

BLOCKISH SCAM EXPOSURE USING AUTOMATION LEARNING

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

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

By

Name:

CHANDRU A

(Reg .no. 38119004)

SIVARAMAN R

(Reg .no. 38119010)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR
NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MAY- 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTEMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHANDRU A (38119004), SIVARAMAN R (38119010)** who carried out the project entitled **“BLOCKISH SCAM EXPOSURE USING AUTOMATION LEARNING”** under my supervision from Dec 2021 to May 2022.

INTERNAL GUIDE

EXTERNAL GUIDE

HEAD OF DEPARTMENT

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

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

Submitted for Viva Voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I CHANDRU A,SIVARAMAN R hereby declare that the Project Report entitled **“BLOCKISH SCAM EXPOSURE USING AUTOMATION LEARNING”** done by me under the guidance of **Dr.J.ALBERT MAYAN,M.E.ph.D., and Dr.L.LAKSHMANAN,M.E.,Ph.D., and Dr.S.VIGNESHWARI, M.E.,Ph.D.,**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:

SIGNATURE OF THE CANDIDATE

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 of Computer Science and Engineering for providing me necessary support and details at the right time during the progressivereviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J.Albert Mayan 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 Computer Science and engineering** who were helpfull in many ways for the completion of the project.

ABSTRACT:

Fraudulent banking operations can cause huge losses to the bank and further affect the economy negatively. What if Block chain Technology and Machine Learning could be combined to detect suspicious banking activity and stop transactions at the source? That is what this paper aims to do is to implemented the block chain is to securely store transaction history, For quick and efficient detection of outliers, which indicate suspicious activity by algorithm host by machine learning. Even a single fraudulent action has a negative impact on the economy and affects all citizens negatively. This is why we must take a stand to prevent fraudulent banking activities. One method of doing this is to employ Artificial Intelligence, particularly Machine Learning, in the banking sector. We then combine it with Block chain technology to ensure secure banking transactions in the future. This will make fraud detection quick, easy and more efficient. The Private Permissioned Block chain contains all data regarding the transactions and can be retrieved in real time. A reinforced K-means clustering algorithm is applied to the block chain to detect discrepancies and point out fraudulent transactions. The Apache ignite Platform offers powerful computing that enables the process to occur in real time. Access control tends to be more flexible, and is easier to implement. When large data is transmitted to the cloud, a security issue emerges. Most organizations would not want their data in the hands of another organization, thus the need for encryption. When the data is transmitted, it is not encrypted because the approaches used to transmit the data require that the data be decrypted. This exposes the data to attacks. Confidentiality breach is the biggest threat to big data thus the encryption could be used as the primary big data protection technique.

CONTENTS		
1.	CHAPTER-1 1.1. INTRODUCTION TO PROJECT 1.2. PURPOSE OF THE PROJECT	 2 2
2.	CHAPTER-2 2.1. HARDWARE & SOFTWARE REQUIRMENT 2.2. INPUT AND OUTPUT 2.3. INPUT DESIGN 2.4. EXISTING STSTEM 2.5. PROPOSED SYSTEM 2.6. LITERATURE SURVEY	 4 5 5 6 7
3.	CHAPTER-3 3.1.SOFTWARE REQUIREMENT SPECIFICATIONS 3.1.1.FUNCTIONAL REQUIREMENTS 3.1.2.NON-FUNCTIONAL REQUIREMENTS 3.1.3PERFORMANCE REQUIREMENTS 3.2.FEASIBILITY REPORT 3.2.1. TECHNICAL FEASIBILITY 3.2.2 .OPERATIONAL FEASIBILITY 3.2.3. ECONOMIC FEASIBILITY 3.3.SYSTEM SECURITY 3.3.1.SECURITY IN SOFTWARE	 10 11 12 12 13 13 14

	3.4. SYSTEM TESTING AND IMPLEMENTATION	15
	3.4.1. STRATEGIC APPROACH OF SOFTWARE TESTING	16
	3.4.2. UNIT TESTING	
4.	CHAPTER-4	
	4.1.SYSTEM DEVELOPEMENT ENVIRONMENT	20
	4.1.1.MODULE DESCRIPTION	21
	4.2.INTRODUCTION TO JAVA	24
	4.3. SERVLETS,JSP	34
	4.4. JDBC	37
	4.5.HTML,JAVASCRIPT	
5.	CHAPTER-5	40
	5.1.SYSTEM ARCHITECTURE	41
	5.2.E-R DIAGRAM	42
	5.3.FLOW DIAGRAMS	45
	5.4.DIAGRAMS	51
	5.5. OUTPUT&SCREENSHOTS	56
	5.6. CODINGS	
6.	CHAPTER-6	
	6.1. CONCLUSION	81
	6.2. FUTURE ENHANCEMENT	82
	6.3. REFERENCES	

CHAPTER-1

1.INTRODUCTION

1.1 INTRODUCTION TO PROJECT

A block chain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that any involved record cannot be altered retroactively, without the alteration of all subsequent blocks. This allows the participants to verify and audit transactions independently and relatively inexpensively. A block chain database is managed autonomously using a peer-to-peer network and a distributed time stamping server. They are authenticated by mass collaboration powered by collective self-interests. Such a design facilitates robust workflow where participants' uncertainty regarding data security is marginal. The use of a block chain removes the characteristic of infinite reproducibility from a digital asset. It confirms that each unit of value was transferred only once, solving the long-standing problem of double spending. Big data analytics is the often-complex process of examining large and varied data sets, or big data, to uncover information such as hidden patterns, unknown correlations, market trends and customer preferences that can help organizations make informed business decisions. Big data analytics is a form of advanced analytics, which involves complex applications with elements such as predictive models, statistical algorithms and what-if analysis powered by high-performance analytics systems. Big data analytics applications enable big data analysts, data scientists, predictive modelers, statisticians and other analytics professionals to analyze growing volumes of structured transaction data, plus other forms of data that are often left untapped by conventional BI and analytics programs. Encryption and access control are similar in the sense that they are both synonymous with privacy and prevention.

1.2 PURPOSE OF THE SYSTEM

To manage combination all the Banking Information.

To Manage the Information Transactions between the User and the client People.

To Manage the Transferring time and Security provided by the each user People.

To Maintain the Document register by the admin.

To Maintain the Accessibility of the System between the Customers and the Peoples.

CHAPTER-2

2.1. HARDWARE AND SOFTWARE REQUIREMENTS

Developing Kit			
	Processor	RAM	Disk Space
Eclipse	Computer with a 2.6GHz processor or higher	2GB	Minimum 20 GB
Database			
MySQL 5.0	Intel Pentium processor at 2.6GHz or faster	Minimum 512 MB Physical Memory; 1 GB Recommended	Minimum 20 GB
HeidiSQL 8.3	Intel Pentium processor at 2.6GHz or faster	Minimum 512 MB Physical Memory; 1 GB Recommended	Minimum 20 GB

Software Requirements:

- **Front end** : Core Java, CSS, JS, Servlet
- **Web application** : J2ee Frameworks, Hibernate
- **Back end** : MySQL 5.1

2.2. INPUT AND OUTPUT

The major inputs and outputs and major functions of the system are follows:

Input:

- The user must create the account for login. All the user details have been stored the data in our database for transferring money purpose.
- The user can send the data from user side to admin side and upload proper current update in our database and getting request and acknowledgement. After getting the file user can get valid information of money transfer design based on near.
- The user can view the need data through our application but they are show the sensitive data's in the database.
- They can show some available data from database server.

Output:

- User can enter the data and by using id they can transfer money from one end to another end from the client and they can secure internal details they can show only nearest branch of banking agency.
- The Application owner can show the user details and it can validate some sensitive information details. Finally it can send the acting admin and client.

2.3. INPUT DESIGN

- Input design is a part of overall system design. The main objective during the input design as given below.
- Input States: User can maintain a database in MySQL server or sql server for his/her business requirement.
- 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:

- In this section user can give the input for storage location and get the output from admin side.

LIMITATIONS

- Data integrity is difficult to manage and maintain in the existing system.
- Class imbalance should be more affected.
- Representativeness of the training set should be affected.
- The existing schemes cannot resist an entity that performs malicious activities for the process.

2.4. PROBLEMS IN EXISTING SYSTEM:

In existing system, when attempting to implement continuous learning ML use cases, there may be an extreme processing bottleneck because the data is typically stored in a standard disk-based transactional database (OLTP). Performing analysis on this data using traditional ML methods typically requires an extract, transform, and load (ETL) process to move the data into an analytical database (OLAP), where the data can be analysed should be an unpredictable, they cause some error due to overlapping data.

2.5. PROPOSED SYSTEM

In proposed system, a fraudulent banking transaction causes a loss of large sums of money and therefore has a negative impact on the economy as a whole. To prevent this, we must employ the latest technologies that we have at our disposal. As we have seen in the above experiment, by applying a simple Machine Learning Algorithm in the Banking sector and combining it with block chain technology, we could easily detect anomalies in banking transactions and prevent potential fraudulent activity at the root

Advantage of Proposed System:

- Wrapper should be get more accuracy and selection process should be efficient.
- Training sets should be easily representative.

- Class should be equally balanced.
- Management of the data parameters will be optimized.
- Single level coding should be more improvised.

2.6. LITERATURE SURVEY:

1. Title: Detecting Fraudulent Accounts on Blockchain: A Supervised Approach
Published Year: 29 October 2019
Author: Michał Ostapowicz, Kamil Żbikowski
2. Title: A Blockchain Framework for Insurance Processes
Published Year: 02 April 2018
Author: Mayank Raikwar; Subhra Mazumdar;
3. Title: Rethinking permissioned blockchains
Published Year: 17, Dec 2017.
Author: Vukolić and Marko
4. Title: Smart contract templates: essential requirements and design options
Published Year: 12 Aug 2016.
Author: C. D. Clack, V. A. Bakshi and L. Braine.
5. Title: Blockchains and smart contracts for the internet of things
Published Year: 20 May 2016.
Author: K. Christidis and M. Devetsikiotis
6. Title: Blockchain contract: Securing a blockchain applied to smart contracts.
Published Year: Watanabe, S. Fujimura,
Author: April 2016.
7. Title: Blockchain or not blockchain that is the question of the insurance and other sectors.
Published Year: F. Lamberti, V. Gatteschi
Author: Dec 2017.

8. Title: Blockchain cryptography and consensus,
Published Year: C. Christian,
Author: Aug2017
9. Title: Cryptography and protocols in hyperledger fabric
Published Year: 6 January, 2017.
Author: E. Androulaki, C. Cachin.
10. Title: Survey of consensus protocols on blockchain applications
Published Year: L. S. Sankar, M. Sindhu
Author: Jan 2017
11. Title: Credit Card Fraud Prevention Using Blockchain.
Published Year: May 2021.
Author: E.M.S.W Balagolla; W.P.C Fernando; R.M.N.S Rathnayake; M.J.M.R.P Wijesekera.
12. Title: *An Overview of Blockchain Technology: Architecture Consensus and Future Trends..*
Published Year: October 2018.
Author: Zheng and H. Dal.
13. Title: The Evolution of Blockchain: A Bibliometric Study.
Published Year: December 2019.
Author: Qi M. Dabbagh, M. Sookhar and N. Safaulin Wang.
14. Title: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts
Published Year: November 2020.
Author: A. Kosba, A. Miller and Z. Wen, Hawk.
15. Title: *Credit Card Fraud Prevention.*
Published Year: August 2020.
Author: W. Yaokumah and G. Ntow-Danso.

CHAPTER-3

3.1. SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

The purpose of this document is to present a detailed description of the Web application system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

PURPOSE

The purpose of this Software Requirement Specification (SRS) is to help the project. It is provided with some requirements which are used in the Transaction Mercator System. All parts; design, coding and testing will be prepared with helping of SRS. The purpose of this document is to detail the requirements placed on the Transaction Mercator System and serves as a contract between the customer and the developers as to what is to be expected of the stock exchange, and how the components of the system are working with each other with external systems.

This document will be checked by the group member's supervisor and it will corrected by members if supervisor orders.

DEVELOPERS RESPONSIBILITIES OVERVIEW:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

3.1.1 FUNCTIONAL REQUIREMENTS:

- Following is a list of functionalities of the browsing enabled system.
- An Activity with a UI that allows you to browser settings. Provide a second Activity that allows users to access the share with permission from the administrator. Handle activity lifecycle appropriately. A precondition for any points in this part of the grade is code that compiles and runs.
- Your application should allow a user to browse the shares, buy and sell the shares with specific metadata. The assignment requires you to create a UI for browsing and a UI for integrating the two.
- The Net beans provide a number of useful layout components, views, and tools that you may want to use to create your location browser. As with the final project, you should design your application to only use the buttons on the Key board and mouse as input. Your application should use the Key board, Mouse and keywords.

3.1.2 NON-FUNCTIONAL REQUIREMENTS:

- The system should be supported Net beans. The member should use the System browser. Each member should have a separate system.
- The system should ask the username and password to open the application. It doesn't permit to unregistered user to access the System.
- The system should have Role based System functions access. Approval Process has to be defined.
- The system should have Modular customization components so that they can be reused across the implementation.
- These are the mainly following:
 - Secure access of confidential data. 24 X 7 availability
 - Better component design to get better performance at peak time
 - Flexible service based architecture will be highly desirable for future extension

3.1.3 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 in the part of the users of the existing system to give the required 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.

3.2 FEASIBILITY REPORT

3.2.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Database Security System'. The current system developed is technically feasible. It is a browser based user interface for construction workflow. Thus, it provides an easy access to the users. The database's purpose is to create, establish and maintain a

workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the rules specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.2.2 Operational Feasibility

The analyst considers the extent that the proposed system will fulfill his departments. That is, whether the proposed system covers all aspects of the working system and whether it has considerable improvements. We have found that the proposed “Big crypt encryption algorithm” will certainly have considerable improvements over the existing system.

3.2.3 Economic Feasibility

The proposed system is economically feasible because the cost involved in purchasing the hardware and the software is within approachable. Working with this system need not require a highly qualified professional. The operating-environment costs are marginal. The less time involved also helps in its economic feasibility.

3.3 SYSTEM SECURITY

INTRODUCTION

Security system can be divided into four related issues: The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the proper functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

3.3.1. SECURITY IN SOFTWARE

System security refers to various validations on data in the form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks and controls:

CLIENT SIDE VALIDATION

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks are imposed:

- JavaScript is used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.

- Tab-indexes are set according to the need and taking into account the ease of use while working with the system.

SERVER SIDE VALIDATION

Some checks cannot be applied on the client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- A server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results in a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- The user is intimated through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled the server side.
- Using server side validation, constraints on several restricted operations are imposed.

3.4. SYSTEM TESTING AND IMPLEMENTATION

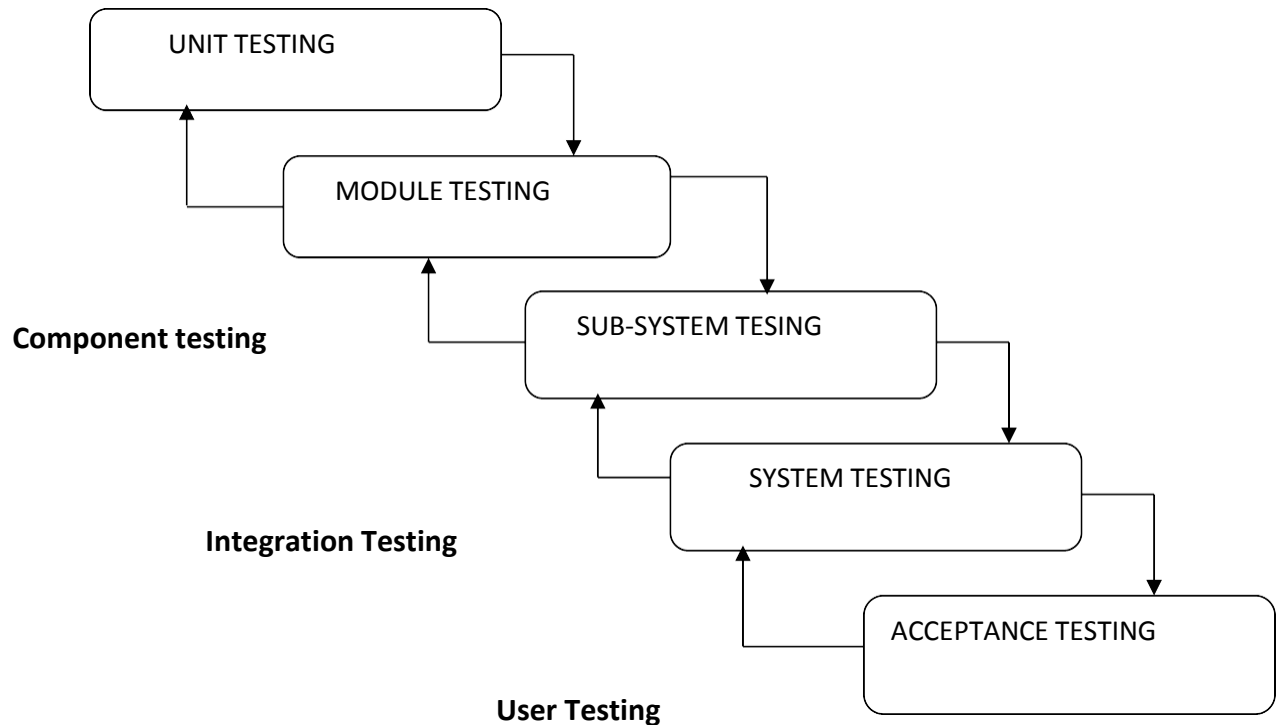
INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

3.4.1. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress is done by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



3.4.2. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .We have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

2. BASIC PATH TESTING

The established technique of flow graph with Cyclamate complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graphs.

Determine the Cyclamate complexity of the resultant flow graph, using formula:

$$V(G) = E - N + 2 \text{ or}$$

$$V(G) = P + 1 \text{ or}$$

$$V(G) = \text{Number of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program, according to the location of the definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loop test the innermost loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of a connected loop.

No	Test Scenario	Expected Result	Test Result
1	Username is correct. Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
2	Username is incorrect. Password is correct.	Username and Password is incorrect.	Username and Password is incorrect.
3	Username is empty. Password is correct.	Username is required.	Username is required.
4	Username is correct. Password is empty.	Password is required.	Password is required
5	Both Username and Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
6	Both Username and Password is empty.	Username and Password is required.	Username and Password is required.
7	Both Username and Password is correct.	Login Successful.	Login Successful.

CHAPTER-4

4.1. SYSTEM DEVELOPEMENT ENVIRONMENT

4.1.1. MODULE DESCRIPTON:

Account type:

In our application, account with personal and business type information should be getting by admin from the user by this we can login and feed our financial and regulatory information into the database for future purpose like lending money. In this module, we have two type of accounts personal and business type from this we have get information through two different forms in this module.

Personal Account:

In this module, users have personal account then they can register in this module those information should be view by admin. Sub modules like account enquiry, account request, account acknowledgement. Account enquiry is used to view the mini statement and transaction details of that account in protected way. Account request and acknowledgement is used to send the request to asking funds through account id and acknowledgment should be view by the related account.

Business Account:

In case of some users have business account then they can register in this module those information should be view by admin. This registration form has some different attributes compared to personal account because they has current account option so this has sensitive data highly for this stored information should be secured.

Account Information:

Personal information and business information should be secured by our concept called big crypt to protect data by selecting attributes based on high priority to select important attribute value. Instead of protect all the data we can select important attributes of big data to protect that main parts so the whole big data information should be protected. In this application many users and clients should be evoked so large number of information that should be protected by admin using big crypt method. . Our application secure the data entirely this makes a time delay for

secure the data base to that end, they conducted an analysis on big data, analyzed the current big data situation. Local account transaction should be done by this page by getting account information and transfer by using domestic account and international money transfer should be done by sweep account information in a secured way.

Money Transaction:

National and international account transaction should be done from various places should be accurately done and by using crypto currency method money should be transferred from one place to another place so accounts has different place should be securely kept and money transaction should be done in secured manner. They do not implement the data security for high priority attributes of the large data. Complexity in occurs on enormous number of data, so Implementation of cyber security is difficult, but pointing of attributes like sensitive transaction details should be secured in this module.

4.2. INTRODUCTION TO JAVA

About Java:

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e. Architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent
- Except for those constraints imposed by the Internet environment. Java gives the programmer, full control

Finally Java is for Internet Programming where c was to System Programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and

Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, have opened the door to an exciting new form of program called the Applet.

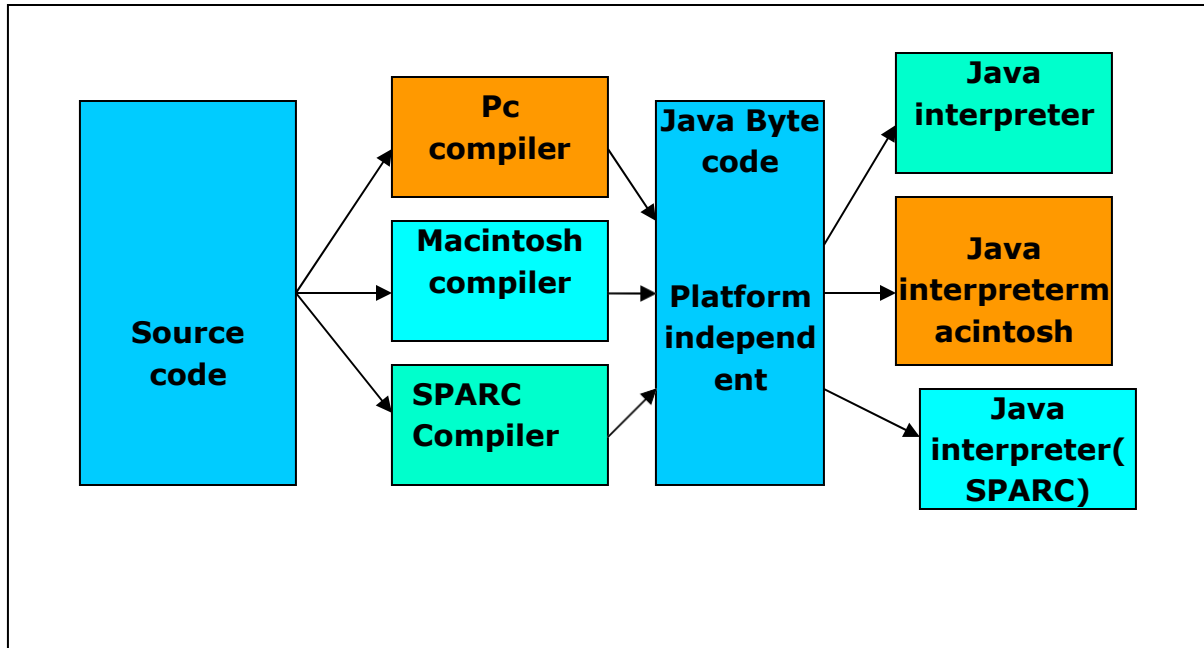
Applications and applets. An application is a program that runs on our Computer under the operating system on that computer. It is more or less like one creating, using C or C++.Java's ability to create Applets makes it important. An Applet is an application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be reacted to the user input and dynamically change.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed for a machine in the same room or across the planet.

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called a Java Virtual Machine (JVM). The JVM is supposed to be executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called a Java Virtual Machine.

Compiling and interpreting Java source code.



During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through the internet and run the Applets.

Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean, usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

4.3 SERVLETS/JSP

A Servlet is a generic server extension. Java classes that can be loaded dynamically to expand the functionality of a server. Servlets are commonly used with web servers. Where they can take the place CGI scripts.

A Servlet is similar to proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server, so it is safe and portable Servlets operate solely within the domain of the server.

Unlike CGI and Fast CGI, which use multiple processes to handle separate program or separate requests, separate threads within the web server process handle all servlets. This means that servlets are all efficient and scalable.

Servlets are portable; both across operating systems and also across web servers. Java Servlets offer the best possible platform for web application development.

Servlets are used as a replacement for CGI scripts on a web server; they can extend any sort of server, such as a mail server that allows servlets extend its functionality, perhaps by performing a virus scan on all attached documents or handling mail filtering tasks.

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming including inextensible scripting solutions platform-specific API's and incomplete interface.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side. They are byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example an HTTP servlet can be used to generate dynamic HTML content when you use servlets to do dynamic content you get the following advantages:

They're faster and cleaner than CGI scripts

They use a standard API (the servlet API)

They provide all the advantages of Java (run on a variety of servers without needing to be rewritten)

Attractiveness of servlets:

There are many features of servlets that make them easy and attractive to use these include:

- Easily configure using the GUI-based Admin tool]
- Can Be Loaded and Invoked from a local disk or remotely across the network.
- Can be linked together or chained, so that one servlet can call another servlet or several servlets in sequence.
- Can be called dynamically from within HTML, pages using server-side include-tags.
- Are secure-even when downloading across the network, the servlet security model and servlet and box protect your system from unfriendly behavior.

Advantages of the Servlet API

One of the great advantages of the servlet API is protocol independent. It assumes nothing about:

- The protocol being used to transmit on the net
- How it is loaded
- The server environment it will be running in

These quantities are important, because it allows the Servlet API to be embedded in many different kinds of servers.

There are other advantages to the servlet API as well these include:

- It's extensible-you can inherit all your functionality from the base classes made available to you
- It's small, simple, and easy to use.

Features of Servlets:

- Servlets are persistent. Servlet is loaded only by the web server and can maintain services between requests.
- Servlets are fast. Since servlets only need to be loaded once, they offer much better performance over their CGI counterparts.
- Servlets are platform independent.
- Servlets are extensible Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.
- Servlets are secure
- Servlets are used with a variety of clients.

Every servlet must implement the `javax.servlet` interface. Most servlets implement it by extending one of two classes, `javax.servlet.GenericServlet` or `javax.servlet.http.HttpServlet`. A protocol-independent servlet should subclass `GenericServlet`. While an `HttpServlet` should subclass `HttpServlet`, which is itself a subclass of `GenericServlet` with added HTTP-specific functionality.

Unlike a java program, a servlet does not have a `main()` method, Instead the server in the process of handling requests invoke certain methods of a servlet. Each time the server dispatches a request to a servlet, it invokes the servlet's `Service()` method.

A generic servlet should override its `service()` method to handle requests as appropriate for the servlet. The `service()` accepts two parameters a request object and a response object. The request object tells the servlet about the request, while the response object is used to return a response

In Contrast,an Http servlet usually does not override the service () method. Instead, it overrides doGet () to handle GET requests and doPost () to handle Post requests. An Http servlet can override either or both of these modules the service() method of HttpServlet handles the setup and dispatching to all the doXXX() methods. Whichiswhy it usually should not be overridden

The remainders in the javax.servlet and Java. servlet. http. The package is largely supported class's.The ServletRequest and ServletResponse classes in Java. servlet is provided access to generic server requests and responses while HttpServletRequest and HttpServletResponse classes in javax.servlet provide access to generic server requests and responses while HttpServletRequest and HttpServletResponse in javax.servlet.http provide access a HTTP requests and responses. The javax.servlet.http provide contains an HttpSession class that provides built-in session tracking functionality and Cookie class that allows quick setup and processing Http Cookies.

Loading Servlets:

Servlets can be loaded from their places. From a directory that is in the CLASSPATH. The CLASSPATH of the JavaWebServer includes service root/classes/, which is where the system classes reside

From the <SERVICE_ROOT/servlets/directory. This is not in the server's classpath. A class loader is used to create servlet form this directory. New servlets can be added-existing servlets can be recompiled and the server will notice these changes. From a remote location.For this a code base like http://nine.eng/classes/foo/ is required in addition to the servlet's class name.Refer to the admin GUI docs on servlet section to see how to set this up.

Loading Remote Servlets

Remote servlets can be loaded from:

- Configuring the admin Tool to setup automatic loading of remote servlets.
- Selection up server side include tags in .html files

Invoking Servlets

A servlet invoker is a servlet that invokes the “server” method of a named servlet. If the servlet is not loaded into the server, then the invoker first loads the servlet (either from local disk or from the network) and then invokes the “service” method. Also like applets, local servlets in the server can be identified by just the class name. In other words, if a servlet name is not absolute, it is treated as local.

A Client can Invoke Servlets in the Following Ways:

- The client can ask for a document that is served by the servlet.
- The client(browser) can invoke the servlet directly using a URL, once it has been mapped using the SERVLET ALIASES Section of the admin GUI
- The servlet can be invoked through server side include tags.
- The servlet can be invoked by placing it in the servlet/directory
- The servlet can be invoked by using it in a filter chain

The Servlet Life Cycle:-

The Servlet life cycle is one of the most exciting features of Servlets. This life cycle is a powerful hybrid of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The Servlet life cycle allows servlet engines to address both the performance and resource problems of CGI and the security concerns of low level server API programming.

The servlet life cycle is highly flexible. Servers have significant leeway in how they choose to support servlets. The only hard and fast rule is that a servlet engine must conform to the following life cycle contract:

- Create and initialize the servlets
- Handle zero or more service for clients
- Destroy the servlet and then garbage collects it.

It's perfectly legal for a servlet to be loaded, created and initialized in its own JVM, only to be destroyed and garbage collected without handling any client request or after handling just one request

The most common and most sensible life cycle implementations for HTTP servlets are: Single Java virtual machine and stateless persistence.

Init and Destroy:-

Just like Applets servlets can define `init ()` and `destroy ()` methods, A servlets `init (ServletConfig)` method is called by the server immediately after the server constructs the servlet instance. Depending on the server and its configuration, this can be at any of these times

- When the server starts
- When the servlet is first requested, just before the `service()` method is invoked
- At the request of the server administrator

The `init ()` method is typically used to perform servlet initialization creates or loading objects that are used by the servlet in the handling of its request. In order to provide a new Servlet any information about itself and its environment, a server has to call a `servletsinit ()` method and pass an object that implements the `ServletConfig` interface.

This `ServletConfig` object supplies a servlet with information about its initialization parameters. These parameters are given to the servlets and are not associated with any single request. They can specify initial values, such as where a counter should begin counting, or default values, perhaps a template to use when not specified by the request,

The server calls a servlet `destroy ()` method when the servlet is about to be unloaded. In the `destroy ()` method, a Servlet should free any resources it has acquired that will not be garbage collected. The `destroy ()` method also gives a servlet a chance to write out its unsaved. Cached information or any persistent information that should be read during the next call to `init ()`.

USER AUTHORIZATION:

One way to perform session tracking is to leverage the information that comes with User authorization. When a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. After the client logs in, the username is available to a servlet through `getRemoteUser ()`.

When use the username to track the session. Once a user has logged in, the browser remembers her user name and resends the name and password as the user views new pages on the site. A servlet can identify the user through her username and they're by Track her session.

The biggest advantage of using user authorization to perform session tracking is that it's easy to implement. Simply tell the protect a set of pages, and use `getRemoteUser()` to identify each client. Another advantage is that the technique works even when the user accesses your site form or exits her browser before coming back.

The biggest disadvantage of user authorization is that it requires each user to register for an account and then log in each time the starts visiting your site. Most users will tolerate registering and logging in as a necessary evil when they are accessing sensitive information, but it's all overkill for simple session tracking. Another problem with user authorization is that a user cannot simultaneously maintain more than one session at the same site.

Hidden Form Fields:

One way to support anonymous session tracking is to use hidden from the fields. As the name implies, these are fields added to an HTML, from that are not displayed in the client's browser, they are sent back to the server when the form that contains them is submitted.

In a sense, hidden form fields define constant variables for a form. For a servlet receiving a submitted form, there is no difference between a hidden field and a visible filed.

As more information is associated with a client's session. It can become burdensome to pass it all using hidden form fields. In these situations, it's possible to pass on just a unique session ID that identifies as a particular client session.

That session ID can be associated with complete information about its session that is stored on the server.

The advantage of hidden form fields is their ubiquity and support for anonymity. Hidden fields are supported in all the popular browsers, they demand no special server requirements, and they can be used with clients that haven't registered or logged in.

The major disadvantage with this technique, however, is that it works only for a sequence of dynamically generated forms, the technique breaks down immediately with static documents, emailed documents bookmarked documents and browser shutdowns.

URL Rewriting:

URL rewriting is another way to support anonymous session tracking. With URL rewriting every local URL the user might click on is dynamically modified. Or rewritten, to include extra information. The extra information can be in the form of extra path information, added parameters, or some custom, server-specific URL change. Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session.

Each rewriting technique has its own advantage and disadvantage

Using extra path information works on all servers, and it works as a target for forms that use both the Get and Post methods. It does not work well if the servlet has to use the extra path information as true path information

The advantages and disadvantages of URL Rewriting closely match those of hidden form fields. The major difference is that URL rewriting works for all dynamically created documents, such as the Help servlet, not just forms. With the right server support, custom URL rewriting can even work for static documents.

Persistent Cookies:

A fourth technique to perform session tracking involves persistent cookies. A cookie is a bit of information. Sent by a web server to a browser that can later be read back from that browser. When a browser receives a cookie, it saves the cookie and thereafter sending the cookie back to

the server each time it accesses a page on that server, subject to certain rules. Because a cookie's value can uniquely identify a client, cookies are often used for session tracking.

Persistent cookies offer an elegant, efficient, easy way to implement session tracking. Cookies provide as automatic an introduction for each request, as we could hope for. For each request, a cookie can automatically provide a client's session ID or perhaps a list of clients' performance. The ability to customize cookies gives them extra power and versatility.

The biggest problem with cookies is that browsers don't always accept cookies sometimes this is because the browser doesn't support cookies. More often it's because the browser doesn't support cookies. More often it's because the user has specifically configured the browser to refuse cookies.

The power of servlets:

The power of servlets is nothing but the advantages of servlets over other approaches, which include portability, power, efficiency, endurance, safety, elegance, integration, extensibility and flexibility.

Portability:

As servlets are written in Java and conform to a well defined and widely accepted API. They are highly portable across operating systems and across server implementation. We can develop a servlet on a Windows NT machine running the Java web server and later deploy it effortlessly on a high-end UNIX server running Apache. With servlets we can really "write once, serve everywhere"

Servlet portability is not the stumbling block it so often is with applets, for two reasons

First, Servlet portability is not mandatory, i.e. servlets have to work only on server machines that we are using for development and deployment

Second, servlets avoid the most error-prone and inconstancy implemented portions of the Java languages.

Power:

Servlets can harness the full power of the core Java. API's: such as Networking and Url access, multithreading, image manipulation, data compression, database connectivity, internationalization, remote method invocation (RMI) CORBA connectivity, and object serialization, among others.

Efficiency and Endurance:

Servlet invocation is highly efficient, Once a servlet is loaded it generally remains in the server's memory as a single object instance, Thereafter the server invokes the servlet to handle a request using a simple, light weighted method invocation. Unlike the CGI, there's no process to spawn or interpreter to invoke, so the servlet can begin handling the request almost immediately, Multiple, concurrent requests is handled the request almost immediately. Multiple, concurrent requests are handled by separate threads, so servlets are highly scalable.

Servlets in general are enduring objects. Because a servlet stays in the server's memory as a single object instance. It automatically maintains its state and can hold onto external resources, such as database connections.

Safety:

Servlets support safe programming practices on a number of levels. As they are written in Java, servlets inherit stronger type safety of the Java language. In addition the servlet API is implemented to be type safe. Java's automatic garbage collection and lack of pointers mean that servlets are generally safe from memory management problems like dangling pointers invalid pointer references and memory leaks.

Servlets can handle errors safely, due to Java exception – handling mechanism. If a servlet divides by zero or performs some illegal operations, it throws an exception that can be safely caught and handled by the server.

A server can further protect itself from servlets through the use of Java security manager. A server can execute its servlets under the watch of a strict security manager.

Elegance:

The elegance of the servlet code is striking. Servlet code is clean, object oriented modular and amazingly simple one reason for this simplicity is the served API itself. This includes methods and classes to handle many of the routine chores of servlet development. Even advanced to operations like cookie handling and session tracking are abstracted into convenient classes.

Integration:

Servlets are tightly integrated with the server. This integration allows a servlet to cooperate with the server in two ways. For e.g.: a servlet can use the server to translate file paths, perform logging, check authorization, perform MIME type mapping and in some cases even add users to the server's user database.

Extensibility and Flexibility:

The servlet API is designed to be easily extensible. As it stands today the API includes classes that are optimized for HTTP servlets. But later it can be extended and optimized for another type of servlets. It is also possible that its support for HTTP servlets could be further enhanced.

Servlets are also quite flexible; Sun also introduced Java server pages. Which offer a way to write snippets of servlet code directly within a static HTML page using syntax similar to Microsoft's Active server pages (ASP)

4.4. JDBC**What is JDBC?**

Any relational database. One can write a single program using the JDBC API, and the JDBC is a Java Api for executing SQL, Statements (As a point of interest JDBC are trademarks names and is not an acronym; nevertheless, Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API

Using JDBC, it is easy to send SQL statements to virtually program will be able to send SQL Statements to the appropriate database. The Combination of Java and JDBC lets a programmer writes it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things

- Establish a connection with a database
- Send SQL statements
- Process the results
- JDBC Driver Types
- The JDBC drivers that we are aware of this time fit into one of four categories
- JDBC-ODBC Bridge plus ODBC driver
- Native-API party-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the `java.sql.Driver` interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC, data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than development and very limited development.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavors. There are four driver categories

Type 01-JDBC-ODBC Bridge Driver

Type 01 drivers use a bridge technology to connect a Java client to an ODBC database service. Sun's JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

Type 02-Native-API party-Java Driver

Type 02 drivers wrap a thin layer of Java around a database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for c/c++ programmers, Because type-02 drivers are implemented using native code. In some cases, they have better performance than their all-Java counterparts. They add an element of risk, however, because a defect in a driver's native code section can crash the entire server

Type 03-Net-Protocol All-Java Driver

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all Java, which makes them useful for applet deployment and safe for servlet deployment

Type-04-native-protocol All-Java Driver

Type o4 drivers are the most direct of the lot. Written entirely in Java, Type 04 drivers understand database-specific networking. Protocols and can access the database directly without any additional software

JDBC-ODBC Bridge

If possible use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC.It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, library, the ODBC driver library, and the database client library).

WHAT IS The JDBC-ODBE Bridge?

The JDBC-ODBC Bridge is a Jdbc driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge

is implemented as the sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intersolv and Java Soft

4.5. HTML, JAVASCRIPT

Hypertext Markup Language(HTML), the languages of the World Wide Web (WWW), allows users to produce web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language, but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but Specialized to hypertext and adapted to the Web. The idea behind Hypertext one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop

HTML provides tags (special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic Html Tags:

<code><!-- --></code>	Specific Comments.
<code><A>.....</code>	Creates Hypertext links.
<code>.....</code>	Creates hypertext links.
<code><Big>.....</Big></code>	Formats text in large-font
<code><Body>..... </Body></code>	contains all tags and text in the Html-document
<code><Center>.....</Center></code>	Creates Text

<code><DD>.....</DD></code>	Definition of a term.
<code><TABLE>..... </TABLE></code>	creates a table
<code><Td>.....</Td></code>	indicates table data in a table.
<code><Tr>.....</Tr></code>	designates a table row
<code><Th>..... </Th></code>	creates a heading in a table.

ADVANTAGE

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent
- HTML tags are not case-sensitive.

JAVA SCRIPT

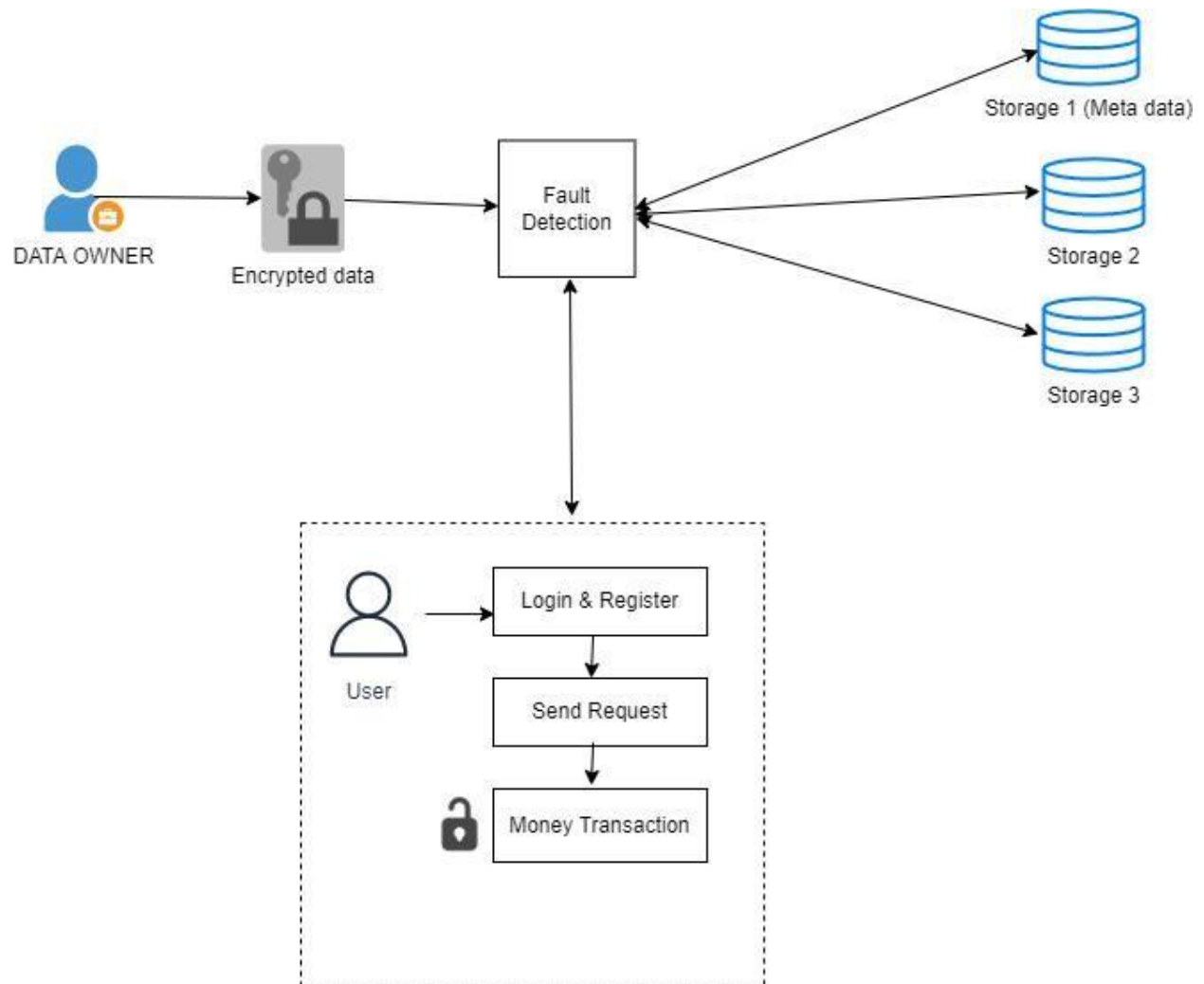
JavaScript is a compact, object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. And Livewire enables you to create server-based applications similar to common gateway interface (CGI) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form input, and page navigation.

For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

CHAPTER-5

5.1 .SYSTEM ARCHITECTURE :



5.2. E – R DIAGRAMS

- The relation upon the system is structured through a conceptual ER-Diagram, which not only specifies the existing entities, but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct, the data modeling activity the attributes of each data object noted, is the ERD can be described as a data object description.
- The set of primary components that are identified by the ERD are
 - Data object
 - Relationships
 - Attributes
 - Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.

5.3. FLOW DIAGRAMS

A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processing, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. The process is further identified with a number that will be used for identification purpose. The development of DFD'S is done on several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit,

which plays a vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analysts to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programmed in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD’S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw the long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SAILENT FEATURES OF DFD'S

1. The DFD shows the flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process, whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL

In Current Physical DFD process label includes the name of the people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly, data flows and data stores are often labelled with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects in the system are removed as much as possible so that the current system is reduced to its essence to the data and the processes that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system, but had problems with how it was implemented typically through the new logical model will differ from the current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

DATA STORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into the data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data

- 1) Data cannot move directly from a source to sink it must be moved from a process
- 2) A source and /or sink have a noun phrase label

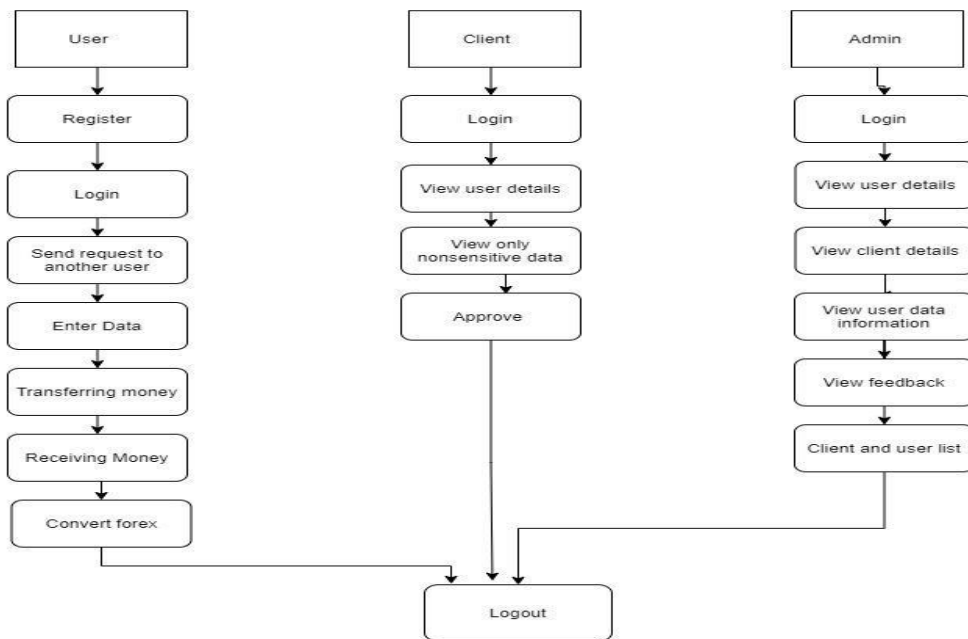
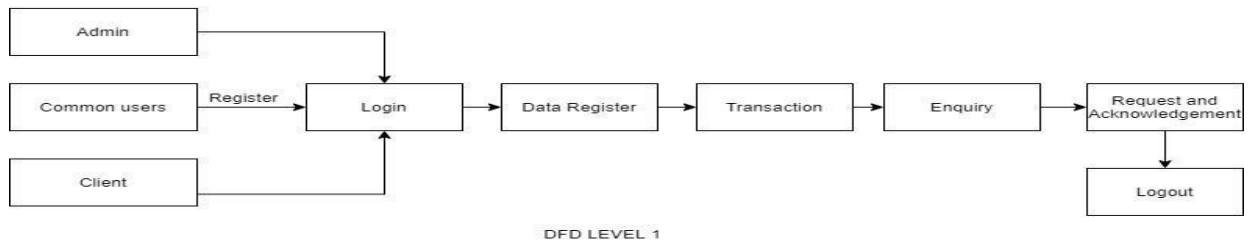
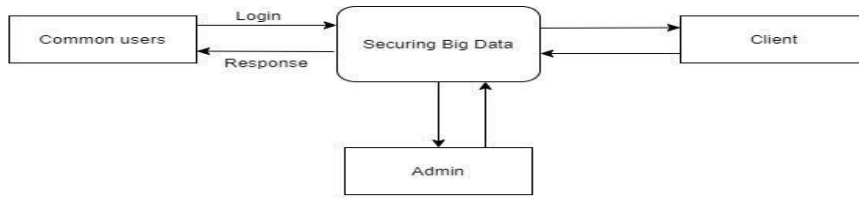
DATA FLOW

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however by two separate arrows since these happen at different type.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data in the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.

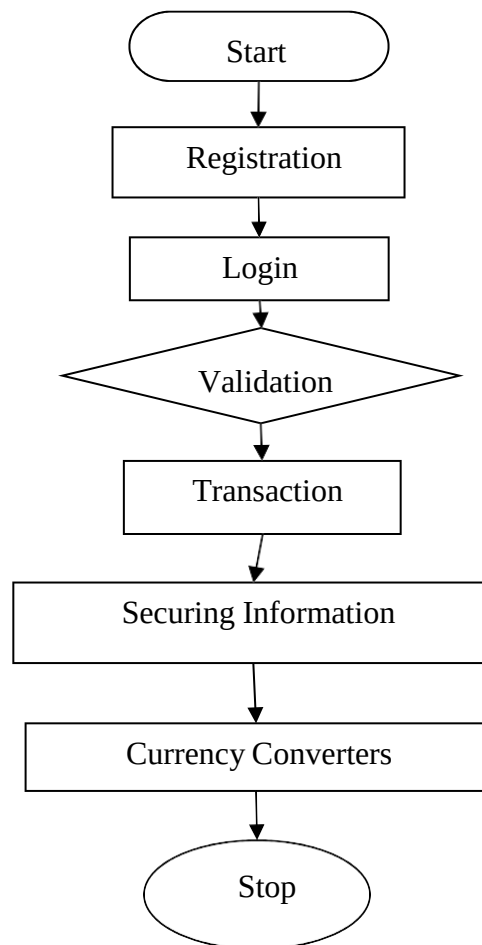
A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

5.4.DIAGRAMS :

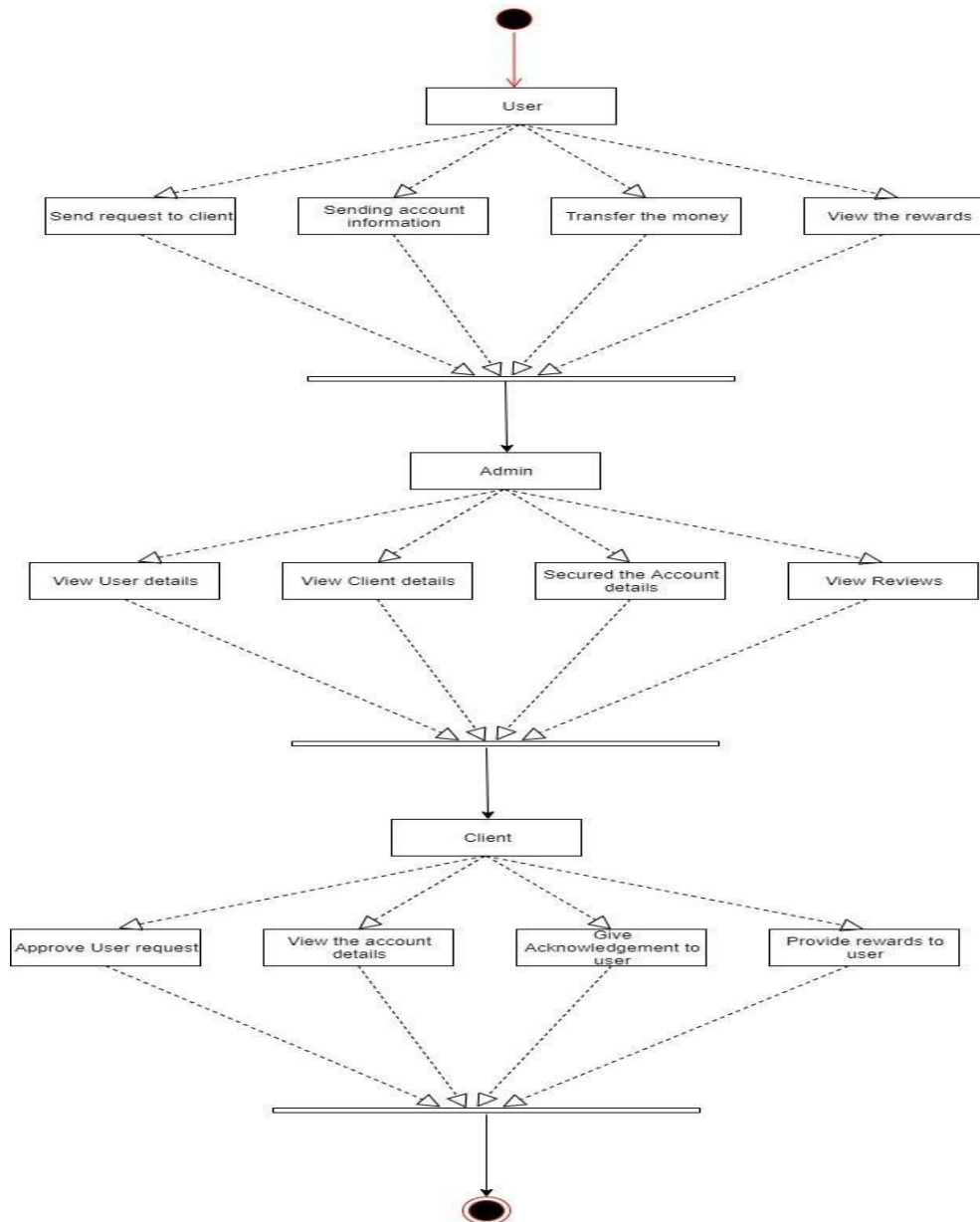
DATA FLOW DIAGRAMS:



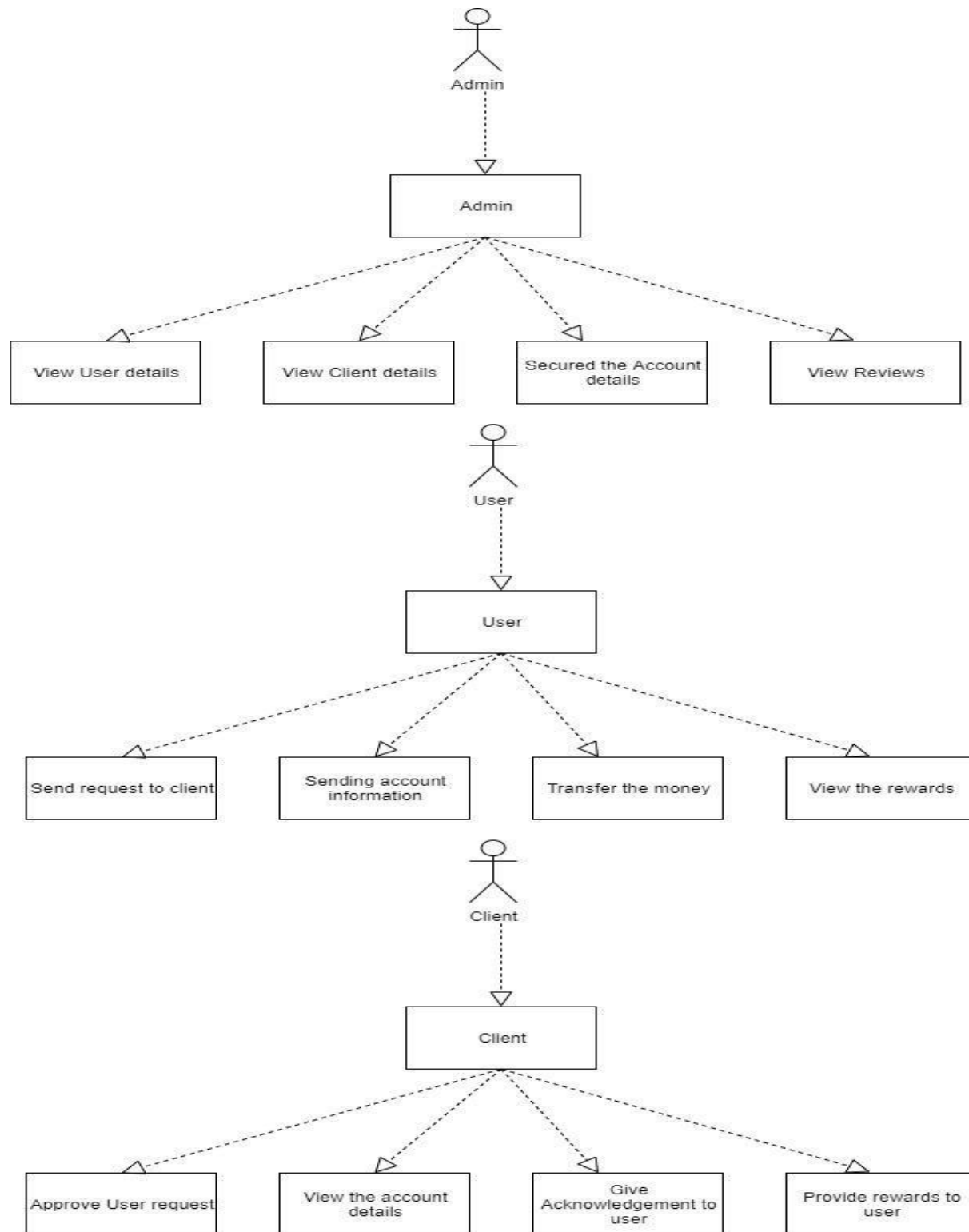
FLOW CHART:



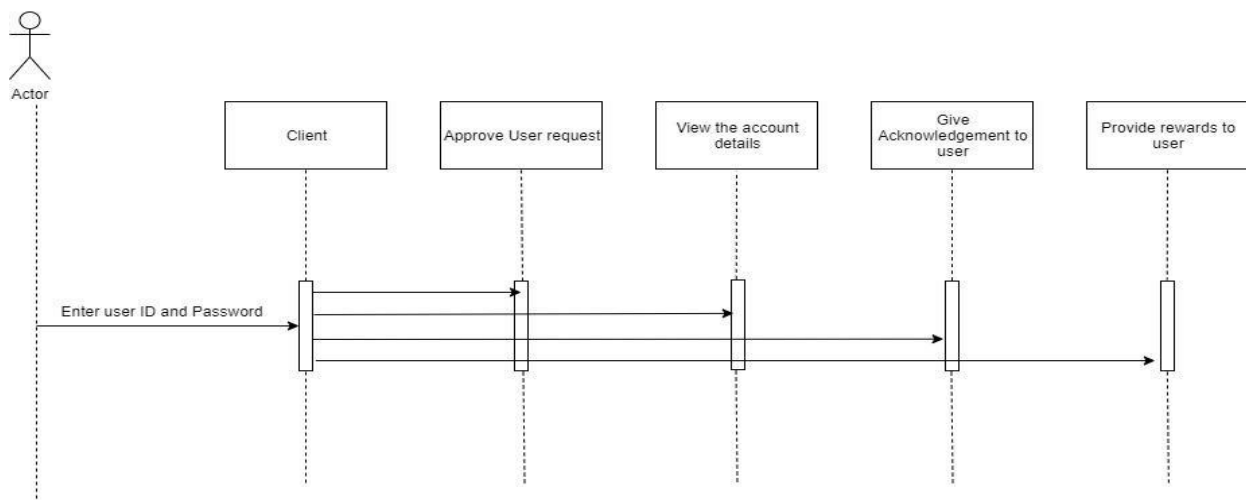
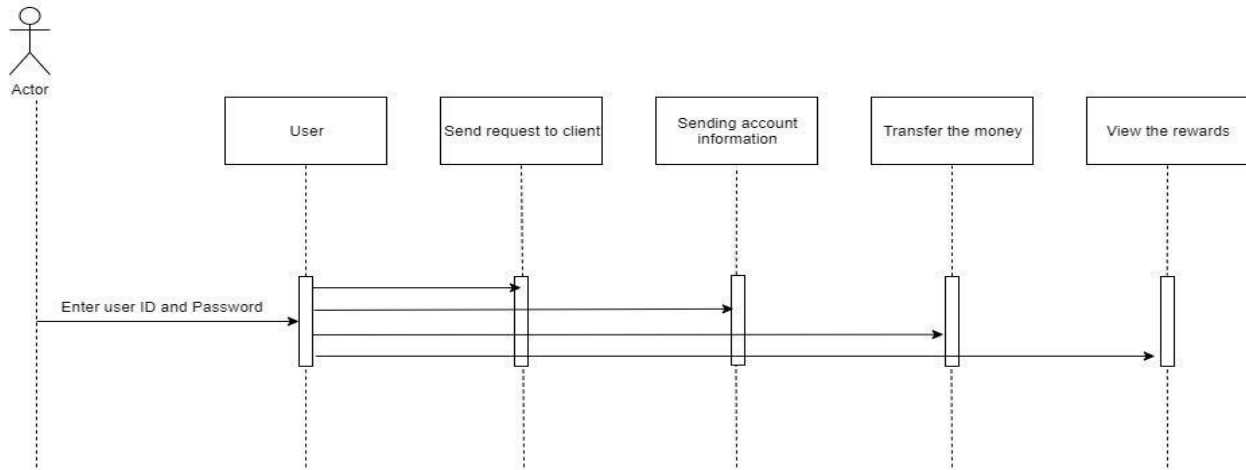
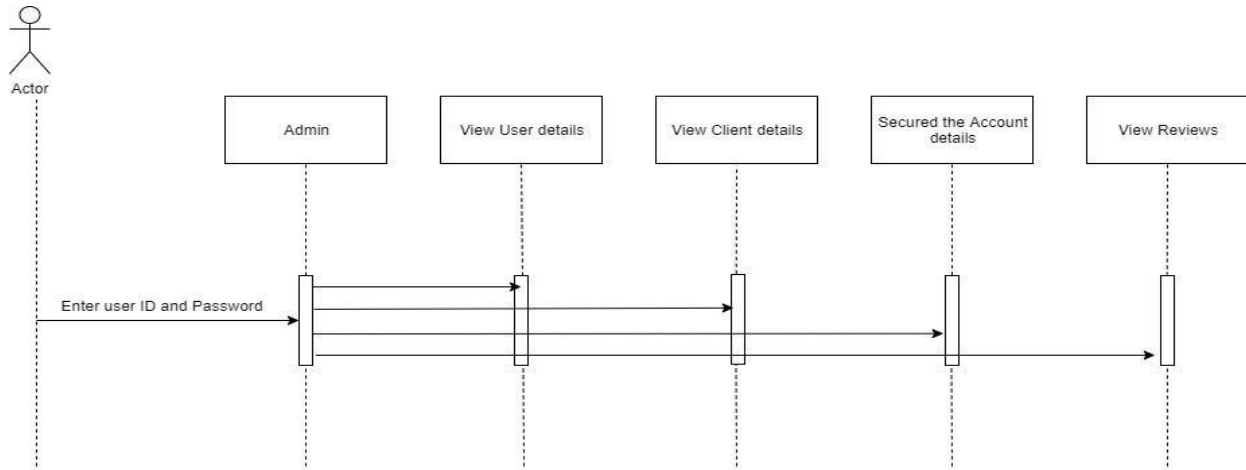
ACTIVITY DIAGRAM :



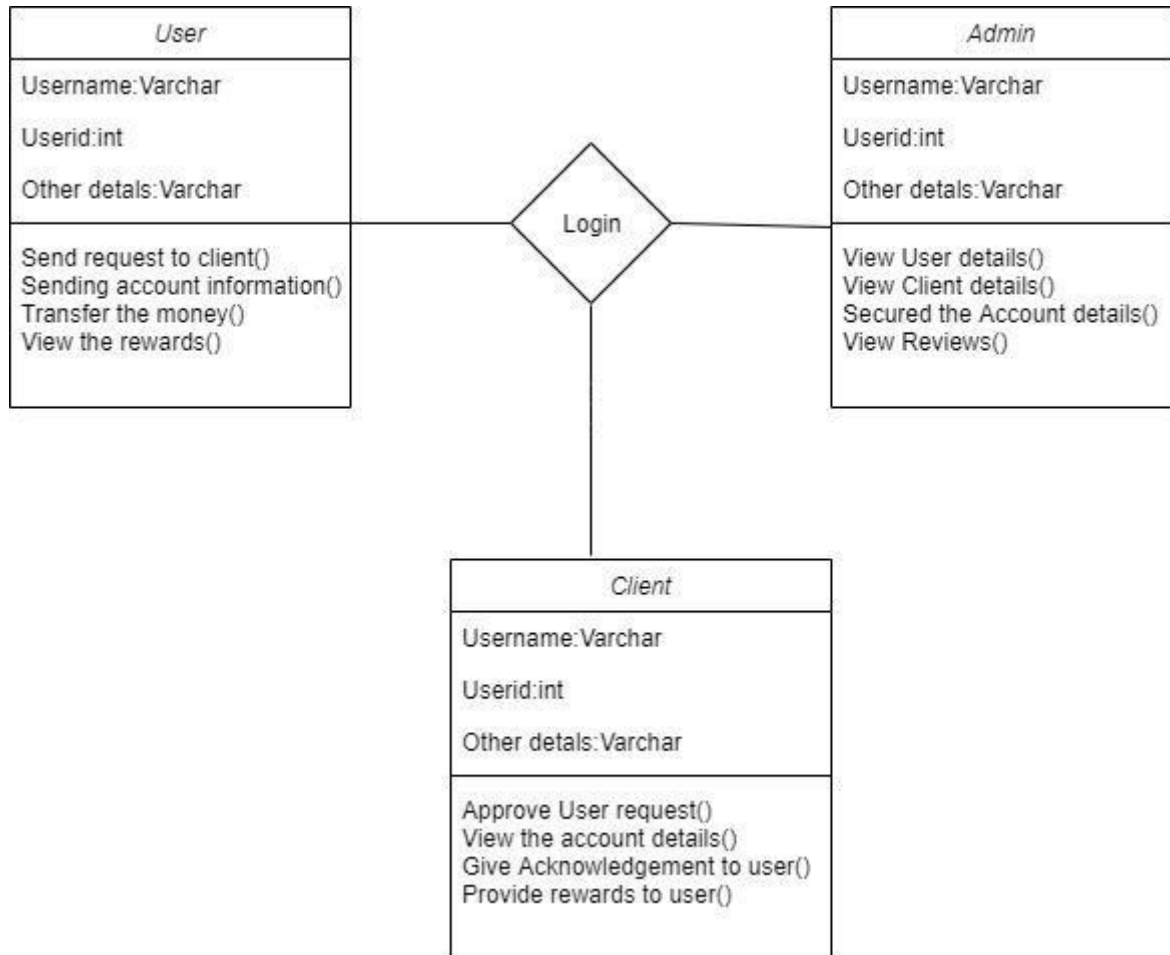
USE CASE DIAGRAM



SEQUENCE DIAGRAM

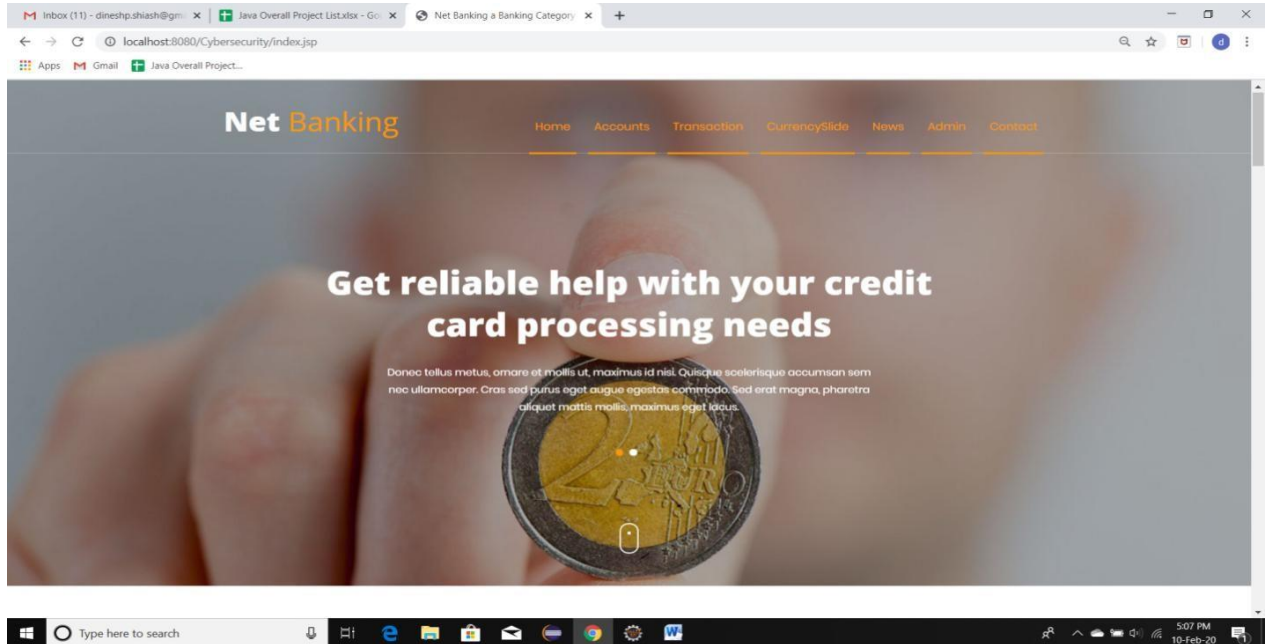


CLASS DIAGRAM

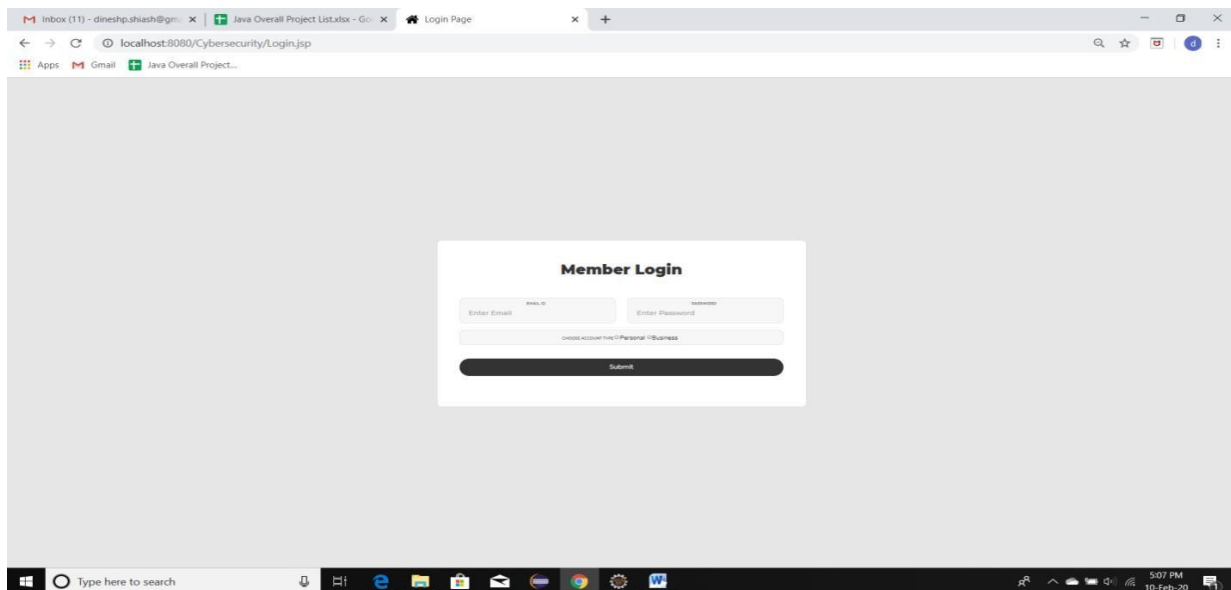


5.5.OUTPUT SCREENS

Homepage:



User Login:



Registration:

Personal Account

EMPLOYEE NAME Enter Your Name	EMPLOYEE ACCOUNT NO Enter Your Ac number
EMPLOYEE BUSINESS OWNER NAME Enter Your Ac number	PIU CODE Enter code
ACCOUNT OPENING DATE dd-mm-yyyy	EMAIL ID Enter your mail
PASSWORD Enter your password	
NICK NAME Enter your nick name	
EMPLOYEE ACCOUNT TYPE Current	
EMPLOYEE NAME Anna Nagar	
EMPLOYEE CODE AN000001	
ADDRESS Enter Your Address	
Submit	

Business Account

EMPLOYEE NAME Enter Your Business Name	EMPLOYEE NAME Enter your name
EMPLOYEE OWNER Enter Your Business Owner Name	EMPLOYEE ACCOUNT NO Enter your Account number
NICK NAME Enter your name	IN EMPLOYEE BUSINESS OWNER NAME Enter your Account number
EMPLOYEE EMAIL ID Enter Your Mail id	EMPLOYEE EMAIL ID Enter your mail
PIU CODE Enter code	ACCOUNT OPENING DATE dd-mm-yyyy
PASSWORD Enter your password	EMPLOYEE BUSINESS Payment
BUSINESS TYPE Type 1	
EMPLOYEE ACCOUNT TYPE Loan Amount	
EMPLOYEE NAME Anna Nagar	
EMPLOYEE CODE AN000001	
ADDRESS India	
CONTACT NUMBER Enter Your Contact No	

Transaction:

The screenshot displays a web browser window with the address bar showing 'localhost:8080/Cybersecurity/domestic_account.jsp'. The browser tabs include 'Inbox (11) - dineshp.shilash@gmail.com', 'Java Overall Project List.xlsx - Google', and 'Insert title here'. The main content area features a form titled 'Domestic Account' with the following fields:

- FROM ACCOUNT: Select An Account ▼
- BENEFICIARY: Select An Beneficiary ▼
- BENEFICIARY ACCOUNT NUMBER
- BENEFICIARY ACCOUNT TYPE
- BENEFICIARY IFIC CODE
- AMOUNT
- REMARKS
- ☐ I HAVE READ AND ACCEPTED THE TERMS AND CONDITIONS

A 'Continue' button with a right-pointing arrow is located at the bottom of the form.

The Windows taskbar at the bottom shows the search bar 'Type here to search', several application icons (including Edge, File Explorer, Mail, and Word), and the system clock indicating 5:09 PM on 10-Feb-20.

Money Request and Acknowledgement:

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/Cybersecurity/viewdownaccountbalance.jsp'. The page content is a light gray area with a white modal box in the center. The modal box has a 'Back' button at the top. Below it, there are three sections: 'ACCOUNT TYPE' with a dropdown menu showing 'Saving and Current Status'; 'ACCOUNT NUMBER' with a dropdown menu showing 'Select An Account'; and 'Mini Statement' with a 'select period' section containing 'From dd-----yyyy' and 'To dd-----yyyy' fields. At the bottom of the modal is a 'View' button.

Database Screen Shot:

The screenshot shows a database management tool interface with the 'addaccount' table selected. The table structure is displayed in a table format with columns: #, Name, Datatype, Length/Set, Unsigned, Allow N..., Zerofill, Default, Comment, Collation, Expression, and Virtuality. The table has 13 columns and 13 rows of data.

#	Name	Datatype	Length/Set	Unsigned	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	Account_Name	VARCHAR	100	☐	☒	☐	NULL				
2	Account_No	BIGINT	20	☐	☒	☐	NULL				
3	Re_Ent_Acc_No	BIGINT	20	☐	☒	☐	NULL				
4	Account_Type	VARCHAR	100	☐	☒	☐	NULL				
5	IFSC_Code	VARCHAR	100	☐	☒	☐	NULL				
6	Branch_Name	VARCHAR	100	☐	☒	☐	NULL				
7	Branch_Code	VARCHAR	100	☐	☒	☐	NULL				
8	Account_Ope...	DATE		☐	☒	☐	NULL				
9	Email_ID	VARCHAR	100	☐	☒	☐	NULL				
10	Nick_Name	VARCHAR	100	☐	☒	☐	NULL				
11	Password	VARCHAR	100	☐	☒	☐	NULL				
12	address	VARCHAR	100	☐	☒	☐	NULL				
13	uniqueid	BIGINT	20	☐	☐	☒	AUTO_INCREME...				

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: accounttxn

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	fromacc	BIGINT	20	☐	☒	☐	NULL				
2	beneficiary	VARCHAR	100	☐	☒	☐	NULL				
3	benaccno	BIGINT	20	☐	☒	☐	NULL				
4	benaccno	VARCHAR	100	☐	☒	☐	NULL				
5	benific	VARCHAR	50	☐	☒	☐	NULL				
6	amount	BIGINT	20	☐	☒	☐	NULL				
7	remarks	VARCHAR	100	☐	☒	☐	NULL				
8	date	DATETIME		☐	☒	☐	NULL				
9	acctype	VARCHAR	100	☐	☒	☐	NULL				
10	txnid	BIGINT	20	☐	☐	☒	AUTO_INCREME...				

Help Discard Save

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: consolidatexn

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	uniqueid	BIGINT	20	☐	☒	☐	NULL				
2	nickname	VARCHAR	100	☐	☒	☐	NULL				
3	amount	BIGINT	20	☐	☒	☐	NULL				
4	customerid	BIGINT	20	☐	☒	☐	NULL				
5	accountno	BIGINT	20	☐	☒	☐	NULL				
6	status	VARCHAR	50	☐	☒	☐	NULL				

Help Discard Save

Host: 127.0.0.1 Database: bank Table: addrequest Data Query

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: addrequest

Comment:

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	uniqueid	BIGINT	20	☐	☒	☐	NULL				
2	nickname	VARCHAR	100	☐	☒	☐	NULL				
3	amount	BIGINT	20	☐	☒	☐	NULL				
4	customerid	BIGINT	20	☐	☒	☐	NULL				
5	status	VARCHAR	100	☐	☒	☐	NULL				
6	accountno	BIGINT	20	☐	☒	☐	NULL				
7	date	DATETIME		☐	☒	☐	NULL				

Help Discard Save

Basic Options Indexes Foreign keys Partitions CREATE code ALTER code

Name: addbusinessacc

Columns: Add Remove Up Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	businessowner	VARCHAR	100	☐	☒	☐	NULL				
2	businessname	VARCHAR	100	☐	☒	☐	NULL				
3	contactnumber	BIGINT	20	☐	☒	☐	NULL				
4	businessemail	VARCHAR	100	☐	☒	☐	NULL				
5	businessaddr1	VARCHAR	200	☐	☒	☐	NULL				
6	businessaddr2	VARCHAR	200	☐	☒	☐	NULL				
7	businessaddr3	VARCHAR	200	☐	☒	☐	NULL				
8	typeofbusiness	VARCHAR	100	☐	☒	☐	NULL				
9	country	VARCHAR	100	☐	☒	☐	NULL				
10	payment	VARCHAR	100	☐	☒	☐	NULL				
11	empname	VARCHAR	100	☐	☒	☐	NULL				
12	empaccountno	BIGINT	20	☐	☒	☐	NULL				
13	renempaccou...	BIGINT	20	☐	☒	☐	NULL				
14	benfacttype	VARCHAR	100	☐	☒	☐	NULL				
15	benfisc	VARCHAR	100	☐	☒	☐	NULL				
16	benbranch	VARCHAR	100	☐	☒	☐	NULL				
17	benbranchco...	VARCHAR	100	☐	☒	☐	NULL				
18	benfopedate	DATE		☐	☒	☐	NULL				
19	Nick_Name	VARCHAR	100	☐	☒	☐	NULL				
20	benmailid	VARCHAR	100	☐	☒	☐	NULL				

Help Discard Save

5.6.CODINGS

Frontend Coding:

Homepage:

```
<!--A Design by W3layouts
Author: W3layout
Author URL: http://w3layouts.com
License: Creative Commons Attribution 3.0 Unported
License URL: http://creativecommons.org/licenses/by/3.0/
-->
<!DOCTYPE html>
<html lang="en">
<head>
<title>Net Banking a Banking Category Flat Bootstrap responsive Website Template | Home ::
w3layouts</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="Net Banking Responsive web template, Bootstrap Web
Templates, Flat Web Templates, Android Compatible web template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson,
Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function()
{ setTimeout(hideURLbar, 0); }, false); function hideURLbar(){ window.scrollTo(0,1); }
</script>
<!-- bootstrap-css -->
<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
<!--// bootstrap-css -->
<!-- css -->
<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
<!--// css -->
<link rel="stylesheet" href="css/owl.carousel.css" type="text/css" media="all">
<link href="css/owl.theme.css" rel="stylesheet">
<link type="text/css" rel="stylesheet" href="css/cm-overlay.css" />
<!-- font-awesome icons -->
<link href="css/font-awesome.css" rel="stylesheet">
<!-- //font-awesome icons -->
<!-- font -->
<link href="//fonts.googleapis.com/css?family=Poppins:300,400,500,600,700" rel="stylesheet">
<link
href="//fonts.googleapis.com/css?family=Open+Sans:400,300,300italic,400italic,600,600italic,7
00,700italic,800,800italic" rel='stylesheet' type='text/css'>
<!-- //font -->
```

```

<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/bootstrap.js"></script>
<script type="text/javascript">
    jQuery(document).ready(function($) {
        $(".scroll").click(function(event){
            event.preventDefault();
            $('html,body').animate({scrollTop:$(this.hash).offset().top},1000);
        });
    });
</script>
<!-- animation -->
<link href="css/animate.css" rel="stylesheet" type="text/css" media="all">
<script src="js/wow.min.js"></script>
    <script>
        new WOW().init();
    </script>
<!-- //animation -->
<script>
$(document).ready(function() {
    $('#owl-demo').owlCarousel({

        autoPlay: 3000, //Set AutoPlay to 3 seconds
        autoPlay:true,
        items : 3,
        itemsDesktop : [640,5],
        itemsDesktopSmall : [414,4]

    });

});
</script>

</head>
<body>
    <!-- banner -->
    <div class="banner">
        <!--header-->
        <div class="header">
            <div class="container">
                <nav class="navbar navbar-default">
                    <div class="navbar-header">
                        <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1">
                            <span class="sr-only">Toggle
navigation</span>
                            <span class="icon-bar"></span>

```

```

        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <h1><a href="index.html">Net

<span>Banking</span></a></h1>

</div>
<!--navbar-header-->
<div class="collapse navbar-collapse" id="bs-example-
navbar-collapse-1">

    <ul class="nav navbar-nav navbar-right">
        <li><a href="index.jsp"
class="active">Home</a></li>
        <li><a href="Accounts.jsp"
class="active">Accounts</a></li>
        <li><a href="BeneficiaryLogin.jsp"
class="active">Transaction</a></li>
        <li><a href="Currency slide.jsp"
class="active">CurrencySlide</a></li>
        <li><a href="News.jsp"
class="active">News</a></li>
        <li><a href="adminLogin.jsp"
class="active">Admin</a></li>
        <li><a href="Contact1.jsp"
class="active">Contact</a></li>
    </ul>
    <div class="clearfix"> </div>
</div>
</nav>
</div>
</div>
<!--//header-->
<div class="w3layouts-banner-info">
    <div class="container">
        <div class="w3layouts-banner-slider">
            <div class="slider">
                <div class="callbacks_container">
                    <ul class="rslides callbacks callbacks1"
id="slider4">
                        <li>
                            <div class="agileits-banner-
info">
                                <h3>Get reliable help
                                with your credit card processing needs</h3>
                                <p>Donec tellus
                                metus, ornare et mollis ut, maximus id nisi. Quisque scelerisque accumsan sem nec ullamcorper.

```

Cras sed purus eget augue egestas commodo. Sed erat magna, pharetra aliquet mattis mollis, maximus eget lacus. </p>

</div>

<div class="agileits-banner-

info">

<h3>Accept Credit

Cards On Your Smartphone Today!</h3>

<p>Sed erat magna,

pharetra aliquet mattis mollis, maximus eget lacus. Donec tellus metus, ornare et mollis ut, maximus id nisi. Quisque scelerisque accumsan sem nec ullamcorper. Cras sed purus eget augue egestas commodo. </p>

</div>

</div>

<script src="js/responsiveslides.min.js"></script>

<script>

// You can also use

"\$(window).load(function() {"

\$(function () {

// Slideshow 4

\$("#slider4").responsiveSlides({

auto: **true,**

pager:**true,**

nav:**false,**

speed: 500,

namespace: "callbacks",

before: **function () {**

\$('.events').append("before

event fired.");

},

after: **function () {**

\$('.events').append("after event

fired.");

}

});

});

</script>

<!--banner Slider starts Here-->

</div>

</div>

</div>

</div>

```

        <div class="bounce animated">
            <a href="#welcome" class="scroll">
                <div class="mouse"></div>
            </a>
        </div>
    </div>
<!-- //banner -->
<!-- welcome -->
<div class="welcome" id="welcome">
    <div class="container">
        <div class="w3-welcome-heading">
            <h2>Welcome</h2>
        </div>
        <div class="w3l-welcome-info">
            <div class="col-sm-6 welcome-grids">
                <div class="welcome-img">
                    
                </div>
            </div>
            <div class="col-sm-6 welcome-grids">
                <div class="welcome-img">
                    
                </div>
            </div>
        </div>
        <div class="clearfix"> </div>
    </div>
    <div class="w3l-welcome-text">
        <p><u>Nam libero tempore cum soluta nobis est eligendi optio
cumque nihil impedit quo minus id quod maxime placeat facere possimus omnis optio cumque
nihil impedit quo minus id quod maxime placeat facere possimus.Sed ut perspiciatis unde omnis
iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque
ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae </p>
    </div>
</div>
<!-- //welcome -->
<!-- about -->
<div class="about" id="about">
    <div class="container">
        <div class="w3-welcome-heading">
            <h3>About Us</h3>
        </div>
        <div class="w3ls-about-grids">
            <div class="col-md-6 about-right">

```

```

        
    </div>
    <div class="col-md-6 about-left">
        <h4>Sed tincidunt lorem </h4>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed tincidunt lorem sed velit fermentum lobortis, eget placerat mauris sed lectus tellus
        <span> Fusce eu semper lacus, sodales id elit a, feugiat
porttitor nulla. Sed porta magna vitae nisl vulputate lacinia.</span></p>
        <ul>
            <li><span class="glyphicon glyphicon-share-
alt"></span> Duis aute irure dolor in reprehenderit voluptate </li>
            <li><span class="glyphicon glyphicon-share-
alt"></span> Excepteur sint occaecat cupidatat non proident</li>
            <li><span class="glyphicon glyphicon-share-
alt"></span> Sunt in culpa qui officia deserunt mollit </li>
            <li><span class="glyphicon glyphicon-share-
alt"></span> Duis aute irure dolor in reprehenderit voluptate </li>
            <li><span class="glyphicon glyphicon-share-
alt"></span> Excepteur sint occaecat cupidatat non proident</li>
        </ul>
    </div>
    <div class="clearfix"> </div>
</div>
</div>
</div>
<!-- //about -->
<!-- services -->
<div class="services" id="services">
    <div class="container">
        <div class="w3-welcome-heading">
            <h3>Our services</h3>
        </div>
        <div class="agileits-services-grids">
            <div class="col-md-8 agileits-services-left">
                <h4>Services Overview</h4>
                <div class="agileits-services-text">
                    <p>Nam libero tempore cum soluta nobis est
eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus omnis
optio cumque nihil impedit quo minus id quod maxime placeat facere possimus.</p>
                </div>
                <div class="credit-grids">
                    <h5>Credit Cards:</h5>
                    <div class="credit-grid-left">
                        <div class="credit-grid">
                            
                            <h6>Visa</h6>

```

nobis est</p>

<p>Nam libero tempore cum soluta

</div>

<div class="credit-grid">

<h6>MasterCard</h6>

nobis est</p>

<p>Nam libero tempore cum soluta

</div>

<div class="credit-grid">

<h6>MasterCard</h6>

nobis est</p>

<p>Nam libero tempore cum soluta

</div>

<div class="clearfix"> </div>

</div>

<div class="credit-grids debit-grids">

<h5>Debit Cards:</h5>

<div class="debit-grids-text">

<p>Morbi nec justo ut ex rhoncus luctus.

Duis id ex egestas, tempus lorem sed, porta urna. Duis sodales eleifend laoreet. Vestibulum luctus venenatis massa, in vulputate mi porta ac.</p>

</div>

<div class="w3-services-grids">

<div class="col-md-4 w3l-services-grid">

<div class="w3ls-services-img">

<i class="fa fa-money" aria-

hidden="true"></i>

</div>

<div class="agileits-services-info">

<h4>Praesent tempor</h4>

</div>

</div>

<div class="col-md-4 w3l-services-grid">

<div class="w3ls-services-img">

<i class="fa fa-credit-card"

aria-hidden="true"></i>

</div>

<div class="agileits-services-info">

<h4>Praesent tempor</h4>

</div>

</div>

<div class="col-md-4 w3l-services-grid">

<div class="w3ls-services-img">

```

aria-hidden="true"></i>
<i class="fa fa-line-chart"
</div>
<div class="agileits-services-info">
<h4>Praesent tempor</h4>
</div>
</div>
<div class="clearfix"> </div>
</div>
</div>
<div class="col-md-4 agileits-services-right">
<h4>Marketing Services</h4>
<div class="services-two-grids">
<div class="agileinfo-services-right">

<h6>Maecenas sollicitudin eros lectus, a
rutrum nisi vulputate quis. Proin tempus, lectus vitae gravida suscipit</h6>
</div>
<!-- date -->
<div id="design" class="date">
<div id="cycler">
<div
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-
hidden="true"></i> Lorem ipsum dolor sit amet,</a>
</div>
<div
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-
hidden="true"></i> Cras vestibulum dapibus <span class="blinking"></span></a>
</div>
<div
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-
hidden="true"></i> Fusce id molestie mauris</a>
</div>
<div
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-

```

```

hidden="true"></i> Aliquam vulputate arcu enim <span class="blinking"></span></a>
</div>
<div>
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-
hidden="true"></i> Donec ac turpis aliquet</a>
</div>
<div>
class="date-text">
<a
href="#" data-toggle="modal" data-target="#myModal"><i class="fa fa-arrow-right" aria-
hidden="true"></i> Morbi nec justo ut ex rhoncus luctus</a>
</div>
</div>
</div>
<script>
function blinker() {
    $('.blinking').fadeOut(500);
    $('.blinking').fadeIn(500);
    1000);
</script>
<script>
function
cycle($item, $cycler){
    setTimeout(cycle, 2000, $item.next(), $cycler);
    $item.slideUp(1000,function(){
    $item.appendTo($cycler).show();
    });
}
cycle($('#cycler div:first'), $('#cycler'));
</script>
</div>
<!-- //date -->
</div>
</div>
<div class="clearfix"> </div>

```

```

        </div>
    </div>
</div>
<!-- //services -->
<!-- works -->
<div class="jarallax blog" id="blog">
    <div class="container">
        <div class="w3-welcome-heading">
            <h3>Our Blog</h3>
        </div>
        <div class="wthree-blog-grids">
            <div class="col-md-6 w3-agileits-blog-grid">
                <div class="col-sm-4 blog-left">
                    <h4>24/09</h4>
                    <ul>
                        <li>Posted : <a href="#">Admin</a></li>
                        <li>Likes : 231</li>
                        <li>Comments : (24)</li>
                    </ul>
                </div>
                <div class="col-sm-8 blog-right">
                    <a href="#" data-toggle="modal" data-
target="#myModal">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque imperdiet
luctus gravida.</a>
                    <p>Sed ut feugiat felis. Morbi tempor ut nibh id
consectetur. Morbi turpis arcu, interdum at feugiat non, iaculis id dui. Cras vestibulum dapibus
volutpat. Fusce id molestie mauris, nec pulvinar nibh. </p>
                </div>
            </div>
            <div class="clearfix"> </div>
        </div>
        <div class="col-md-6 w3-agileits-blog-grid">
            <div class="col-sm-4 blog-left">
                <h4>09/12</h4>
                <ul>
                    <li>Posted : <a href="#">Admin</a></li>
                    <li>Likes : 102</li>
                    <li>Comments : (14)</li>
                </ul>
            </div>
            <div class="col-sm-8 blog-right">
                <a href="#" data-toggle="modal" data-
target="#myModal">Etiam ut nibh quis magna placerat euismod ac id eros. Cras eleifend nulla
vel eros blandit.</a>
                <p>Duis sodales eleifend laoreet. Vestibulum luctus
venenatis massa, in vulputate mi porta ac. Maecenas sollicitudin eros lectus, a rutrum nisi
vulputate quis. Proin tempus, lectus vitae gravida</p>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="clearfix"> </div>
    </div>
    <div class="clearfix"> </div>
</div>
</div>
<!-- //works -->
<!-- news -->
<div class="news" id="news">
    <div class="container">
        <div class="w3-welcome-heading">
            <h3>News & Events</h3>
        </div>
        <div class="w3ls-news-grids">
            <div class="news-right">
                <div class="col-md-4 news-right-grid">
                    <div class="agile-news-info">
                        
                        <h4><a href="#" data-toggle="modal" data-
target="#myModal">Sunt in culpa qui officia velit</a></h4>
                        <span>19th June | 10:00 - 12:00</span>
                        <p> Integer interdum eros vitae sem ultrices,
sed eleifend tellus tincidunt. Nam nisl arcu, porttitor sit amet</p>
                        <div class="agileinfo-news-button">
                            <a href="#" class="hvr-shutter-in-
horizontal" data-toggle="modal" data-target="#myModal">More</a>
                        </div>
                    </div>
                </div>
                <div class="col-md-4 news-right-grid">
                    <div class="agile-news-info">
                        
                        <h4><a href="#" data-toggle="modal" data-
target="#myModal">Neque porro quisquam est</a></h4>
                        <span>24th Sept | 09:00 - 11:00</span>
                        <p> Integer interdum eros vitae sem ultrices,
sed eleifend tellus tincidunt. Nam nisl arcu, porttitor sit amet</p>
                        <div class="agileinfo-news-button">
                            <a href="#" class="hvr-shutter-in-
horizontal" data-toggle="modal" data-target="#myModal">More</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="col-md-4 news-right-grid">
  <div class="agile-news-info">
    
    <h4><a href="#" data-toggle="modal" data-
target="#myModal">Etiam ut nibh quis magna</a></h4>
    <span>04th Oct | 12:00 - 02:00</span>
    <p> Integer interdum eros vitae sem ultrices,
sed eleifend tellus tincidunt. Nam nisl arcu, porttitor sit amet</p>
    <div class="agileinfo-news-button">
      <a href="#" class="hvr-shutter-in-
horizontal" data-toggle="modal" data-target="#myModal">More</a>
    </div>
  </div>
</div>
<div class="clearfix"> </div>
</div>
<div class="clearfix"> </div>
</div>
<!-- modal -->
<div class="modal about-modal fade" id="myModal" tabindex="-
1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-
title"><span>Banking</span></h4>
      </div>
      <div class="modal-body">
        <div class="agileits-w3layouts-
info">
          
          <p>Duis venenatis, turpis eu
bibendum porttitor, sapien quam ultricies tellus, ac rhoncus risus odio eget nunc. Pellentesque ac
fermentum diam. Integer eu facilisis nunc, a iaculis felis. Pellentesque pellentesque tempor enim,
in dapibus turpis porttitor quis. Suspendisse ultrices hendrerit massa. Nam id metus id tellus
ultrices ullamcorper. Cras tempor massa luctus, varius lacus sit amet, blandit lorem. Duis auctor
in tortor sed tristique. Proin sed finibus sem.</p>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    <!-- //modal -->
</div>
</div>
<!-- //news -->
<!-- feedback -->
<div class="jarallax feedback">
    <div class="container">
        <div class="w3-welcome-heading">
            <h3>Clients Feedback</h3>
        </div>
        <div class="agileits-feedback-grids">
            <div id="owl-demo" class="owl-carousel owl-theme">
                <div class="item">
                    <div class="feedback-info">
                        <div class="feedback-top">
                            <p> Sed semper leo metus, a lacinia
eros semper at. Etiam sodales orci sit amet vehicula pellentesque. </p>
                        </div>
                        <div class="feedback-grids">
                            <div class="feedback-img">
                                
                            </div>
                            <div class="feedback-img-info">
                                <h5>Mary Jane</h5>
                                <p> Vestibulum</p>
                            </div>
                            <div class="clearfix"> </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="item">
        <div class="feedback-info">
            <div class="feedback-top">
                <p> Sed semper leo metus, a lacinia
eros semper at. Etiam sodales orci sit amet vehicula pellentesque. </p>
            </div>
            <div class="feedback-grids">
                <div class="feedback-img">
                    
                </div>
                <div class="feedback-img-info">
                    <h5>Peter guptill</h5>
                    <p> Vestibulum</p>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="clearfix"> </div>
    </div>
</div>
<div class="item">
    <div class="feedback-info">
        <div class="feedback-top">
            <p> Sed semper leo metus, a lacinia
eros semper at. Etiam sodales orci sit amet vehicula pellentesque. </p>
        </div>
        <div class="feedback-grids">
            <div class="feedback-img">
                
            </div>
            <div class="feedback-img-info">
                <h5>Steven Wilson</h5>
                <p>Vestibulum</p>
            </div>
        <div class="clearfix"> </div>
    </div>
</div>
<div class="item">
    <div class="feedback-info">
        <div class="feedback-top">
            <p> Sed semper leo metus, a lacinia
eros semper at. Etiam sodales orci sit amet vehicula pellentesque. </p>
        </div>
        <div class="feedback-grids">
            <div class="feedback-img">
                
            </div>
            <div class="feedback-img-info">
                <h5>Mary Jane</h5>
                <p>Vestibulum</p>
            </div>
        <div class="clearfix"> </div>
    </div>
</div>
<div class="item">
    <div class="feedback-info">
        <div class="feedback-top">

```



```

        <div class="w3-agileits-subscribe-form">
            <form action="#" method="post">
                <input type="text" placeholder="Subscribe"
name="Subscribe" required="">
                <button class="btn1">Subscribe</button>
            </form>
        </div>
    </div>
</div>
<!-- //subscribe -->
<!-- map -->
<div class="map-grid">
    <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1859251.8642025779!2d-
76.08274894689792!3d40.06224332601239!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m
2!1s0x89c0fb959e00409f%3A0x2cd27b07f83f6d8d!2sNew+Jersey%2C+USA!5e0!3m2!1sen!2si
n!4v1474436926209" style="border:0" allowfullscreen></iframe>
    </div>
<!-- //map -->
<!-- footer -->
<div class="jarallax footer">
    <div class="container">
        <div class="footer-logo">
            <h3><a href="index.html">Net <span>Banking</span></a></h3>
        </div>
        <div class="agileinfo-social-grids">
            <h4>We are social</h4>
            <div class="border"></div>
            <ul>
                <li><a href="#"><i class="fa fa-facebook"></i></a></li>
                <li><a href="#"><i class="fa fa-twitter"></i></a></li>
                <li><a href="#"><i class="fa fa-rss"></i></a></li>
                <li><a href="#"><i class="fa fa-vk"></i></a></li>
            </ul>
        </div>
        <div class="copyright">
            <p>Â© 2017 Net Banking. All rights reserved | Design by <a
href="http://w3layouts.com">W3layouts</a></p>
        </div>
    </div>
</div>
<!-- //copyright -->
<script src="js/jarallax.js"></script>
<script src="js/SmoothScroll.min.js"></script>
<script type="text/javascript">
    /* init Jarallax */

```

```

        $('jarallax').jarallax({
            speed: 0.5,
            imgWidth: 1366,
            imgHeight: 768
        })
    </script>
<script type="text/javascript" src="js/move-top.js"></script>
<script type="text/javascript" src="js/easing.js"></script>
<!-- here stars scrolling icon -->
<script type="text/javascript">
    $(document).ready(function() {
        /*
            var defaults = {
                containerID: 'toTop', // fading element id
                containerHoverID: 'toTopHover', // fading element hover id
                scrollSpeed: 1200,
                easingType: 'linear'
            };
        */

        $.UItoTop({ easingType: 'easeOutQuart' });

    });
</script>
<!-- //here ends scrolling icon -->
<script src="js/owl.carousel.js"></script>
</body>
</html>

```

Backend Codings:

Login:

```

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class Login
 */
@WebServlet("/Login")
public class Login extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Login() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // TODO Auto-generated method stub
        Map map = new HashMap();
        HttpSession ses = request.getSession();
        List customerId = new ArrayList<>();
        try {
            String user = request.getParameter("email");
            String pass = request.getParameter("pass");
            String acctype = request.getParameter("acctype");
            Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bank","root","root");
            PreparedStatement ps=null;PreparedStatement ps1=null;
            if(acctype.equals("Personal")) {
                ps=con.prepareStatement("select * from addaccount where
Email_ID=? and Password=?");
            } else {
                ps=con.prepareStatement("select * from addbusinessacc where
benfmailid=? and password=?");
            }
            ps.setString(1, user);
            ps.setString(2, pass);
            ResultSet rs = ps.executeQuery();

```

```

        if(rs.next()) {
            ses.setAttribute("Nick_Name", rs.getString("Nick_Name"));
            ses.setAttribute("uniqueid", rs.getString("uniqueid"));
            if(acctype.equals("Personal")) {
                ses.setAttribute("Account_No",
rs.getString("Account_No"));
                ses.setAttribute("Account_Name",
rs.getString("Account_Name"));
            }else {
                ses.setAttribute("Account_No",
rs.getString("empaccountno"));
                ses.setAttribute("Account_Name",
rs.getString("empname"));
            }
            if(acctype.equals("Personal")) {
                ps1=con.prepareStatement("select * from addaccount
where uniqueid!="+rs.getString("uniqueid"));

            }else {
                ps1=con.prepareStatement("select * from addbusinessacc
where uniqueid!="+rs.getString("uniqueid"));
            }
            ResultSet rs1 = ps1.executeQuery();
            while(rs1.next()) {
                customerId.add(rs1.getString("uniqueid"));
            }
            ses.setAttribute("customerId", customerId);
            response.sendRedirect("Txnaccounts.jsp");
        }else {
            response.sendRedirect("Login.jsp");
        }
    }catch(Exception e) {
        e.printStackTrace();
    }
}

}

```

Accounts:

```

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

```

```

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;


import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


/**
 * Servlet implementation class AddBusinessAcc
 */
@WebServlet("/AddBusinessAcc")
public class AddBusinessAcc extends HttpServlet {

    private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddBusinessAcc() {

        super();

        // TODO Auto-generated constructor stub
    }

```

```

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    HttpSession ses = request.getSession();
    String a=request.getParameter("businessowner");
    String b=request.getParameter("businessname");
    String c=request.getParameter("contactnumber");
    String d=request.getParameter("businesemail");
    String e=request.getParameter("businessaddr1");
    String f=request.getParameter("businessaddr2");
    String g=request.getParameter("businessaddr3");
    String h=request.getParameter("typeofbusiness");
    String i=request.getParameter("country");
    String j=request.getParameter("payment");
    String k=request.getParameter("empname");
    String l=request.getParameter("empaccountno");
    String m=request.getParameter("renempaccountno");
    String n=request.getParameter("benfacctype");
    String o=request.getParameter("benfifsc");
    String p=request.getParameter("benfbranch");

```

```

String q=request.getParameter("benfbranchcode");

Date r=Date.valueOf(""+request.getParameter("benfopedate"));

String s=request.getParameter("benfnickname");

String w=request.getParameter("benfemail");

String x=request.getParameter("benfpass");


try{

    Class.forName("com.mysql.jdbc.Driver");

    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/bank","root","root");


    PreparedStatement ps=con.prepareStatement("insert into
addbusinessacc(businessowner,businessname,contactnumber,businesemail,businessaddr1,busine
ssaddr2,businessaddr3,typeofbusiness,country,payment,empname,empaccountno,renempaccount
no,benfacctype,benfifsc,benfbranch,benfbranchcode,benfopedate,Nick_Name,benfmailid,passwo
rd) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");


    ps.setString(1,a);

    ps.setString(2,b);

    ps.setString(3,c);

    ps.setString(4,d);

    ps.setString(5,e);

    ps.setString(6,f);

    ps.setString(7,g);

    ps.setString(8,h);

    ps.setString(9,i);

```

```
ps.setString(10,j);
ps.setString(11,k);
ps.setString(12,l);
ps.setString(13,m);
ps.setString(14,n);
ps.setString(15,o);
ps.setString(16,p);
ps.setString(17,q);
ps.setDate(18,r);
ps.setString(19,s);
ps.setString(20,w);
ps.setString(21,x);

int i1=ps.executeUpdate();
if(i1>0) {
    ses.setAttribute("msg", "Successfully Added");
    response.sendRedirect("AddBusinessSucc.jsp");
}

}catch (Exception e2) {
    e2.printStackTrace();
}

out.close();
}}
```

CHAPTER-6

6.1. CONCLUSION:

Our scope is to We conduct theoretical analysis and extensive experiments on different datasets with different learners, and our approaches show promising results. In this we observe also existing concepts' like e computational and memory efficient randomized feature selection algorithms based on semi-randomized selection, warm up, and cool down techniques also been analyzed by this we get better result than previous system. , we apply our proposed feature selection approach on real datasets. In this application, we apply our algorithms for polymer property prediction through data's; it is a crucial step of data-driven machine learning approach where the geometric and chemical information on the polymers are converted to a numerical representation.

6.2. FUTURE WORK:

- We are planed parsing the object like natural language to framework process.
- Coming day's we can implement this concept are works without any support of framework they reduce some time and business logics.
- They can be map configuration so it's easier to maintain the process in multiple server side.

References :

- 1.E.M.S.W Balagolla,W.P.C Fernando,R.M.N.S Rathnayake, M.J.M.R.P Wijesekera, A. N. Senarathne, K.Y. Abeywardhana” Credit Card Fraud Prevention Using Blockchain” May 2021.
- 2.Zheng and H. Dal.” An Overview of Blockchain Technology: Architecture Consensus and Future Trends” October 2018.
- 3.Qi M. Dabbagh, M. Sookhar and N. Safaulin Wang.” The Evolution of Blockchain: A Bibliometric Study” December 2019.
- 4.A. Kosba, A. Miller and Z. Wen, Hawk.” The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts” November 2020.
- 5.W. Yaokumah and G. Ntow-Danso” Credit Card Fraud Prevention” August 2020.
- 6.Michał Ostapowicz,Kamil Żbikowski,Detecting Fraudulent Accounts on Blockchain,A Supervised Approach,29 October 2019
- 7.A Blockchain Framework for Insurance Processe,s 02 April 2018,Mayank Raikwar; Subhra Mazumdar
- 8.Vukolić and Marko,Rethinking permissioned blockchains,17,Dec 2017.
- 9.Smart contract templates: essential requirements and design options,12 Aug 2016,C. D. Clack, V. A. Bakshi and L. Braine.
- 10.Blockchains and smart contracts for the internet of things,20 May 2016,K. Christidis and M. Devetsikiotis
- 11.Blockchain contract: Securing a blockchain applied to smart contracts,Watanabe, S. Fujimura, April2016.
- 12.Blockchain or not blockchain that is the question of the insurance and other sectors,F. Lamberti, V. Gatteschi,Dec 2017.
- 13.Blockchain cryptography and consensus,C,Christian,Aug2017
- 14.Architecture of the hyperledger blockchain fabric, Aug 2016.
- 15.Cryptography and protocols in hyperledger fabric 6 January, 2017,E. Androulaki, C. Cachin.
- 16.Survey of consensus protocols on blockchain applications,L. S. Sankar, M. Sindhu,Jan 2017.

