

```
public class PaytmController {
```

```
    @Autowired
```

```
    private PaytmDetails paytmDetails;
```

```

@Autowired
private Environment env;

@Autowired
private OrderRepository orderRepository;

@GetMapping("/homepayment")
public String home() {
    return "homepayment";
}

@PostMapping(value = "/pgredirect")
public ModelAndView getRedirect(@RequestParam(name = "CUST_ID")
String customerId,
                                @RequestParam(name = "TXN_AMOUNT") String
transactionAmount,
                                @RequestParam(name = "ORDER_ID") String orderId) throws
Exception {

    ModelAndView modelAndView = new ModelAndView("redirect:" +
paytmDetails.getPaytmUrl());
    TreeMap<String, String> parameters = new TreeMap<>();
    paytmDetails.getDetails().forEach((k, v) -> parameters.put(k, v));
    parameters.put("MOBILE_NO", env.getProperty("paytm.mobile"));
    parameters.put("EMAIL", env.getProperty("paytm.email"));
    parameters.put("ORDER_ID", orderId);
    parameters.put("TXN_AMOUNT", transactionAmount);
    parameters.put("CUST_ID", customerId);
    String checksum = getChecksum(parameters);
    parameters.put("CHECKSUMHASH", checksum);
    modelAndView.addAllObjects(parameters);

```

```

        return modelAndView;
    }

    @PostMapping(value = "/pgresponse")
    public String getResponseRedirect(HttpServletRequest request, Model
model) {

        Map<String, String[]> mapData = request.getParameterMap();
        TreeMap<String, String> parameters = new TreeMap<String, String>();
        mapData.forEach((key, val) -> parameters.put(key, val[0]));
        String paytmChecksum = "";
        if (mapData.containsKey("CHECKSUMHASH")) {
            paytmChecksum = mapData.get("CHECKSUMHASH")[0];
        }
        String result;

        boolean isValidChecksum = false;
        System.out.println("RESULT : " + parameters.toString());
        try {
            isValidChecksum = validateCheckSum(parameters,
paytmChecksum);
            if (isValidChecksum &&
parameters.containsKey("RESPCODE")) {
                if (parameters.get("RESPCODE").equals("01")) {
                    result = "Payment Successful";

orderRepository.save(UI.getCurrent().getSession().getAttribute(OrderDetails.class));
                } else {
                    result = "Payment Failed";
                }
            } else {

```

```

        result = "Checksum mismatched";
    }
} catch (Exception e) {
    result = e.toString();
}
model.addAttribute("result", result);
parameters.remove("CHECKSUMHASH");
model.addAttribute("parameters", parameters);
return "report";
}

private boolean validateChecksum(TreeMap<String, String> parameters,
String paytmChecksum) throws Exception {
    return verifySignature(parameters, paytmChecksum, paytmChecksum);
}

private String getChecksum(TreeMap<String, String> parameters) throws
Exception {
    return generateSignature(parameters,
paytmDetails.getMerchantKey());
}
}

```

- Payment gateway details

```
package com.example.application.paytm;
```

```
import java.util.Map;
```

```
import org.springframework.boot.context.properties.ConfigurationProperties;
```

```
import org.springframework.context.annotation.Configuration;
```

```
@Configuration
```

```
@ConfigurationProperties("paytm.payment.sandbox")
```

```
public class PaytmDetails {
```

```
    private String merchantId;
```

```
    private String merchantKey;
```

```
    private String channelId;
```

```
    private String website;
```

```
    private String industryTypeId;
```

```
    private String paytmUrl;
```

```
    private Map<String, String> details;
```

```
    public String getMerchantId() {  
        return merchantId;  
    }  
}
```

```
    public void setMerchantId(String merchantId) {  
        this.merchantId = merchantId;  
    }  
}
```

```
    public String getMerchantKey() {  
        return merchantKey;  
    }  
}
```

```
public void setMerchantKey(String merchantKey) {  
    this.merchantKey = merchantKey;  
}
```

```
public String getChannelId() {  
    return channelId;  
}
```

```
public void setChannelId(String channelId) {  
    this.channelId = channelId;  
}
```

```
public String getWebsite() {  
    return website;  
}
```

```
public void setWebsite(String website) {  
    this.website = website;  
}
```

```
public String getIndustryTypeId() {  
    return industryTypeId;  
}
```

```
public void setIndustryTypeId(String industryTypeId) {  
    this.industryTypeId = industryTypeId;  
}
```

```
public String getPaytmUrl() {  
    return paytmUrl;  
}
```

```

    }

    public void setPaytmUrl(String paytmUrl) {
        this.paytmUrl = paytmUrl;
    }

    public Map<String, String> getDetails() {
        return details;
    }

    public void setDetails(Map<String, String> details) {
        this.details = details;
    }
}

```

B. SCREENSHOTS

