**Mini Project Report On**

**“An Efficient Secure Electronic Payment System for E-Commerce”**

Submitted to

**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY, HYDERABAD**

In partial fulfilment of the requirement for the award of Degree of

**BACHELOR OF TECHNOLOGY**

In

**“Computer Science and Engineerin"**

By

Karupakula Dharani (19D41A0588)

Jinkala .saiteja (19D41A0574)

K. gowtham kumar (19D41A0584)

K. Lokesh (19D41A05B2)

Under the guidance of

**Mr. S.SATHVIK PRASAD**

**(ASST.PROF)**

Department of Computer Science and Engineering

**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

(Affiliated to JNTU Hyderabad, Approved by AICTE)

Facing main road, Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy Dist.- 501 510

(2022-2023)



**CERTIFICATE**

This is to certify that the project work report entitled **“An Efficient Secure Electronic Payment System for E-Commerce”**which is being submitted by **J.SAITEJA[19D41A0574],K.GOUTHAMKUMAR[19D41A0584],K.DHARANI[19D41A0588],**

**K.LOKESH[19D41A05B2],** in partial fulfilment for the award of the Degree of **BACHERLOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** of **SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY, HYDERABAD,** is a record of the Bonafide work carried out by them under our guidance and supervision.

Guided By HOD

**Mr. S.SATHVIK PRASAD DR.T.CHARAN SINGH**

**ACKNOWLEDGMENT**

We would like to express our deep-felt appreciation and gratitude to **Mr. S.SATHVIK PRASAD**, Department of C.S.E, our project guide, for her skilful guidance, constant supervision, timely suggestion, keen interest and encouragement in completing the individual seminar within the stipulated time.

We express our sincere gratitude to Dr. T. Charan Singh, Professor & Head, Department of Computer Science and Engineering, Sri Indu College of Engineering and Technology, for his suggestions, motivation, and cooperation in the successful completion of the work.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped me for the completion of our project work.

We wish to express our gratitude to the Members of staff and all the others who helped us in more than one way through our project.

**Karupakula.Dharani**

**(19D41A0588)**

**Jinkala.Saiteja**

**(19D41A0574)**

**K.Gowtham kumar**

**(19D41A0584)**

**K.Lokesh**

**(19D41A05B)**

**CONTENTS**

1. **INTRODUCTION**
   1. 1.1. INTRODUCTION TO PROJECT
   2. 1.2. PURPOSE OF THE PROJECT
   3. 1.3. EXISTING SYSTEM & ITS DISADVANTAGES
   4. 1.4. PROPOSED SYSTEM & ITS ADVANTAGES
2. **SYSTEM ANALYSIS**
   1. STUDY OF THE SYSTEM
   2. INPUT & OUTPUT REPRESENTATION
   3. PROCESS MODELS USED WITH JUSTIFICATION
   4. SYSTEM ARCHITECTURE
3. **FEASIBILITY STUDY**
   1. TECHNICAL FEASIBILITY
   2. OPERATIONAL FEASIBILITY
   3. ECONOMIC FEASIBILITY
4. **REQUIREMENT SPECIFICATIONS**
   1. FUNCIONAL REQUIREMENTS
   2. PERFORMANCE REQUIREMENTS
   3. SOFTWARE REQUIREMENTS
   4. HARDWARE REQUIREMENTS
5. **SYSTEM DESIGN**
   1. . INTRODUCTION
   2. UML DIAGRAMS
   3. NORMALIZATION
   4. DATA DICTIONARY
6. **OUTPUT SCREENS**
7. **SYSTEM TESTING** 
   1. INTRODUCTION TO TESTING
   2. TESTING STRATEGIES
8. **SYSTEM SECURITY**

8.1 INTRODUCTION

* 1. SECURITY IN SOFTWARE

**9. CONCLUSION**

**10. BIBLIOGRAPHY**

**ABSTRACT**

E-commerce implies an electronic purchasing and marketing process online by using typical Web browsers. As e-commerce is quickly developing on the planet, particularly in recent years, many areas of life are affected, particularly the improvement in how individuals regulate themselves non-financially and financially in different transactions. In electronic payment or e-commerce payment, the gateway is a major component of the structure to assure that such exchanges occur without disputes while maintaining the common security over such systems. Most Internet payment gateways in e-commerce provide monetary information to customers using trusted third parties directly to a payment gateway. Nonetheless, it is recognized that the cloud Web server is not considered a protected entity. This article aims to develop an efficient and secure electronic payment protocol for e-commerce where consumers can immediately connect with the merchant properly. Interestingly, the proposed system does not require the customer to input his/her identity on the merchant’s website even though the customer can hide his/her identity and make a temporary identity to perform the service. It has been found that our protocol has

**INTRODUCTION**

**1.1 Introduction to the project**

E-commerce was introduced to the consumer and business worlds as a unique approach in 1990 [1]. E-commerce has expanded since then and improved enormously, giving the world’s customers and companies incredible benefits. E-commerce history is closely linked to Internet history. When the Internet was opened to the public in 1991, online shopping was made possible [1,2]. E-commerce is characterized as a primary business model by means of the selling process of goods, the purchasing of resources, and the distribution or exchange over the Internet of items, services, and knowledge [3]. E-commerce can be used with mobile payment systems, which allow customers to pay for their shopping by using smartphones. Mobile business is a major e-commerce extension that enables customers with wireless handheld devices, e.g. tablets, smartphones, and laptops, to carry out online commercial transactions [6]. E-commerce is becoming very popular nowadays since the customer can spend from home; solutions are affordable, with items delivered to the home with no hassle. The popularity of e-commerce is mainly because of its online business perspective. It makes it possible to gain and sell goods online, to provide various services and information through the Internet, and to exchange money immediately between businesses [7]. Many individuals are excited about obtaining their own online website for their company, as it is possible to market items online around the world. Customers are also interested in online shopping since they do not wish to waste valuable time shopping. E-commerce implies an electronic purchasing and marketing process online by using typical Web browsers. It is described as the selling and buying of services or goods through wireless technology.

Developed nations tend to be more acquainted with systems, whereas Internet shopping is exploding in developing nations. The foremost goals of an electronic payment system are increasing efficiency, improving protection, and improving customer convenience and ease of use.

In the electronic payment system, the payment gateway is an essential component of the infrastructure to confirm that such exchanges happen with no concerns and to ensure that the common security over electronic systems is maintained [8,9]. Such a system will help secure a purchase along with a person’s transaction information. A payment gateway defends transaction information by encrypting personal information, such as credit/debit card details, to guarantee that information is transferred securely between a consumer and the transaction processor. Each online exchange should go through a managed transaction gateway. The secure electronic payment structure includes four system segments [10]. The interaction between the segments operates through protected communication tunnels. Secure communication tunnels offer a protected method for interaction between two or more people, or between segments, such as the buyer to the merchant, on the transaction gateway. The e-payment system must be harmless for online transaction applicants, for instance, fee gateway server, bank account server, and merchant server.

This paper is divided into six sections. Section 1 introduces electronic payments and their related study. Section 2 includes an overview of the existing system and the formulation of the problem. Section 3 describes the RSA cryptosystem. Section 4 addresses how the model will be implemented. Section 5 discusses the security analysis and proposed method advantages. Finally, the last section presents the conclusions and future work.

much-improved security effectiveness in terms of confidentiality, integrity, non-repudiation, anonymity availability, authentication, and authorization.

**EXISTING SYSTEM**

A secure and privacy-preserving electronic payment approach can be found in which the authors suggested electronic tokens as being an abstraction of basic fiat currency of equivalent benefit in order to provide privacy and protection in digital payments, presenting an intermediate entity in the method that mediates a transaction between the payer and the pay.

**Disadvantages**

1. Missing privacy
2. Disclosing user’s data

**PROPOSED SYSTEM**

A secure protocol in e-commerce enhances the security of the e-commerce process, which can also improve the security of existing work. Interestingly, the proposed system does not require the customer to input his/her identity on the merchant website even though the customer can hide his/her identity and make a temporary identity to process a request for the service. The proposed system is made up of five entities: client (C), merchant (M), payment gateway (PG), user bank

**Advantages**

1. Improve the security
2. Improving the efficiency

**2.SYSTEM ANALYSIS**

**2.1.STUDY OF THE SYSTEM**

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI’S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

**2.2.INPUT & OUTPUT REPRESENTATION**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**INPUT STAGES:**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**INPUT MEDIA:**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**OUTPUT DESIGN:**

In general are:

* External Outputs whose destination is outside the organization.
* Internal Outputs whose destination is with in organization and they are the User’s main interface with the computer. Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs
* Operational outputs whose use is purely with in the computer department.
* Interface outputs, which involve the user in communicating directly with the system.

**OUTPUT DEFINITION**

# The outputs should be defined in terms of the following points:

* + - Type of the output
    - Content of the output
    - Format of the output
    - Location of the output
    - Frequency of the output
    - Volume of the output
    - Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

* Will decimal points need to be inserted
* Should leading zeros be suppressed.

**OUTPUT MEDIA:**

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

* The suitability for the device to the particular application.
* The need for a hard copy.
* The response time required.
* The location of the users
* The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

**2.3.PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

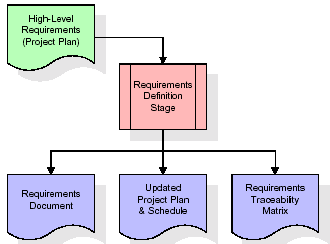
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

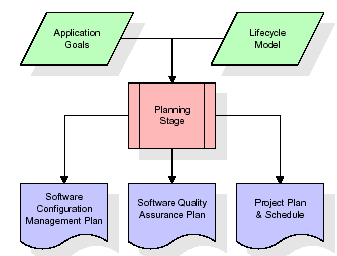
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of+ various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator

**Analysis Stage:**

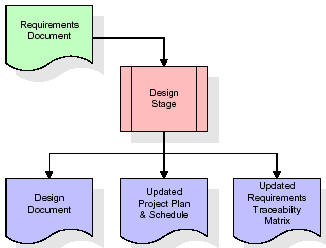
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

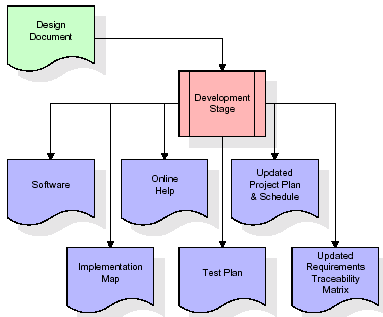
The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

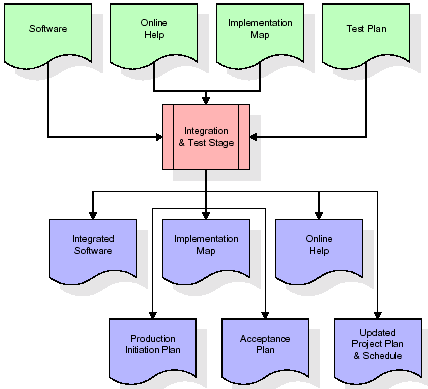
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

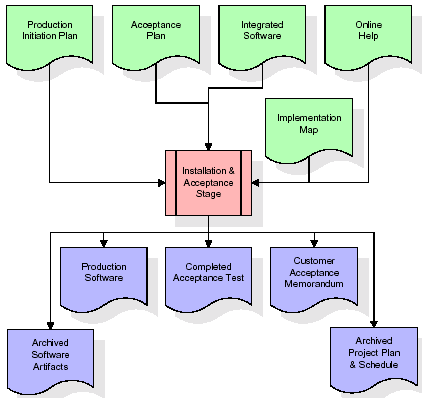


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

**Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will under go training on that particular assigned category.

For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**2.4.SYSTEM ARCHITECTURE**

**Architecture flow:**

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tire architecture.

**SERVER**

**User**

**Data Base**

**Request**

**Response**

**URL Pattern:**

**Presentation Layer**

**SERVLETS AT THE SERVER SIDE**

**URL Request sent through the browser**

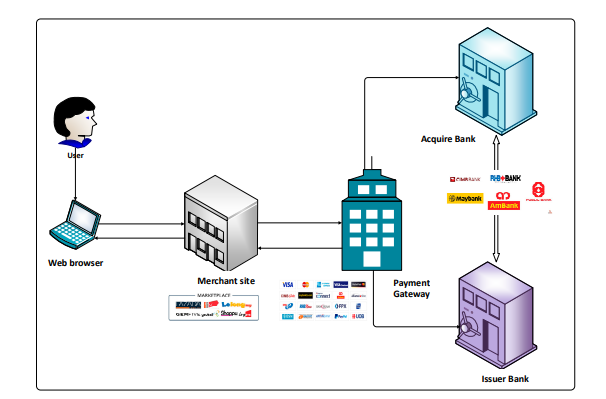
**Response sent from the servlet**

**DATABASE**

**Verifying or updating the database** **through a statement**

**Reply from the database according to the statement**

URL pattern represents how the requests are flowing through one layer to another layer and how the responses are getting by other layers to presentation layer through server in architecture diagram.



**Fig: Architecture**

**3.FEASIBILITY STUDY**

After responsibility the plan Online Grocery System, study and investigating all the current or compulsory functionalities of the organization, the next job is to do the viability study for the project. All plans feasible – given limitless resources and immeasurable time. All the conceivable ways to deliver a solution to the given problem are find by feasibility study. This planned answer would please all the worker need and must be flexible plenty so that future vicissitudes can be simply done founded on the future imminent supplies.

**3.1. Economic Feasibility**

This is very important aspects to be considered while developing a project. We decided the technology for our project founded on smallest conceivable charge influence. Entirely tools and system fee obligates to be done by developer. Completely we have projected that the benefits the creator is going to receive from the planned system will surely dazed the initial prices and the later on organizational cost for system.

**3.2. Technical Feasibility**

The technical feasibility education contains study of function, presentation and restraints that may move the ability to achieve a suitable system. For this possibility study, we deliberate whole functionality to be in the organization, as labelled in the System Obligation Specification (SOS), and checked if the whole thing was possible using the different types of frontend and backend podiums.

* 1. **Operational Feasibility**

This planned system is fully GUI grounded that is very user responsive and all inputs to be occupied all self-descriptive even to a layman. In addition, a proper exercise has been conducted to knowing core of the system to the operators so that they feel relaxed with new system initial. As far our study is apprehension the users are relaxed and happy as the structure has cut down their tons and

doing.

**4.1 FUNCTIONAL REQUIREMENTS SPECIFICATION**

This application mainly consist four modules

1. **Object Aspect**
2. **Behavior Aspect**
3. **Structure Aspect**

**Object Aspect:**

In this module we are selecting one web application. In this module it finds the request goes to from which jsp/servlet to which jsp and the response goes to from which jsp/servlet file to which jsp.

**Behavior Aspect:**

In this module it will find the behavior of the application. It will find the control or action goes from where to where.

**Structure Aspect:**

In this module it will shows the structure of whole application.

**4.2 PERFORMANCE** **REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

The system should be able to interface with the existing system

* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**4.3 Soft Ware Requirements**

* **Operating System** : Windows Family or higher version
* **Techniques**  : JDK 1.7
* **Data Bases** : Mysql
* **Server** :Apache Tomcat

**4.4 Hard ware Requirements:**

* **Processor:: Pentium-III (or) Higher**
* **Ram:: 64MB (or) Higher**
* **Cache:: 512MB**
* **Hard disk:: 10GB**

**SYSTEM DESIGN**

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

****

****

****

****

**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**COLLABORATION:**

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



**Activity Diagram**

Login Check

Login Again

No yes

View products

Generate Account No

Admin

Bank

Merchant

Data user

View Users & Authorize

Create account

Create Account

Add Merchants & View

View Request& transfer Amount

Upload products

searchFiles&Buy

Add Banks & View

View purchased Products

End

**Dataflow Diagram**

TPA

System

Data Owner

Data Server

Request

KGC

Response

**5.3. NORMALIZATION**

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files. The tables are organized to:

* Reduced duplication of data.
* Simplify functions like adding, deleting, modifying data etc..,
* Retrieving data
* Clarity and ease of use
* More information at low cost

# Normalization

Normalization is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints on the kind of functional dependencies that could be associated with the relation. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

**First Normal Form:**

A relation R is in first normal form if and only if all underlying domains contained atomic values only.

**Second Normal Form:**

A relation R is said to be in second normal form if and only if it is in first normal form and every non-key attribute is fully dependent on the primary key.

**Third Normal Form:**

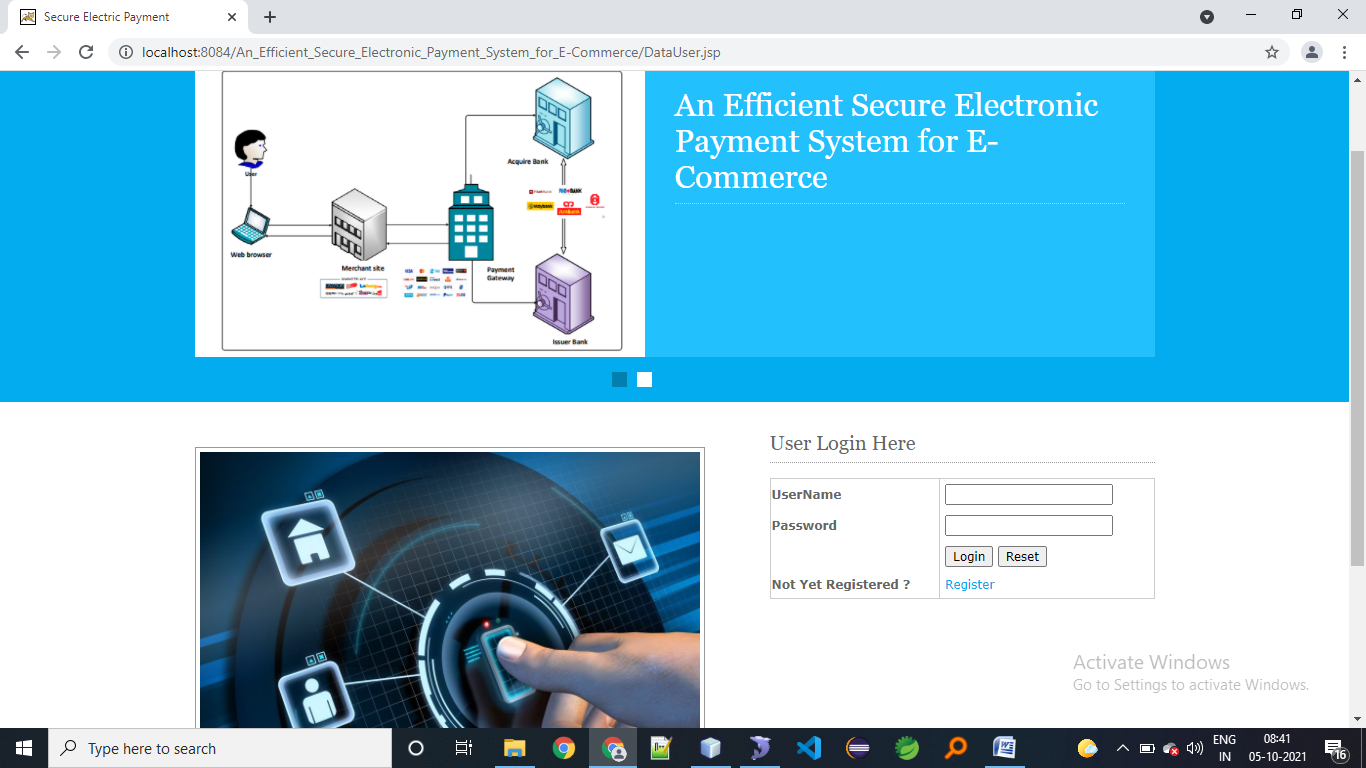
A relation R is said to be in third normal form if and only if it is in second normal form and every non key attribute is non transitively depend on the primary key.

**Screen shots**

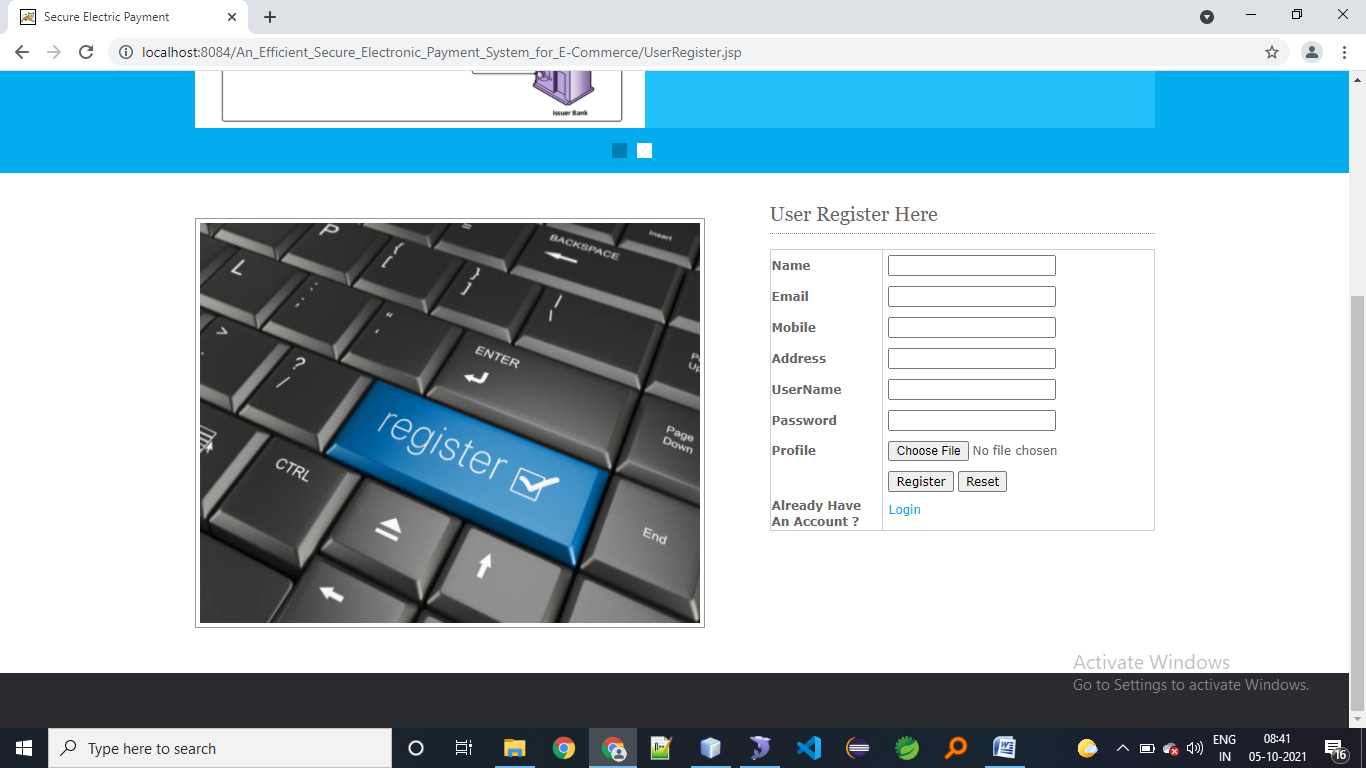
Homepage



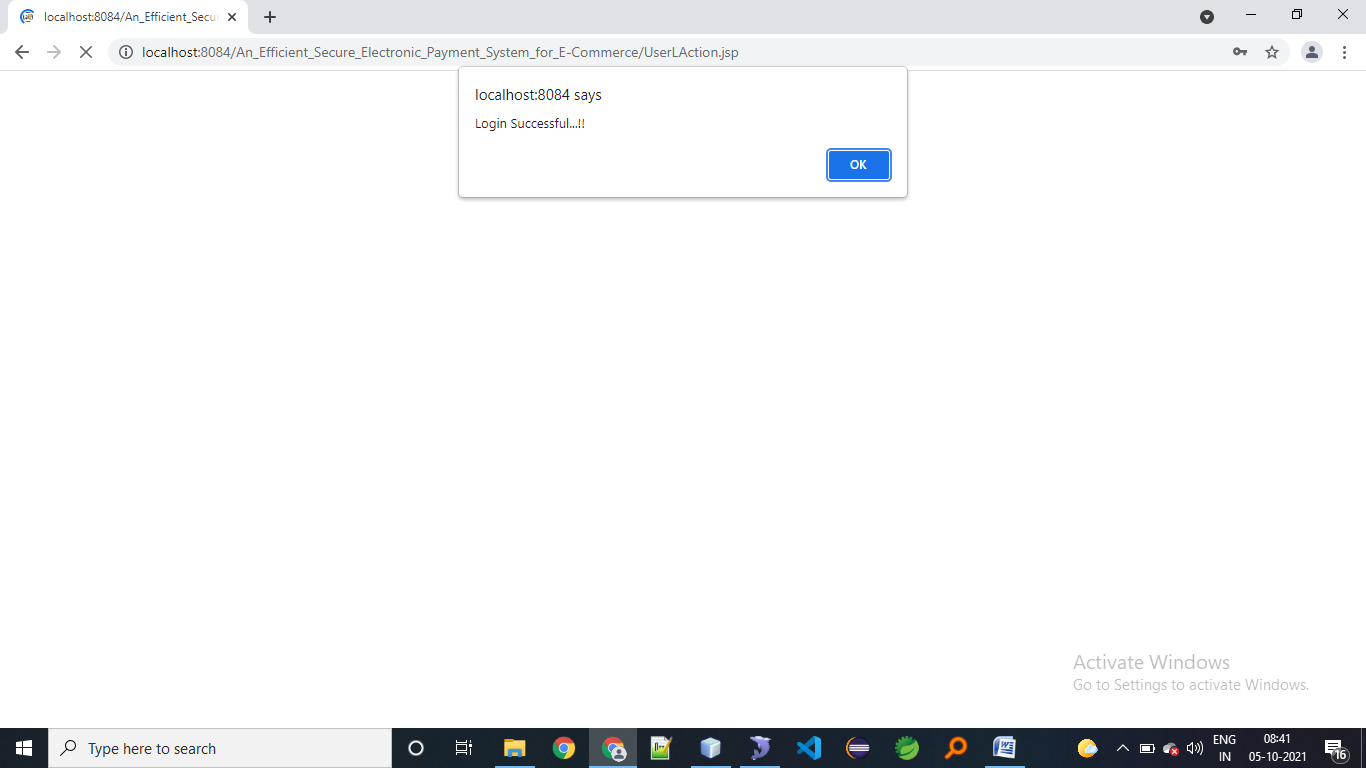
**USER LOGIN**



**USER REGISTER**



**LOGIN STATUS**



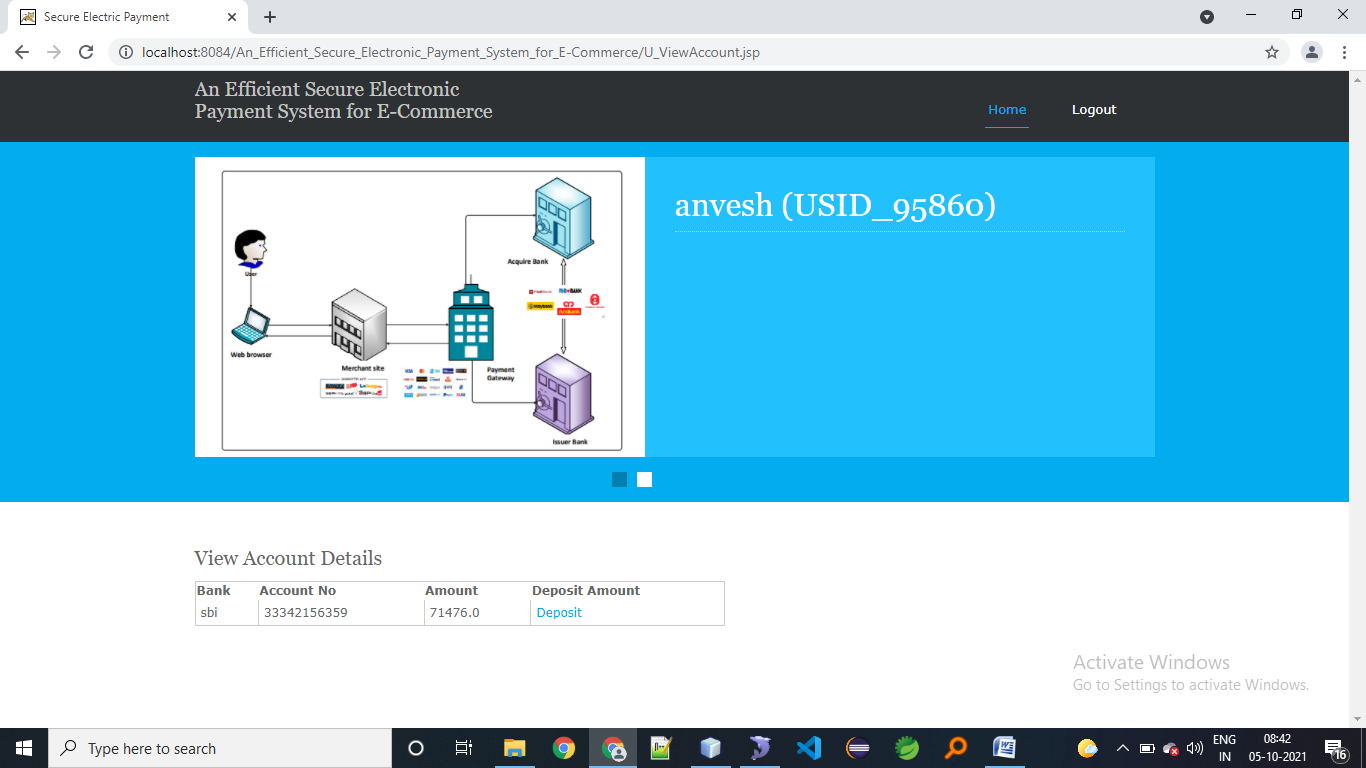
**USER HOME PAGE**



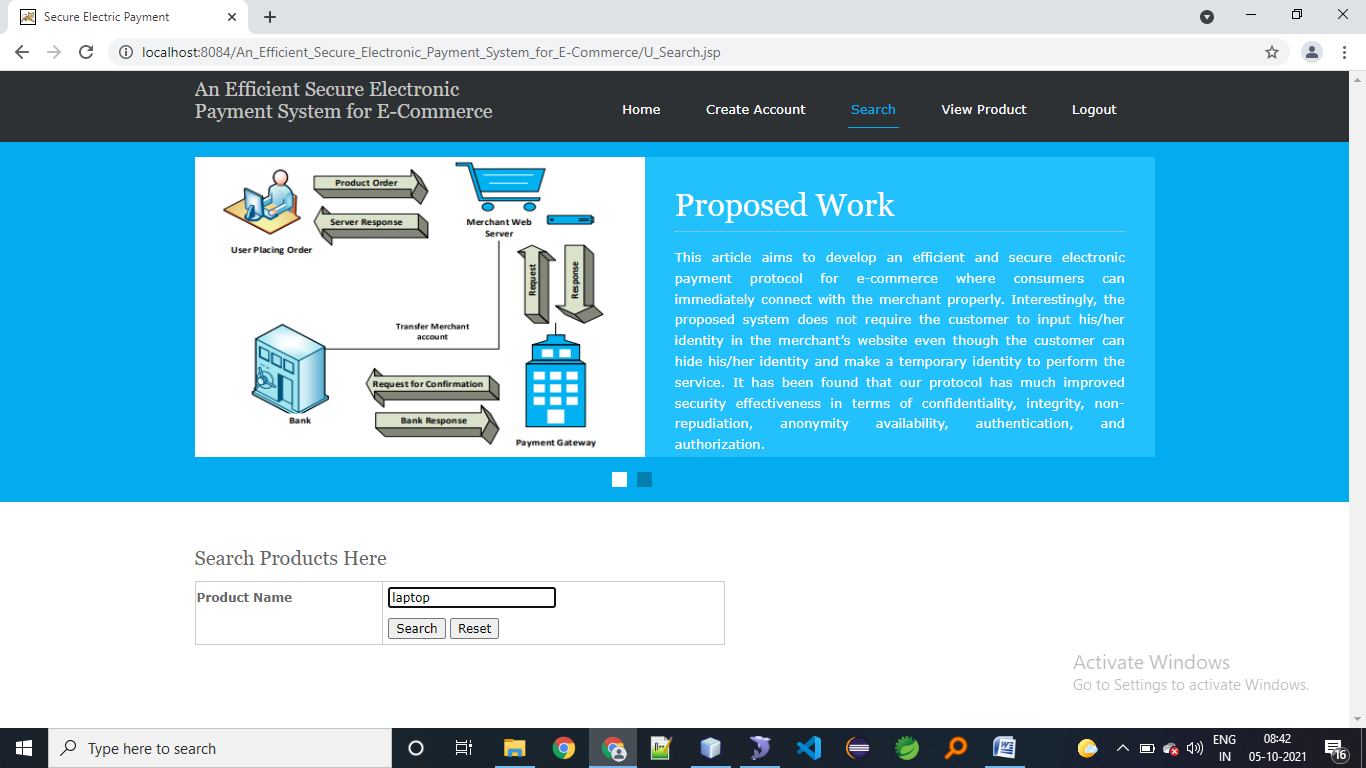
**CREATE ACCOUNT**



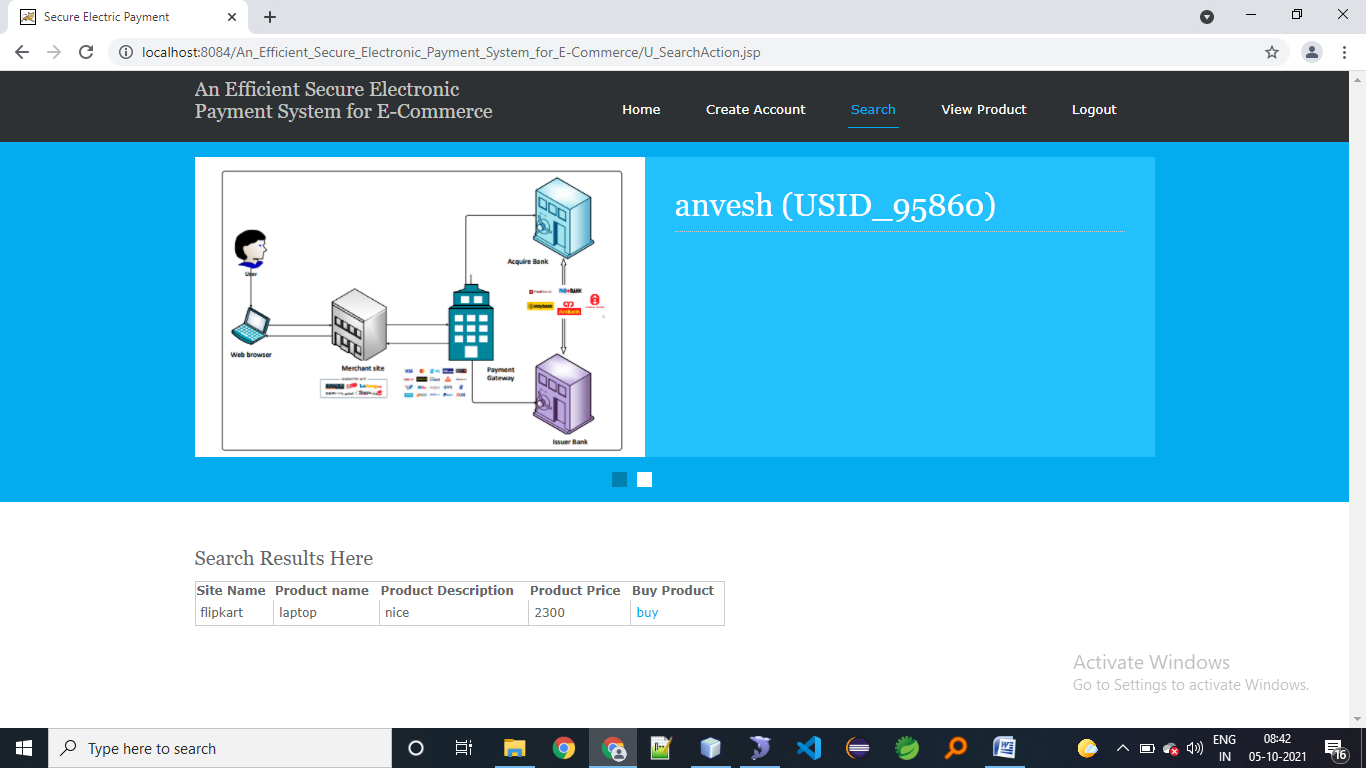
**VIEW ACCOUNT**



**Search product**



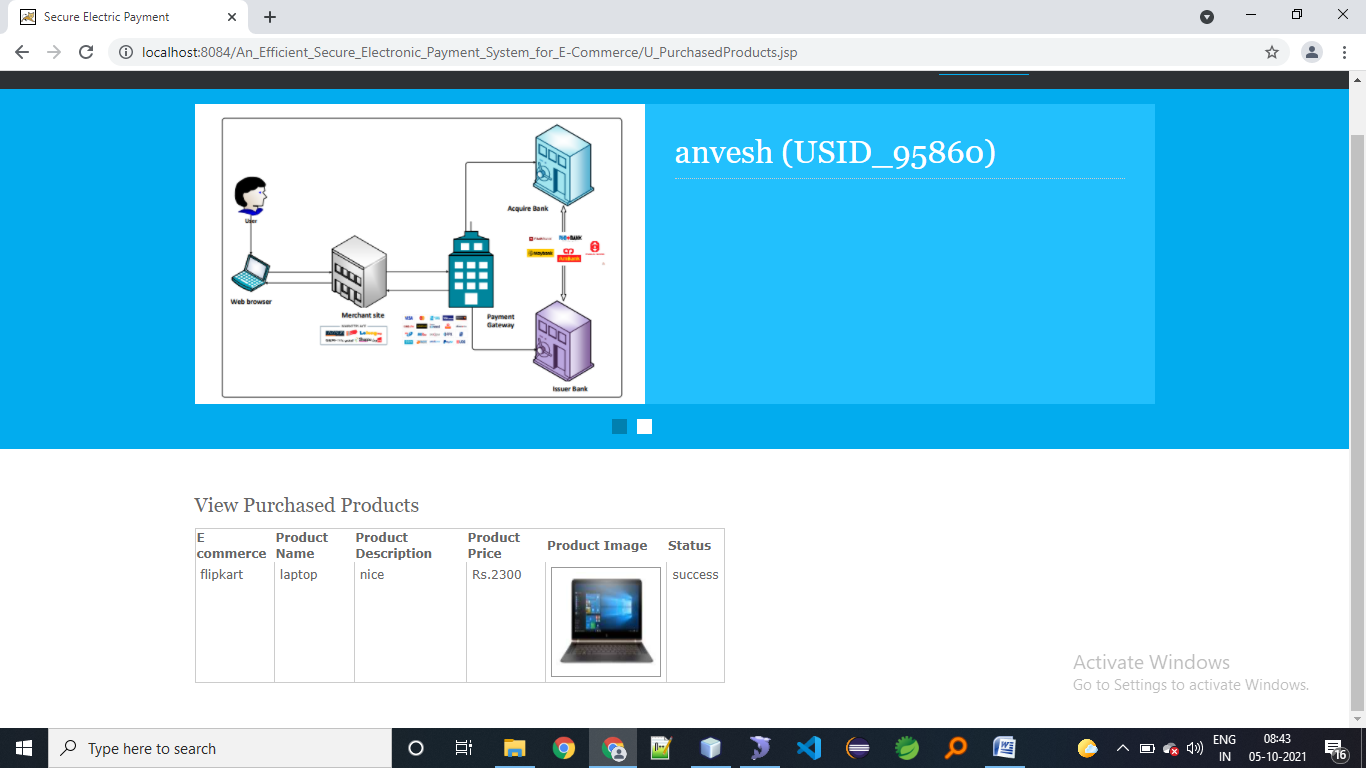
**Search action**



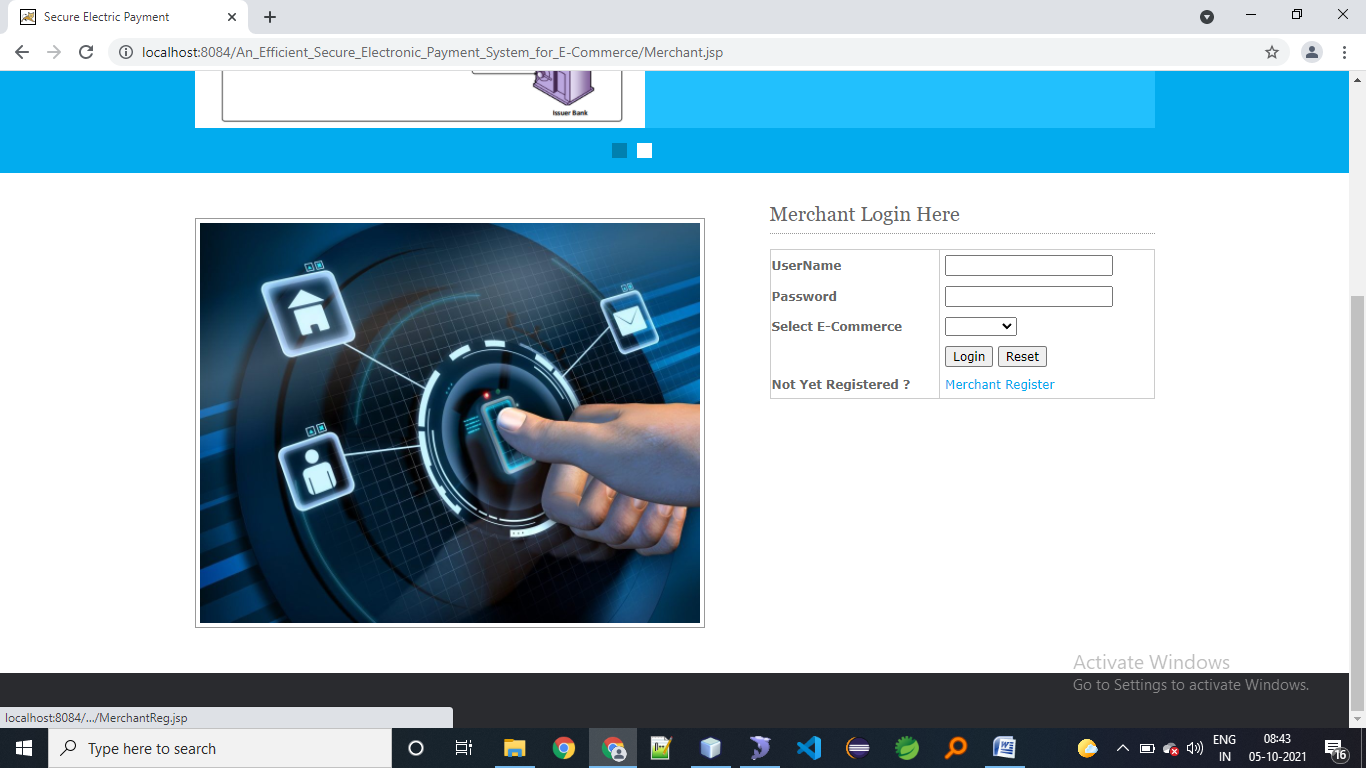
**Payment gateway**



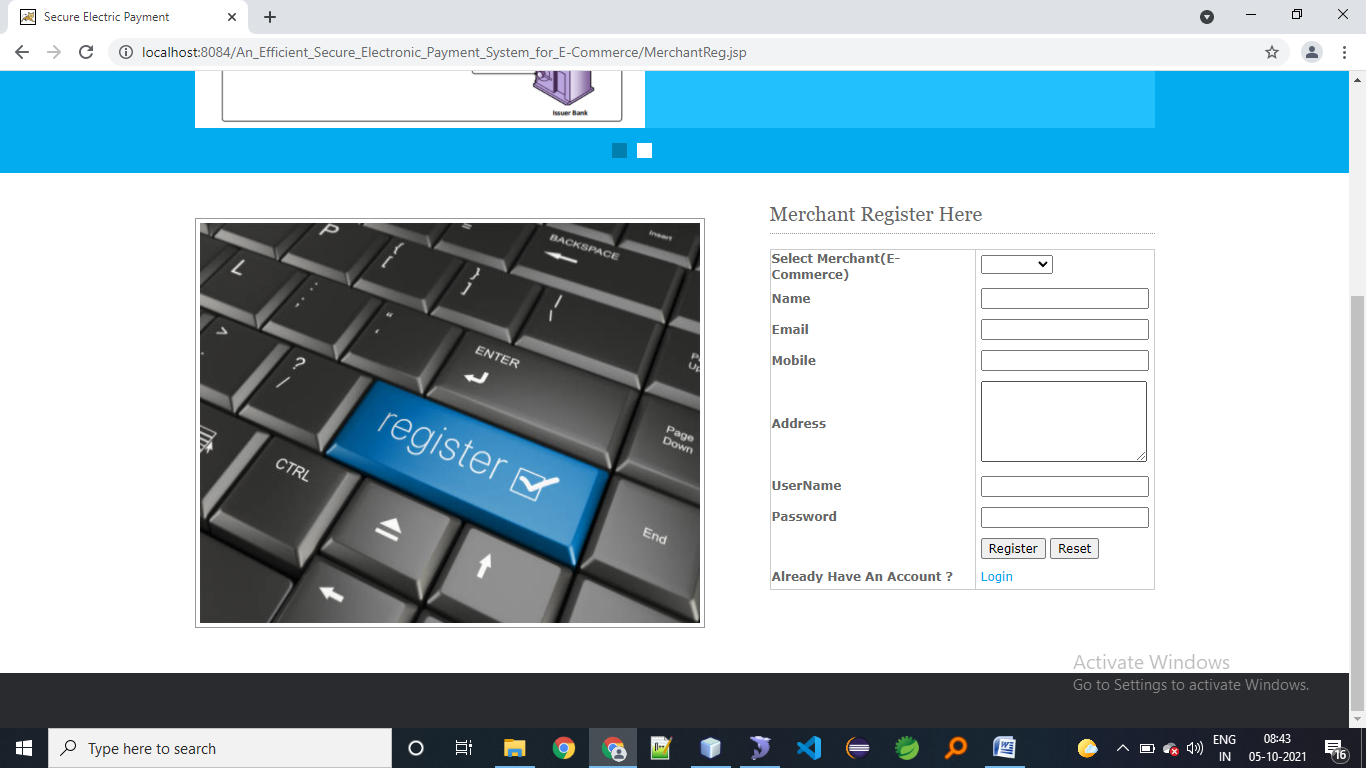
**View purchased products**



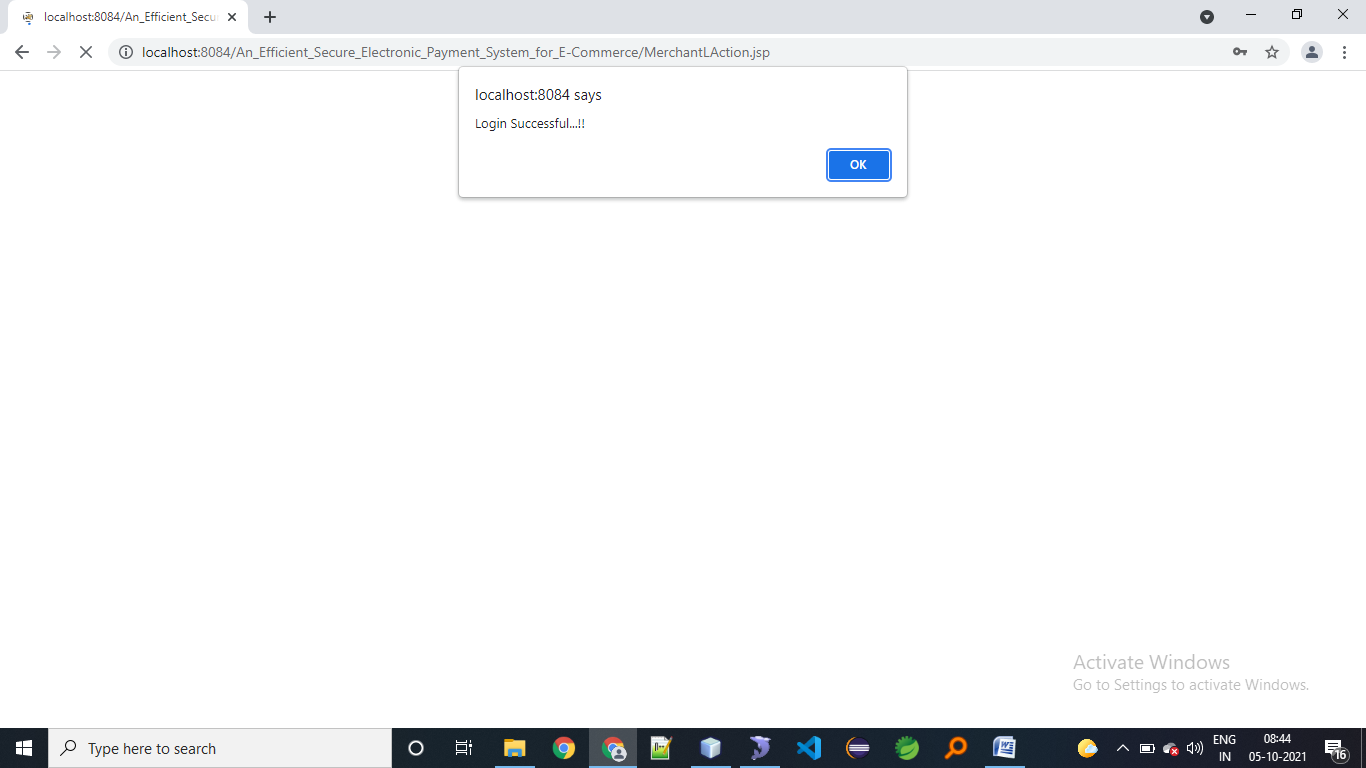
**Merchant login**



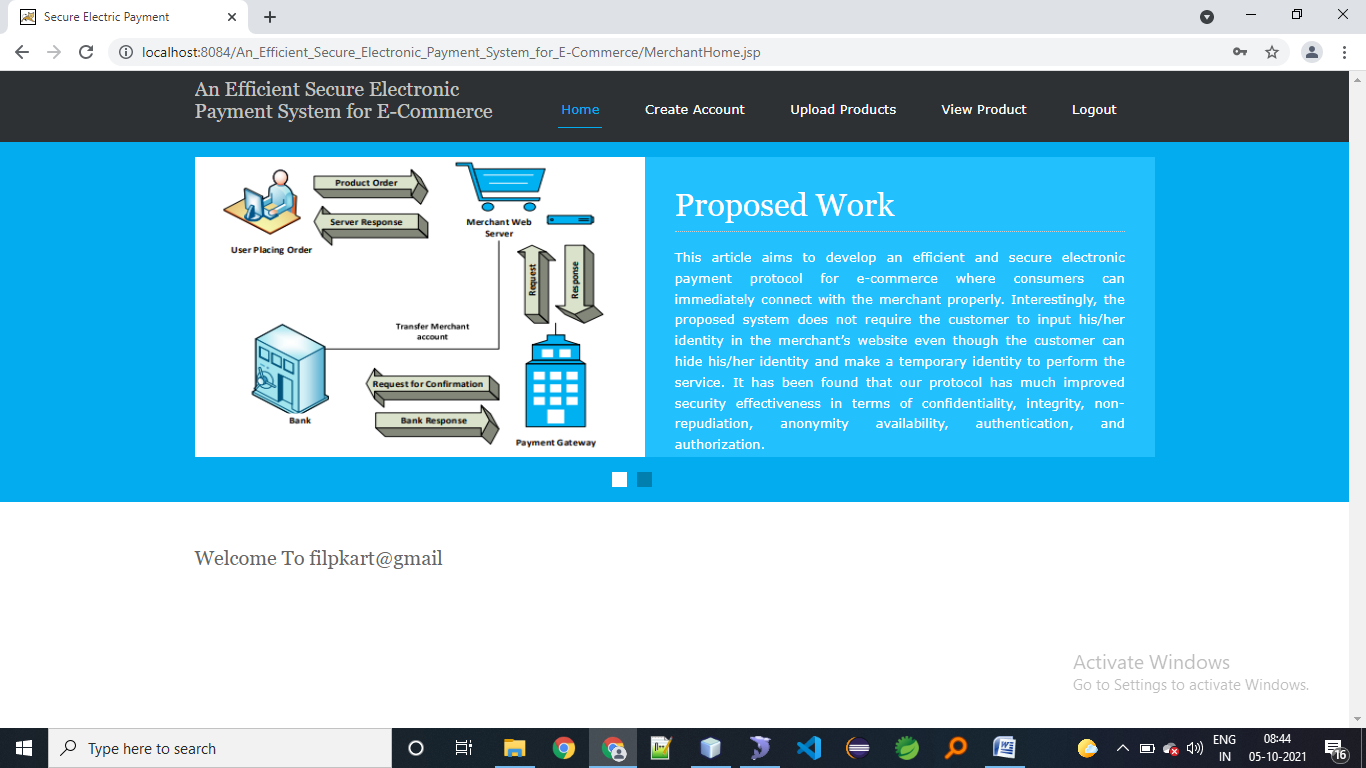
**Merchant register**



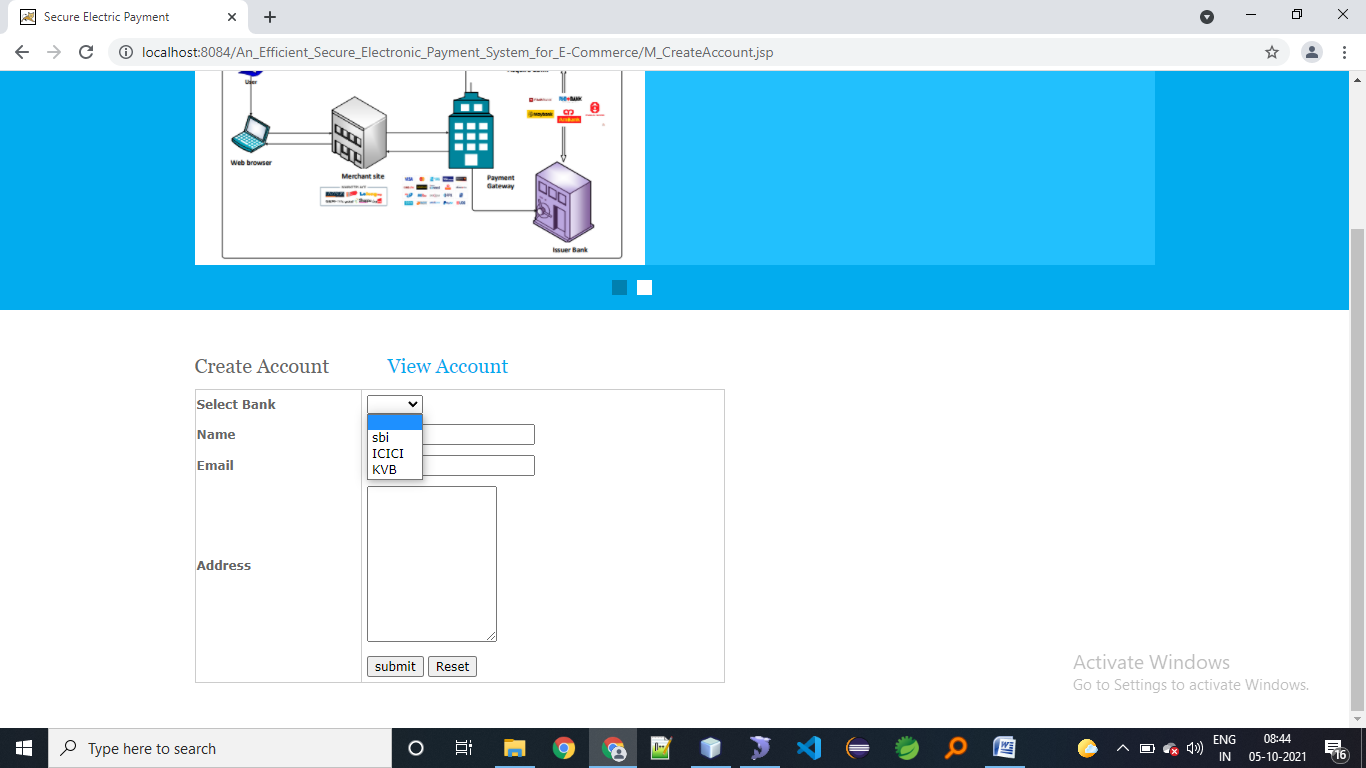
**Merchant login status**



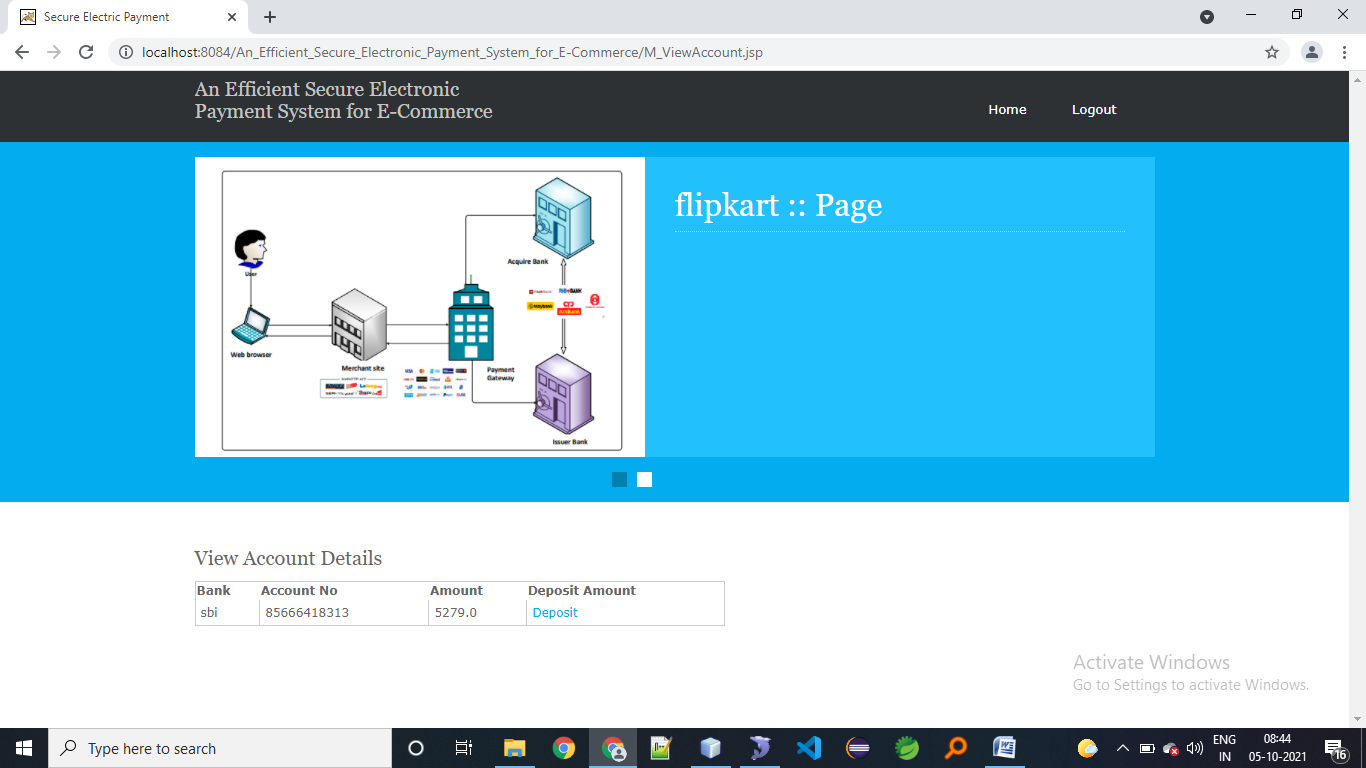
**Merchant home**



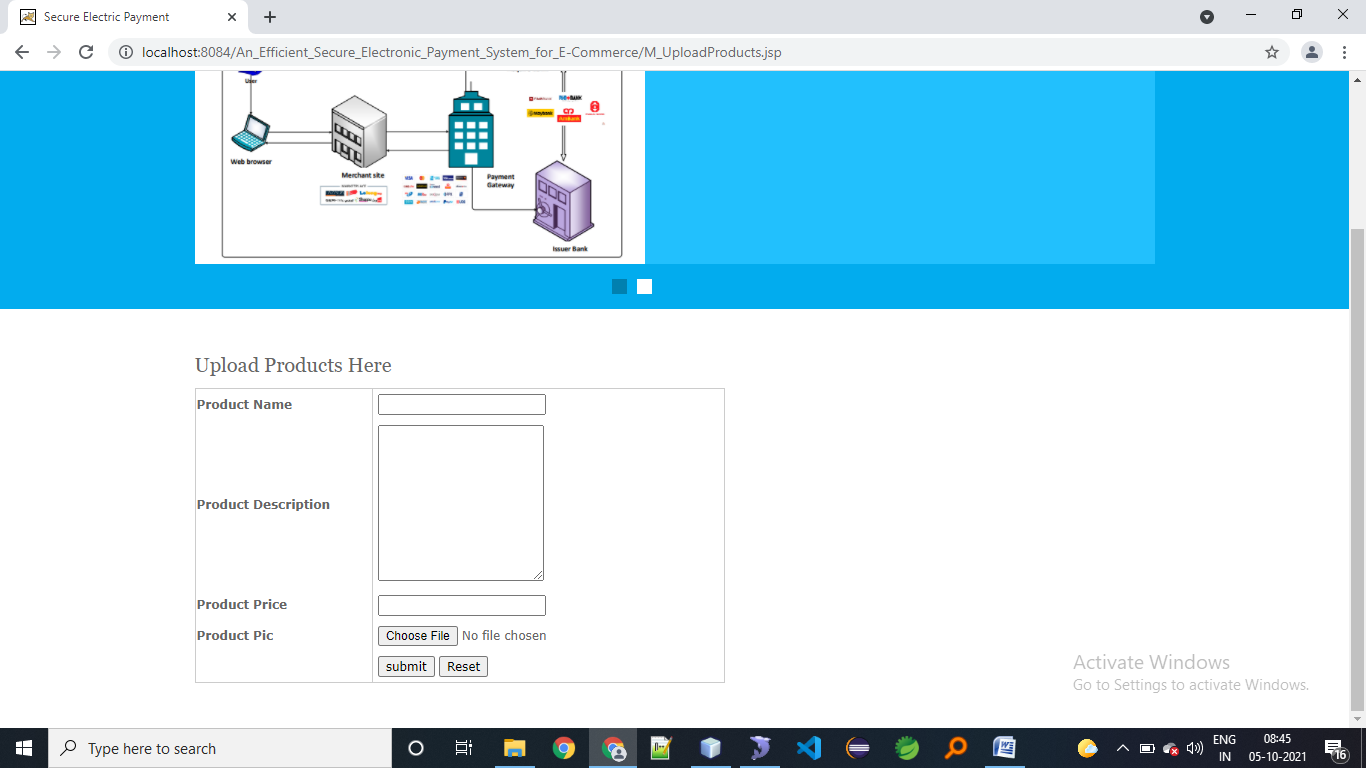
**Merchant create account**



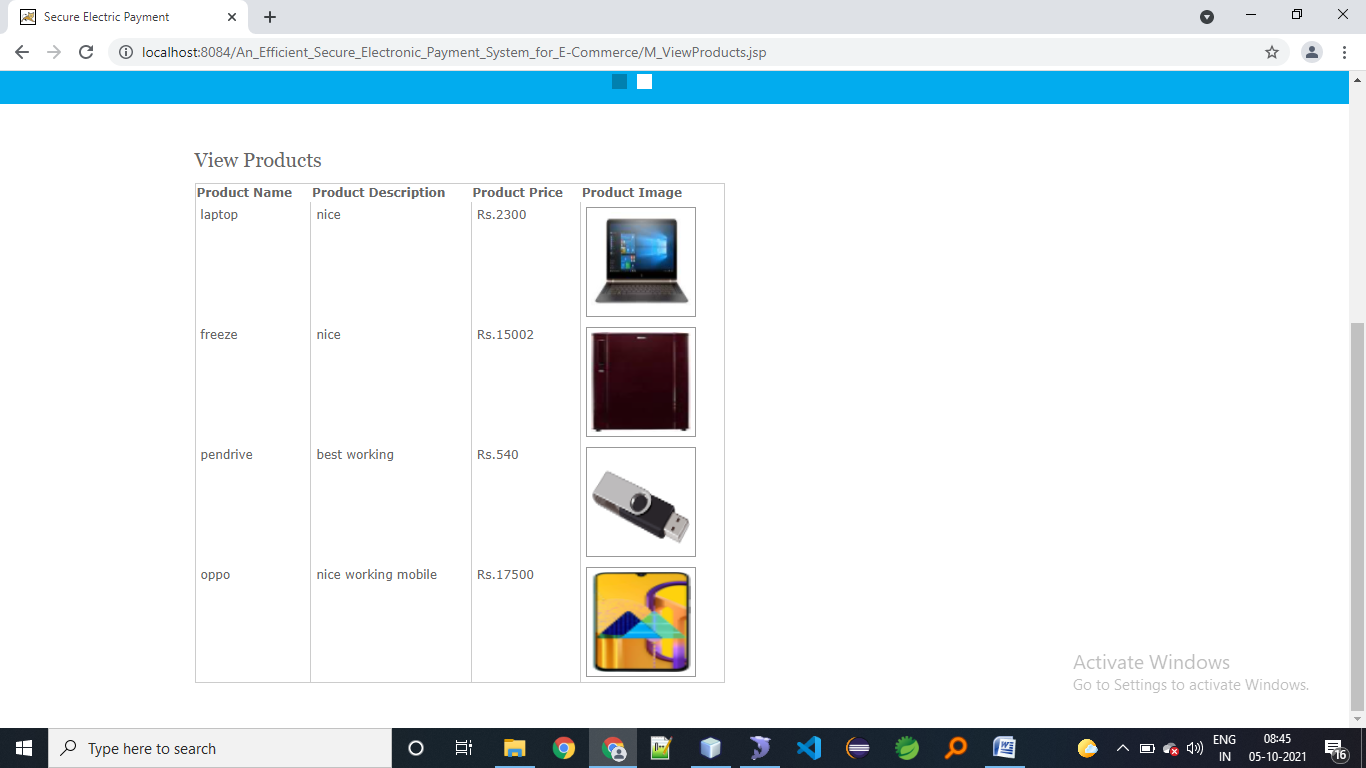
**View merchant account details**



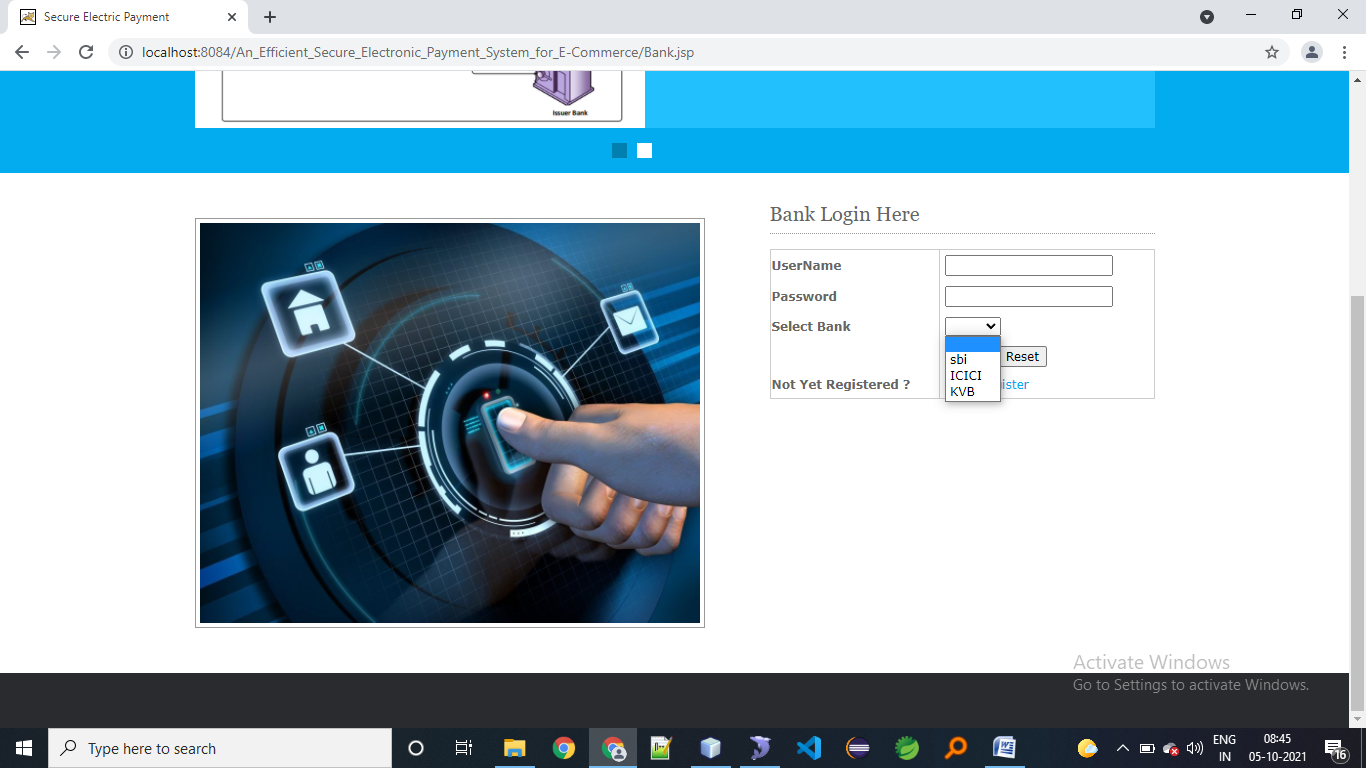
**Upload products**



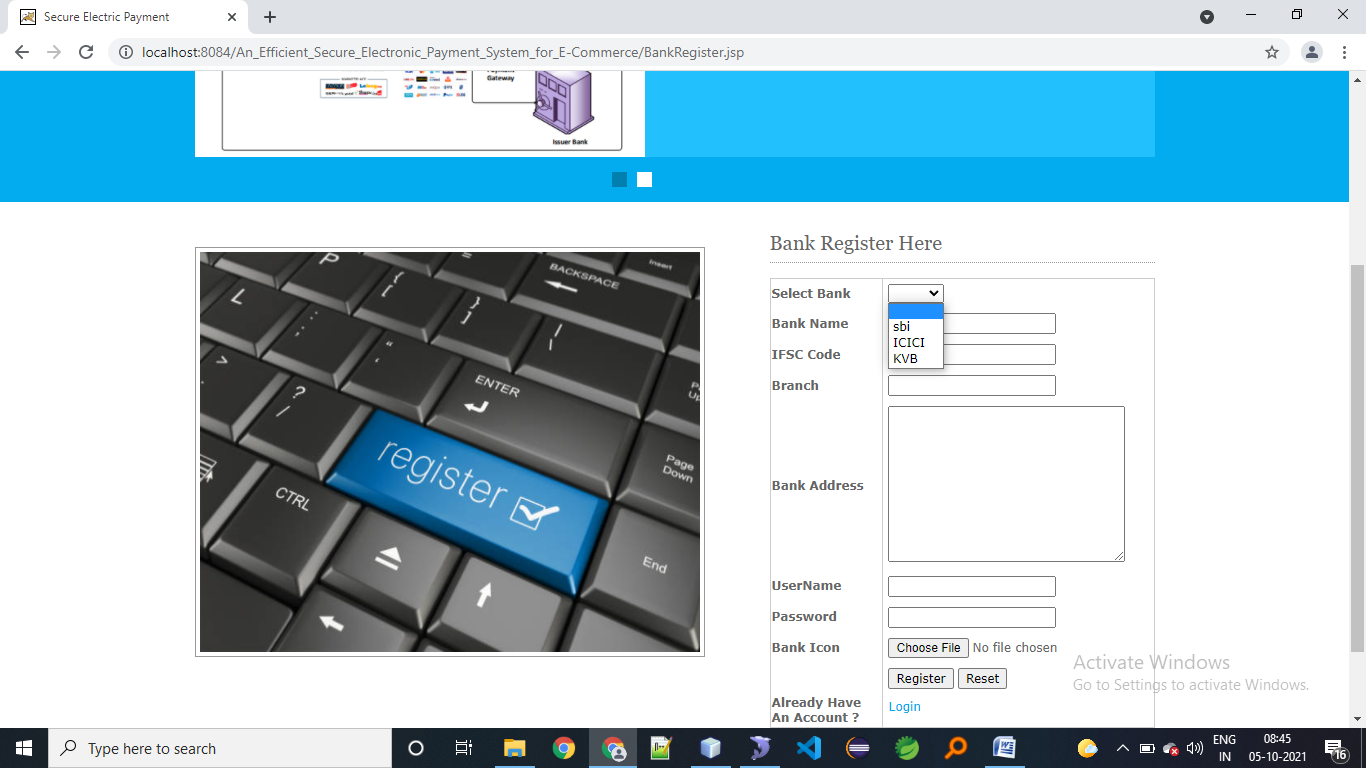
**View uploads**



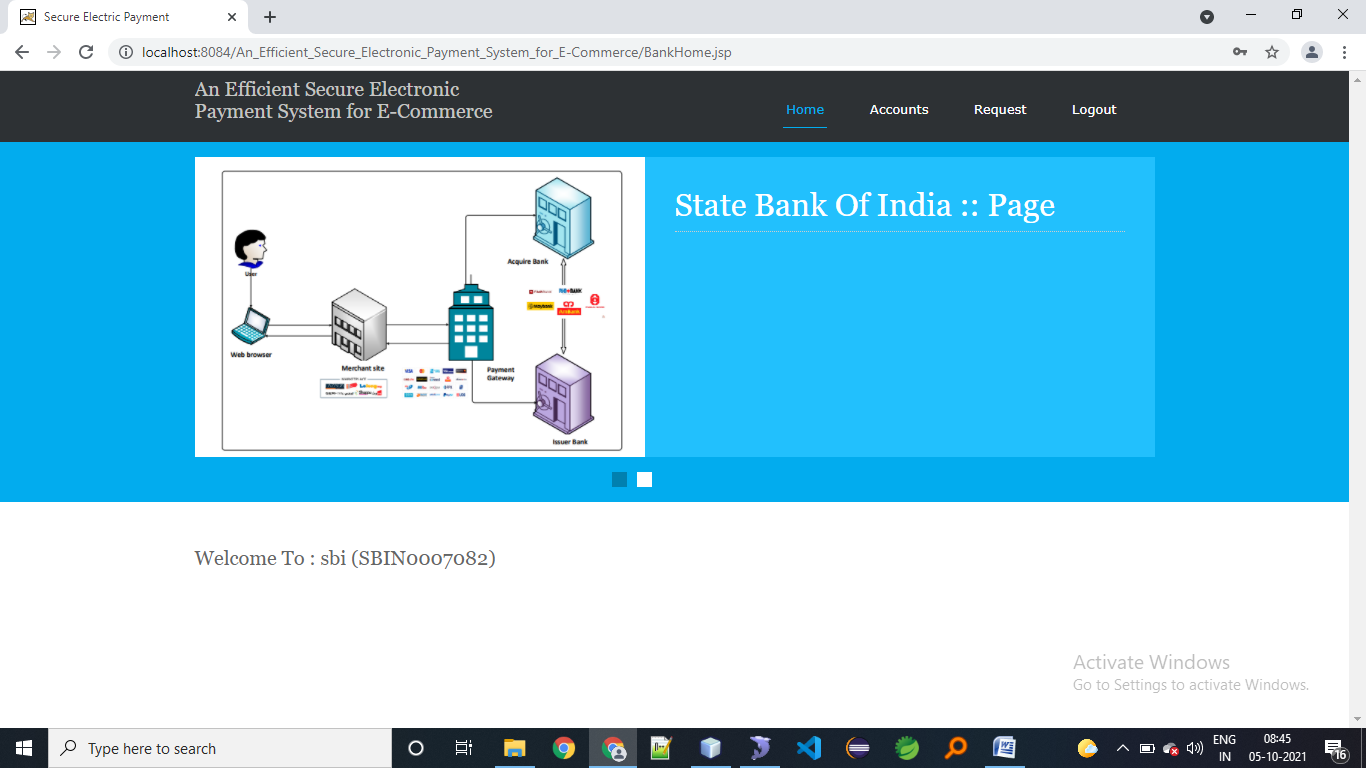
**Bank login**



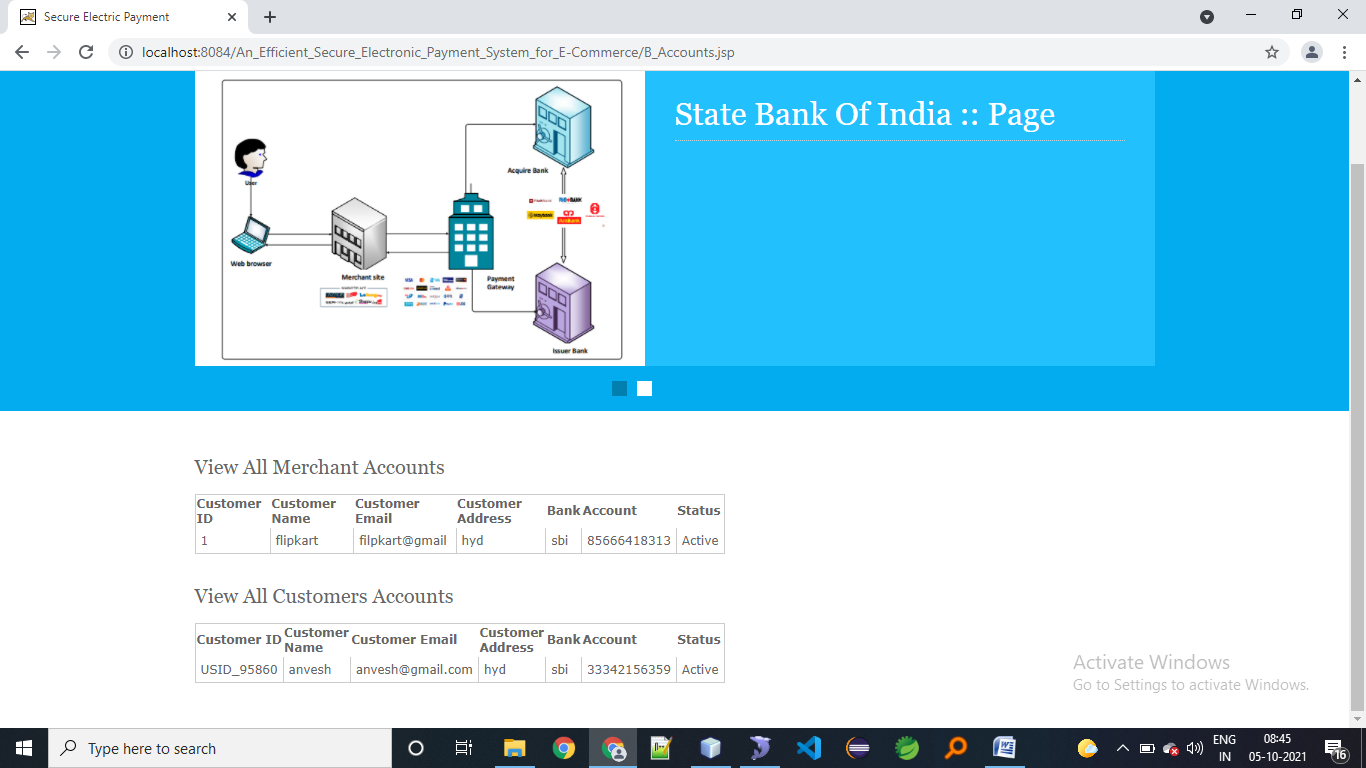
**Bank registration**



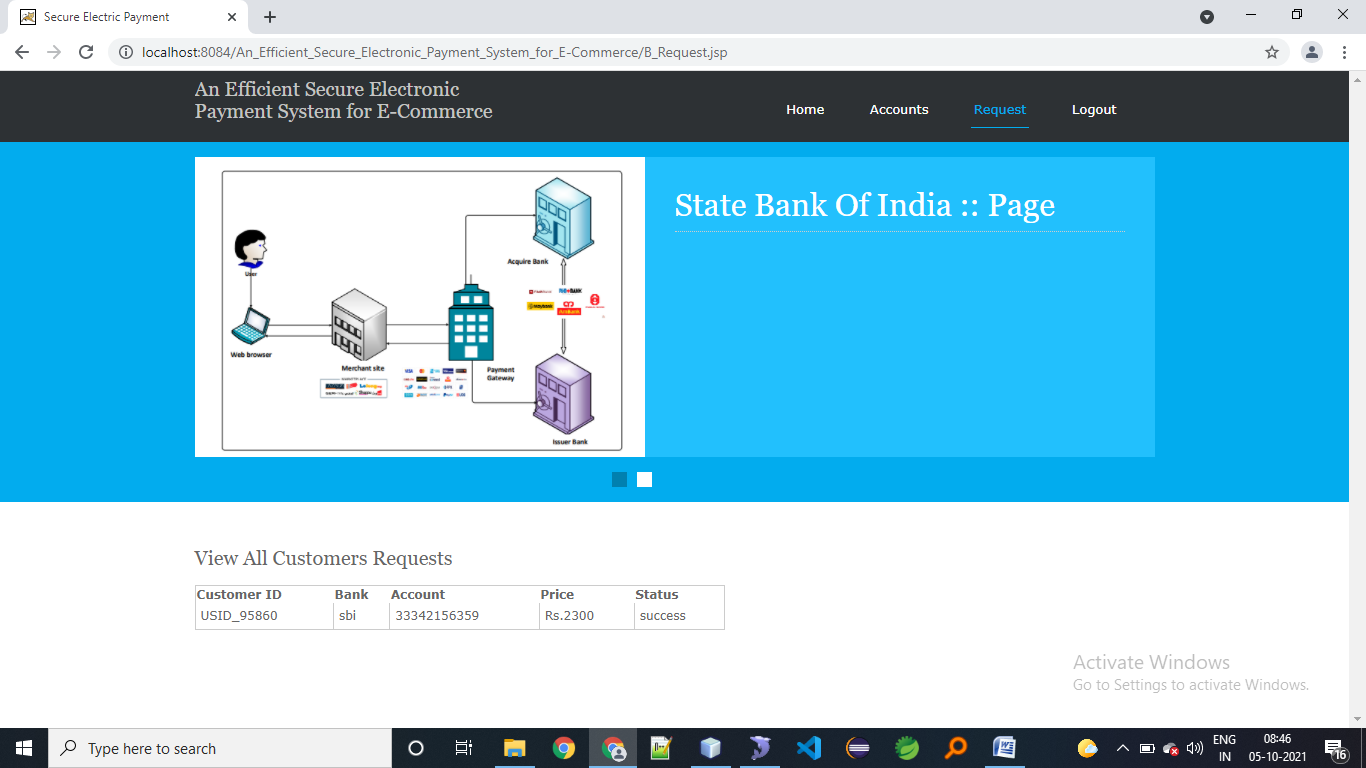
**Bank home page**



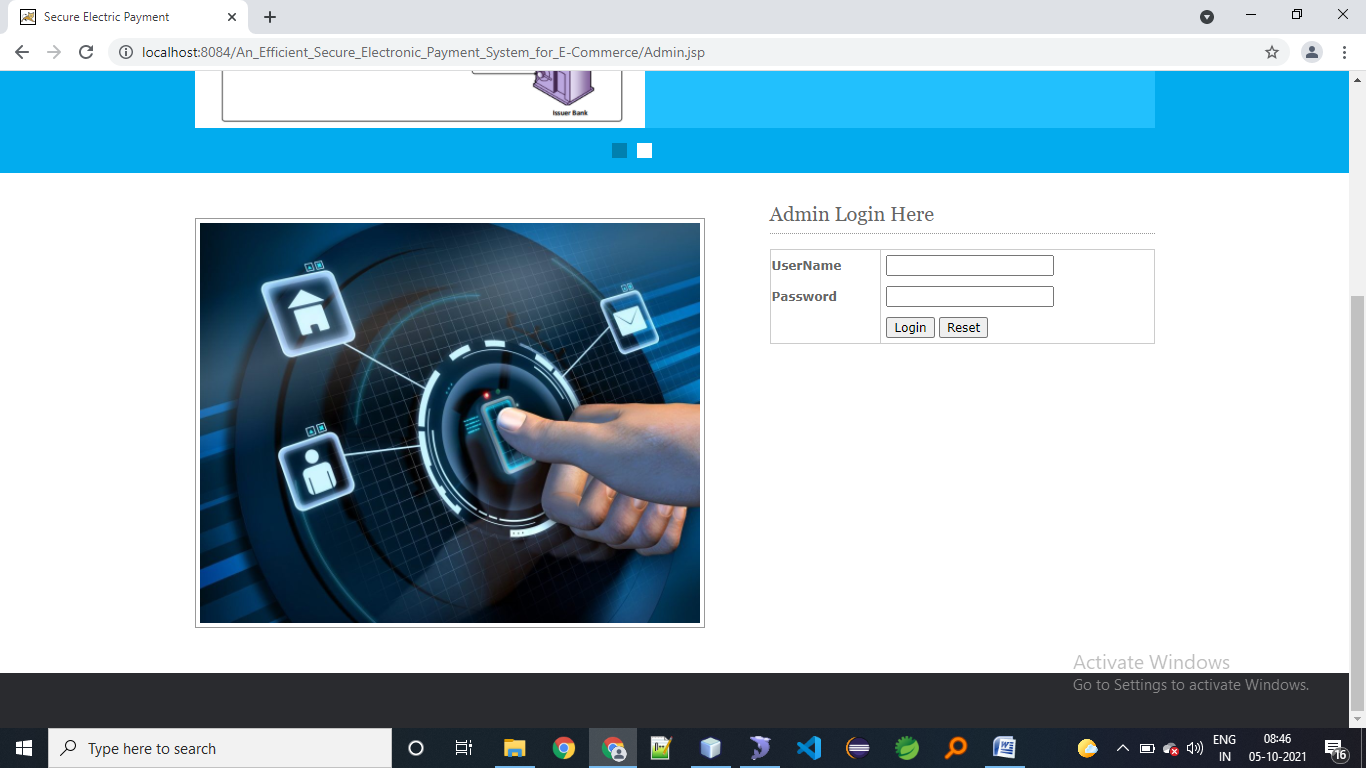
**Accounts**



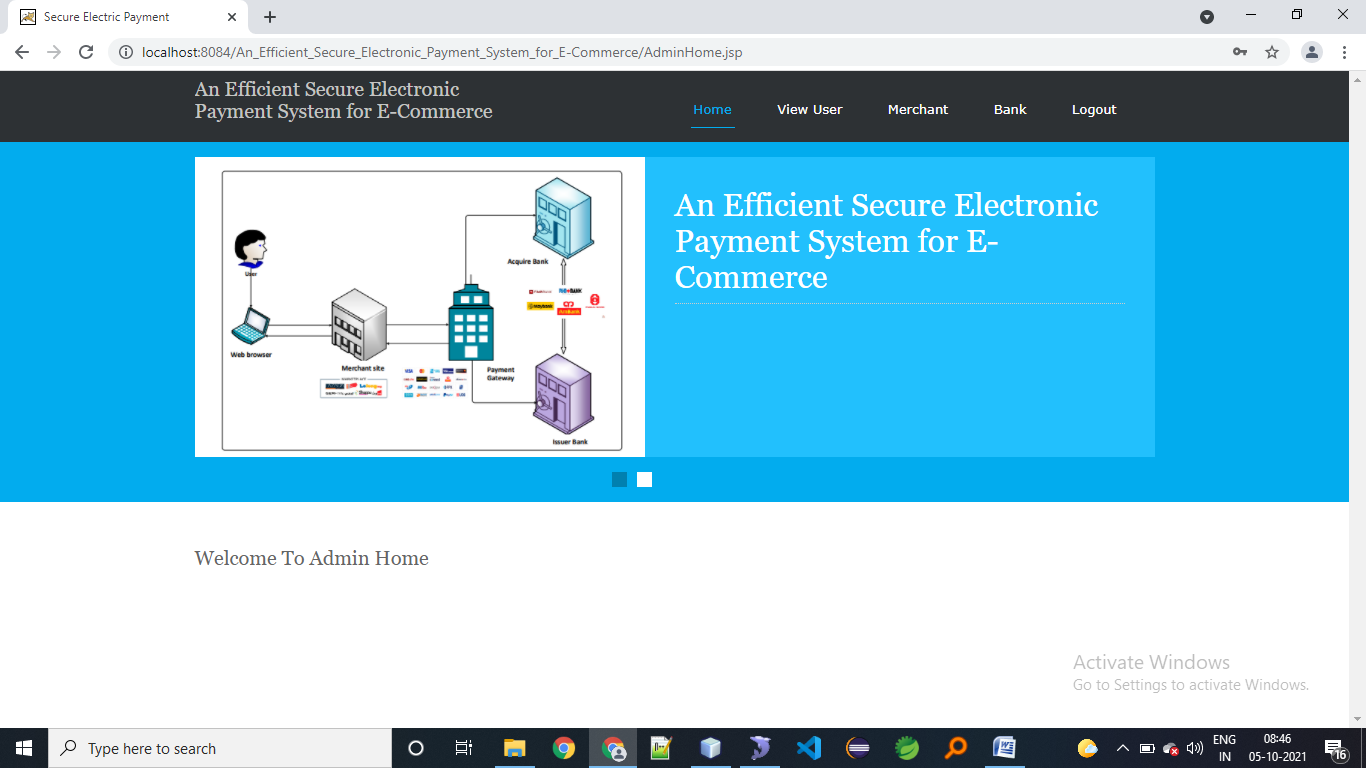
**Transfer request**



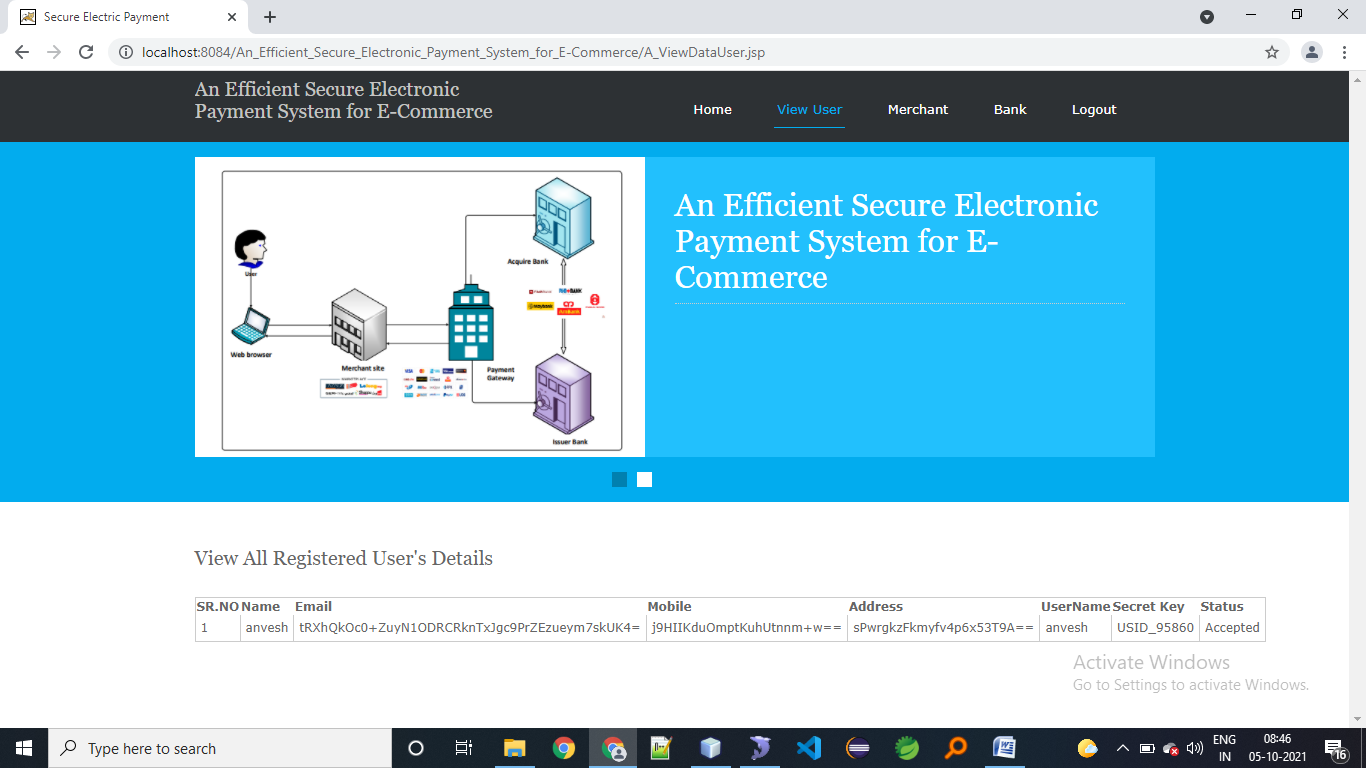
**Admin login**



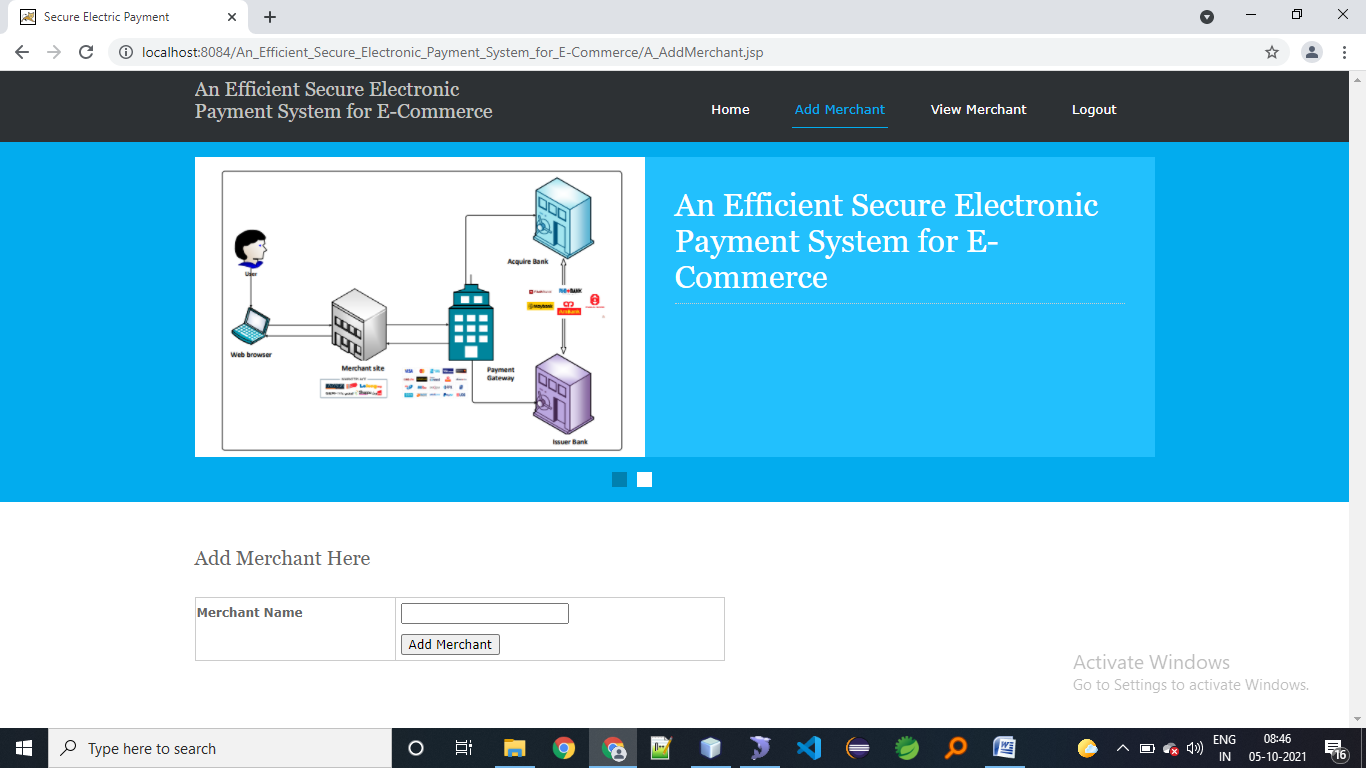
**Admin home page**



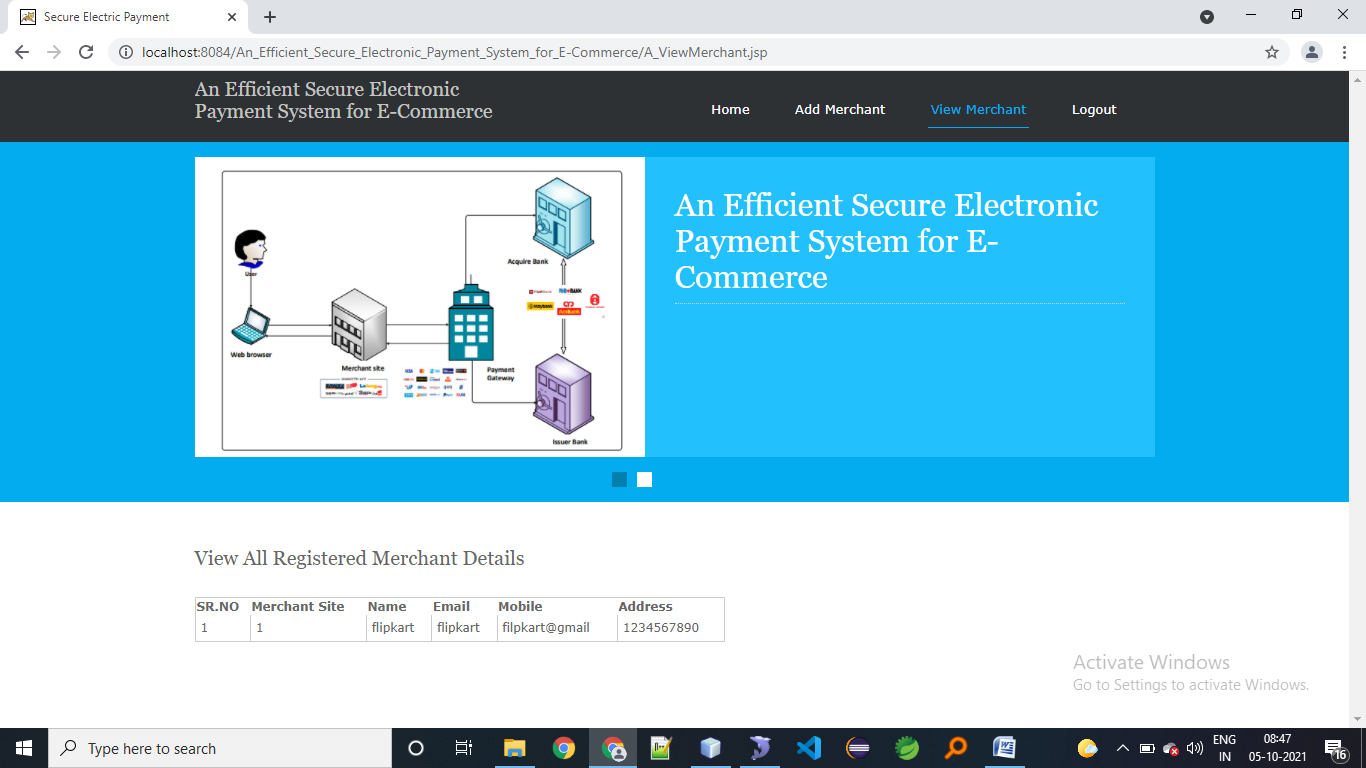
**View users details by admin**

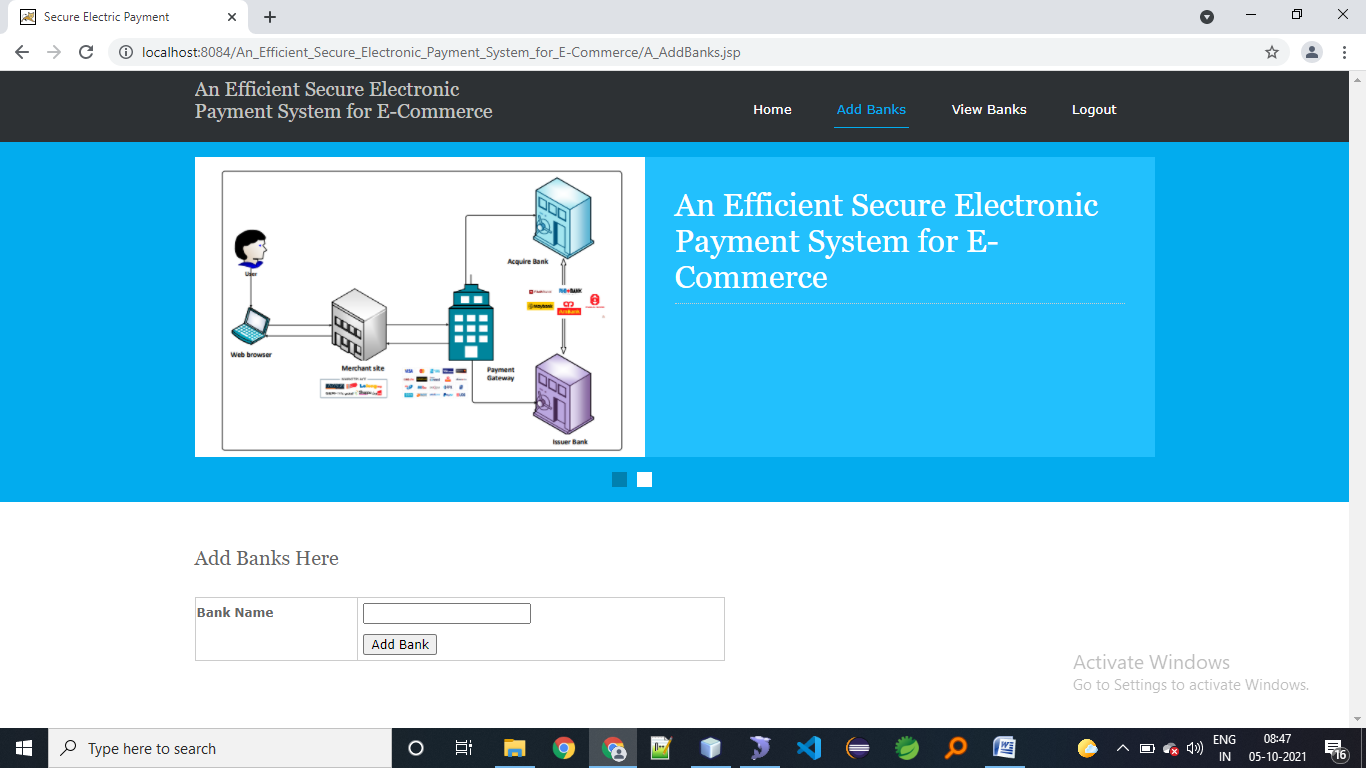


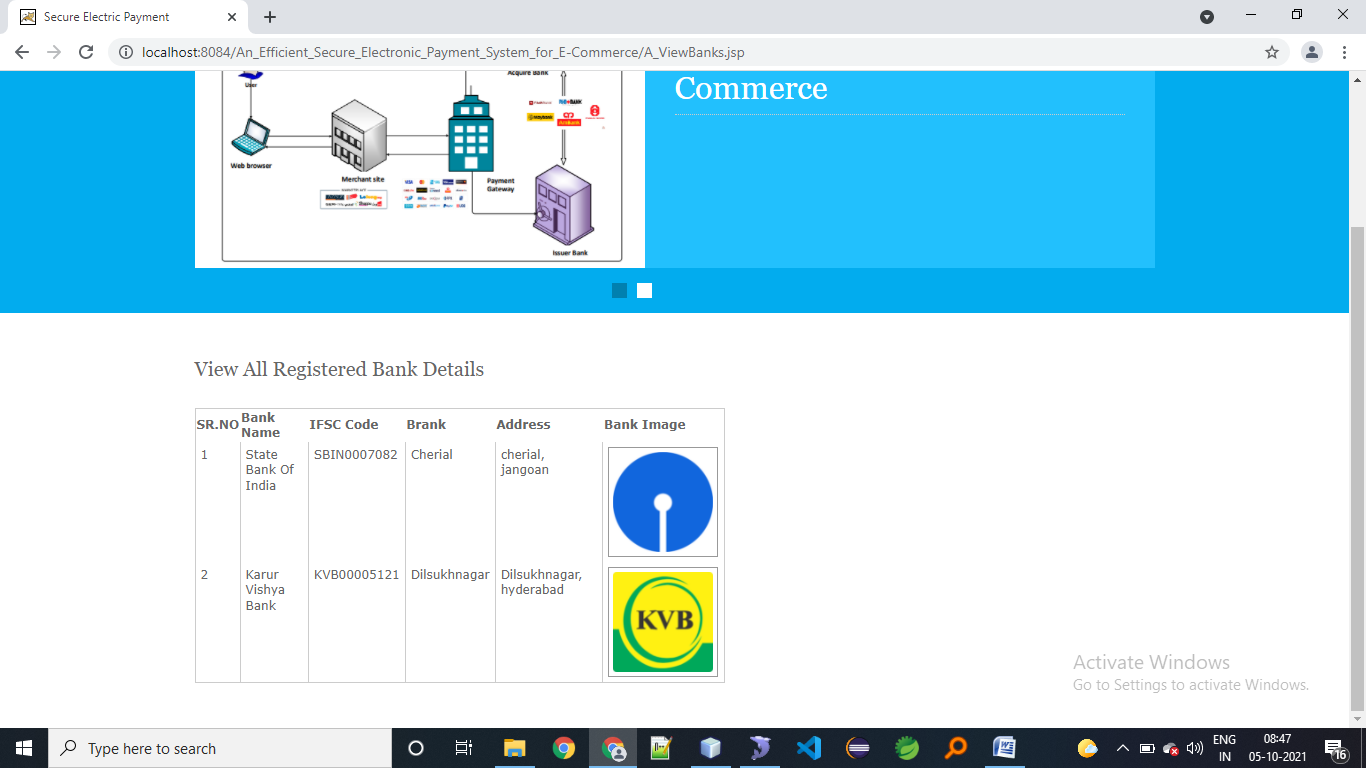
**Add merchant(ecommerce sites)**



**View registered merchants**

**add banks by admin**

**view all registered banks**



**SYSTEM TESTING**

**7.1 INTRODUCTION TO TESTING**

**Introduction to Testing:**

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

**7.2 TESTING IN STRATEGIES**

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**Unit Testing:**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

**Each module can be tested using the following two Strategies:**

**Black Box Testing:**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structure or external database access
* Performance errors
* Initialization and termination errors.

In this testing only the output is checked for correctness.

The logical flow of the data is not checked.

**White Box testing :**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

* Guarantee that all independent paths have been Executed.
* Execute all logical decisions on their true and false Sides.
* Execute all loops at their boundaries and within their operational bounds
* Execute internal data structures to ensure their validity.

**Integrating Testing :**

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

**System Testing:**

Involves in-house testing of the entire system before delivery to the user. It's aim is to satisfy the user the system meets all requirements of the client's specifications.

**Acceptance Testing :**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

#### Test Approach :

**Testing can be done in two ways:**

* Bottom up approach
* Top down approach

**Bottom up Approach:**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

**Top down approach:**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

**Validation:**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed

**SYSTEM SECURITY**

**8.1 INTRODUCTION**

System Security:

Setting Up Authentication for Web Applications

Introduction:

To configure authentication for a Web Application, use the <login-config> element of the web.xml deployment descriptor. In this element you define the security realm containing the user credentials, the method of authentication, and the location of resources for authentication.

**8.2 SECURITY IN SOFTWARE**

To set up authentication for Web Applications:

1. Open the web.xml deployment descriptor in a text editor or use the Administration Console. Specify the authentication method using the <auth-method> element. The available options are:

**BASIC**

Basic authentication uses the Web Browser to display a username/password dialog box. This username and password is authenticated against the realm.

**FORM**

Form-based authentication requires that you return an HTML form containing the username and password. The fields returned from the form elements must be: j\_username and j\_password, and the action attribute must be j\_security\_check. Here is an example of the HTML coding for using FORM authentication:

<form method="POST" action="j\_security\_check">

   <input type="text" name="j\_username">  
   <input type="password" name="j\_password">

</form>

The resource used to generate the HTML form may be an HTML page, a JSP, or a servlet. You define this resource with the <form-login-page> element.

The HTTP session object is created when the login page is served. Therefore, the session.isNew () method returns FALSE when called from pages served after successful authentication.

**CONCLUSION**

E-commerce has extremely enhanced in popularity over the last decades, and, in methods, it is changing typical payment methods right into online. With the increasing popularity of e-commerce, the market for digital payments has exploded in the last decades, and payment in e-commerce, particularly mobile payment, is currently extremely preferred and plays a growing role. The principal issue is a better requirement for a secure payment system and online authentication on the client side and the Web server side both in growth and in the development of e-commerce. In this research, we suggested an efficient, secure electronic payment system for e-commerce. We introduced a comparison between our suggested framework and the other three existing systems, which use RSA and DES to secure debit/credit card details and keep them anonymous. Most of the clients want an e-commerce program, as there are many advantages. Clients need such a secure system, because it satisfies all specifications and is a sufficient system. We proposed a secure electronic payment system for e-commerce environments on the basis of these requirements. In our proposed method, the transaction gateway functions as a proxy to communicate between the client/merchant and the bank. The security analysis demonstrated that the proposed plan has better protection effectiveness in terms of confidentiality, non-repudiation, integrity, availability, and anonymity. The extension of this article will focus on the utilization of our proposed framework in real-world applications by proving its ability to avoid various attacks and determine the time necessary for electronic payment.

**Bibliography**

**REFERENCES**

1. Miva. The History of Ecommerce: How Did It All Begin?—Miva Blog. Available online: https://www.miva. com/blog/the-history-of-ecommerce-how-did-it-all-begin/ (accessed on 16 June 2020).

2. Alam, S.S.; Ali, M.H.; Omar, N.A.; Hussain, W.M.H.W. Customer satisfaction in online shopping in growing markets: An empirical study. Int. J. Asian Bus. Inf. Manag. 2020, 11, 78–91. [CrossRef]

3. Noor Ardiansah, M.; Chariri, A.; Rahardja, S.; Udin, U. The effect of electronic payments security on e-commerce consumer perception: An extended model of technology acceptance. Manag. Sci. Lett. 2020, 10, 1473–1480. [CrossRef]

4. Soare, C.A. Internet Banking Two-Factor Authentication using Smartphones. J. Mob. Embed. Distrib. Syst. 2012, 4, 12–18.

5. Satar, N.S.M.; Dastane, O.; Ma’arif, M.Y. Customer value proposition for E-Commerce: A case study approach. Int. J. Adv. Comput. Sci. Appl. 2019, 10, 454–458. [CrossRef]

6. Narwal, B. Security Analysis and Verification of Authenticated Mobile Payment Protocols. In Proceedings of the 4th International Conference on Information Systems and Computer Networks (ISCON 2019), Mathura, India, 21–22 November 2019; pp. 202–207. [CrossRef]

7. Bezhovski, Z. The Future of the Mobile Payment as Electronic Payment System. Eur. J. Bus. Manag. 2016, 8, 2222–2839.

8. Masihuddin, M.; Islam Khan, B.U.; Islam Mattoo, M.M.U.; Olanrewaju, R.F. A Survey on E-Payment Systems: Elements, Adoption, Architecture, Challenges and Security Concepts. Indian J. Sci. Technol. 2017, 10, 1–19. [CrossRef]

9. Liao, X.; Ahmad, K. Factors Affecting Customers Satisfaction on System Quality for E-Commerce. In Proceedings of the 2019 International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 9–10 July 2019; pp. 360–364. [CrossRef]

10. Mazumder, F.K.; Jahan, I.; Das, U.K. Security in Electronic Payment Transaction. Int. J. Sci. Eng. Res. 2015, 6, 955–960.