

AES ENCRYPTION FOR ALLERGEN-SAFE COSMETIC MANUFACTURING PROCESS ON SUPPLIER CONTAINERS

*Report submitted to the SASTRA Deemed to be University
in partial fulfillment of the requirements
for the award of the degree of*

Bachelor of Technology

Submitted by

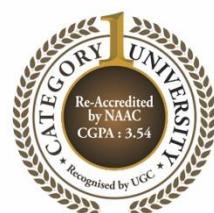
NAME: GEETIKA CHANDRASHEKHAR

(Reg. No.: 124014010, Information and Communication Technology)

May 2024



SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJAVUR, TAMIL NADU, INDIA-613 401



SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJA VUR- 613 401

Bonafide Certificate

This is to certify that the project report titled “AES Encryption for Allergen-Safe Cosmetic Manufacturing Process on Supplier Containers” submitted in partial fulfillment of the requirements for the award of the degree of B. Tech. Information and Communication Technology to the SASTRA Deemed to be University, is a bona-fide record of the work done by Ms. Geetika Chandrashekhar(Reg. No. 124014010) during the final semester of the academic year 2023-24, in the School of Computing, under my supervision. This report has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other similar title to any candidate of any University.

Signature of the Project Supervisor : 

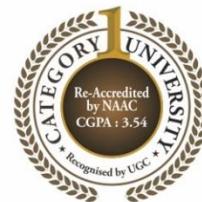
Name with Affiliation : Dr. Lavanya M, AP-III / SOC

Date : 22-04-2024

Mini Project *Viva voce* held on _____

Examiner 1

Examiner 2



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING
THANJA VUR- 613 401

Declaration

I declare that the project report titled “AES Encryption for Allergen-Safe Cosmetic Manufacturing Process on Supplier Containers” submitted by me is an original work done by me under the guidance of Dr. Lavanya M, Assistant Professor III, School of Computing, SASTRA Deemed to be University during the final semester of the academic year 2023-24, in the School of Computing. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, or other similar title to any candidate of any University.

Signature of the candidate : 

Name of the candidate : Geetika Chandrashekhar

Date : 22-04-2024

Acknowledgements

I would like to thank our Honourable Chancellor **Prof. R. Sethuraman** for providing me with an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

I would like to thank our Honourable Vice-Chancellor **Dr.S. Vaidhyasubramaniam** and **Dr. S. Swaminathan**, Dean, Planning & Development, for the encouragement and strategic support at every step of our college life.

I extend my sincere thanks to **Dr. R. Chandramouli**, Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

I extend my heartfelt thanks to **Dr. V. S. Shankar Sriram**, Dean, School of Computing. **Dr. R. Muthaiah**, Associate Dean, Research, **Dr. K. Ramkumar**, Associate Dean, Academics, **Dr. D. Manivannan**, Associate Dean, Infrastructure, **Dr. R. Algeswaran**, Associate Dean, Students Welfare.

My guide **Dr. Lavanya M**, Assistant Professor -III, School of Computing was the driving force behind this whole idea from the start. Her deep insight in the field and invaluable suggestions helped me in making progress throughout our project work. I also thank the project review panel members for their valuable comments and insights which made this project better.

I would like to extend our gratitude to all the teaching and non-teaching faculties of the School of Computing who have either directly or indirectly helped us in the completion of the project.

I gratefully acknowledge all the contributions and encouragement from my family and friends resulting in the successful completion of this project. I thank you all for providing me an opportunity to showcase my skills through project.

List of Figures

Figure No.	Title	Page No.
3.1	System Architecture	13
3.2	Entity Relationship Diagram of Project	14
3.3	Data Flow Diagram of Implementation Idea	16
3.4	Use Case Diagram of Flow	17
3.5	Class Diagram of Project	18
6.1	blc_chain table	33
6.2	login table	33
6.3	encry_manuf table	34
6.4	microbial_test table	34
6.5	supplprod_finalres table	35
6.6	supp_log table	35
6.7	supplier_cont table	36
6.8	Supplierres table	36
6.9	testing_res table	36
6.10	cont_qualityinsp table	37

Abbreviations

CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
SHA	Secure Hashing Algorithm
AES	Advanced Encryption Standard
JS	JavaScript
JSP	Java Server Pages
CGI	Common Gateway Interface
HTTP	Hyper Text Transfer Protocol
SGML	Standard Generalized Markup Language
WWW	World Wide Web
IAM	Identity and Access Management
MFA	Multi-Factor Authentication
ERD	Entity Relationship Diagram
DFD	Data Flow Diagram

Abstract

The proposed cosmetic production system introduces a groundbreaking paradigm shift by integrating advanced blockchain technology with robust encryption standards, specifically SHA-512 and AES, to address existing challenges in transparency, security, and safety within the cosmetics manufacturing industry. The proposed solution adheres to the current industry requirements, while also setting new standards for data integrity, traceability, and customer. SHA-512, a cryptographic hash technique that produces a 64-byte hash value that is resistant to collisions, strengthens the foundation of security for the system. Every stage of the process, from obtaining ingredients to formulation, testing, quality control, and container delivery, is painstakingly recorded in the blockchain's unchangeable ledger. It functions as a transparent, unalterable record. Microbial testing procedures are strategically integrated to detect potential allergens, a critical addition that enhances consumer safety. This meticulous approach aligns with the current industry need for product authenticity, ethical sourcing, and safety, setting the stage for future industry norms. In contrast to the existing cosmetics manufacturing sector, which relies on traditional and less secure systems, the proposed blockchain-based system provides comprehensive solutions. It addresses the transparency and security demands of modern consumers by ensuring data security and transparency. Furthermore, the system facilitates end-to-end traceability. The suggested technique greatly increases overall data security and improves the openness of the sourcing, testing, and quality assurance procedures for ingredients. By strengthening the system against unwanted access and manipulation, encryption techniques of AES and SHA-512 lessen the dangers that come with the existing dependence on less secure technologies.

Specific Learning:

- Website Development(in Java) and deploying in Apache Tomcat server
- Tech Stack (Frontend – HTML, CSS, JavaScript; Backend – Servlets and JSP, Database - HeidiSQL)
- Implementation of SHA 256 Algorithm and AES Algorithm

KEYWORDS: *Cosmetic manufacturing, Blockchain, SHA 256, Java, AES*

Table of Contents

Title	Page No.
Bonafide Certificate	ii
Declaration	iii
Acknowledgments	iv
List of Figures	v
Abbreviations	vi
Abstract	vii
1. Introduction	1
2. System Development Environment	5
2.1 Introduction To Java	5
2.2 Domain Knowledge	5
2.3 Eclipse	7
2.4 HTML	7
2.5 CSS	8
2.6 JS	8
2.7 APACHE TOMCAT SERVER	8
2.8 SQL	9
2.9 JSP and SERVLETS	9
2.10 JDBC	10
3. System Design	11
3.1 Modules	11
3.2 E – R Diagrams	13
3.3 Flow Diagrams	14
3.4 Use Case Diagram	17
3.5 Class Diagrams	18
4. Hardware & Software Requirements	19
5. Source Code	20
6. Database Table Structures	33
7. Output Snapshots	38
8. Conclusion and Future Plans	48
9. References	49

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

A new method of making cosmetics is being developed in response to the modern demands of the cosmetic industry for increased transparency and quality control. Modern blockchain technology, strengthened by AES and SHA-512 encryption standards, is easily integrated into this creative system, guaranteeing unmatched data security and integrity throughout the whole production cycle. Complete transparency and traceability are ensured by its careful recording and safe storage of data on ingredient sourcing, formulation, quality control, testing, and container supply. In addition, the system incorporates stringent microbiological testing procedures to detect and reveal any allergies, hence augmenting customer security and confidence. The system faces problems such as installation costs, data transfer, regulatory compliance, and reluctance to change, but delivering previously unheard-of benefits in transparency, security, and safety. These issues must be resolved for the system to be successfully used in the cosmetic industry. By solving the shortcomings of the current system, especially concerning transparency, data security, and allergy disclosure, the suggested blockchain-based solution transforms the cosmetic production process. The system enables customers to access full information on the legitimacy and safety of cosmetic products by utilizing blockchain technology and strict encryption requirements to ensure that every stage of cosmetic manufacture is properly recorded and safely maintained. By detecting and revealing possible allergens, stringent microbiological testing procedures increase consumer confidence and empower allergy sufferers to make educated decisions. The suggested approach has benefits like data security, transparency, and allergy testing, however, its effective use in the cosmetics business will require careful implementation and planning.

1.2 SCOPE OF PROJECT

The goal of this project is to completely restructure the cosmetics production process using cutting-edge technology and best practices to satisfy the ever-evolving needs of the cosmetics industry. Blockchain technology is used with the AES and SHA-512 encryption standards to offer unparalleled data security and integrity across the whole cosmetic production process. The project's scope is expanded to include the careful

documentation and safekeeping of information related to the procurement of ingredients, formulation, quality assurance, testing, and container delivery. To evaluate and reveal any possible allergenic content in the cosmetic product, it also entails the application of stringent microbiological testing procedures, which improves safety and transparency. It also seeks to improve accountability, openness, and trust within the ecosystem of cosmetic manufacturing, fulfilling the rising demand for product authenticity, ethical sourcing, and safety. By setting new industry benchmarks, this project is not only poised to meet consumer expectations but also to pioneer a future where responsible, trustworthy, and allergen-safe cosmetic manufacturing practices prevail.

1.3. EXISTING SYSTEM

The cosmetics manufacturing sector is mostly dependent on traditional production and record-keeping systems. Procurement of ingredients, formulation, quality control, and testing processes are carried out using standard techniques, with little data traceability. Paper records and centralized databases are the main methods of storing data, which leaves the possibility for data manipulation and security flaws. The extensive openness and security that modern customers expect from the cosmetics sector are absent from the current system. Even though it's frequently regulated, ingredient sourcing sometimes lacks comprehensive information on suppliers and their credentials. Since manual procedures are used throughout the formulation and testing stages, comprehensive product traceability is difficult to offer. Furthermore, there may be gaps in safety and transparency because the system only automatically handles allergen material. In essence, the cosmetic manufacturing industry, in its current state, faces limitations in data security, transparency, and safety measures, which the proposed blockchain-based system with microbial testing seeks to address comprehensively.

DISADVANTAGES:

- **Lack of Transparency:** Customers find it difficult to learn the provenance and certifications of the substances used in their cosmetics since the current system is opaque regarding the source of these ingredients.
- **Data Vulnerability:** Because data is kept in paper records and centralized systems, it is vulnerable to alteration and illegal access. This jeopardizes the accuracy of vital data about the manufacturing process.

- **Limited Traceability:** Because the traditional system does not offer end-to-end traceability for cosmetic items, it is challenging to follow a product's whole path from the source of its ingredients to its destination.
- **Inadequate Allergen Testing:** The current system does not routinely conduct allergen testing, which might lead to the incorrect identification of cosmetic items containing allergens and the possibility of allergic responses.

1.4. PROPOSED SYSTEM

The proposed system is a creative approach to cosmetic production that seeks to rectify the present system's flaws and usher in a new era of transparency, security, and safety. It creates an impenetrable environment with state-of-the-art blockchain technology reinforced by the unbreakable SHA-512 encryption standard and AES encryption. Every step of the manufacturing process for cosmetics, including sourcing materials, creating product formulas, testing, guaranteeing quality, and providing packaging, is meticulously recorded and securely stored on the blockchain's immutable ledger. An essential upgrade to the proposed system is the inclusion of comprehensive microbiological testing protocols. The purpose of these tests is to find and reveal any possible allergies in cosmetic items, giving customers thorough knowledge about the product. The inclusion of allergen testing not only ensures customer well-being but also aligns the cosmetic industry with modern demands for allergen disclosure and safety assurance.

ADVANTAGES:

- **Enhanced Transparency:** The suggested approach makes full use of blockchain technology to provide total transparency in the production of cosmetics. Customers are encouraged to have faith in the authenticity of products by having access to comprehensive information on ingredient source, quality assurance, and testing.
- **Unprecedented Data Security:** The suggested system offers the greatest levels of data security and integrity protection thanks to SHA-512 and AES encryption. As a result, there is far less chance of data breaches, manipulation, or illegal access.
- **Allergen Testing:** It is possible to identify and reveal possible allergens in cosmetic goods by incorporating stringent microbial testing procedures. This guarantees customer safety and enables allergy sufferers to make wise decisions.

- **Detailed Supply Chain Visibility:** The suggested method uses blockchain technology to capture and retain data regarding certifications, sustainability initiatives, and container suppliers. This comprehensive approach extends transparency to the entire supply chain.

CHAPTER 2

SYSTEM DEVELOPMENT ENVIRONMENT

The infrastructure, resources, and tools needed to develop, test, and implement software applications are collectively referred to as the system development environment. It includes all individuals engaged in the development process as well as hardware and software. For the purpose of developing, coding, debugging, and maintaining software systems, the environment offers a regulated setting.

2.1 INTRODUCTION TO JAVA

James Gosling started the original development of the Java programming language at Sun Microsystems, and it was eventually launched in 1995 as a key part of the company's Java platform. The language was originally known as "Oak," but in 1995 it was renamed as "Java." This language was primarily driven by the requirement for a platform-independent language. Lastly, Java is the language of choice for Internet programming, just as C was for system programming. Java is an advanced object-oriented programming language. A high-level language program cannot be directly executed on any computer. It must first be converted into that specific machine language. This is accomplished via the Java Compiler, which converts Java programs (.java files with source code within) into machine code. Java Virtual Machine (JVM) is a virtual machine that resides in the real machine (your computer) and the machine language for JVM is byte code. JVM executes the byte code generated by the compiler and produces output. JVM is the one that makes Java platform independent.

2.2 DOMAIN KNOWLEDGE

Cybersecurity, also goes by the names "cybersecurity" or "cybersecurity," is the act of protecting digital dangers such as loss, theft, and breaches from computer networks, devices, and data. It consists of a wide range of instruments, protocols, rules, and safeguards to safeguard IT systems and the information they transport or store.

- **Information security:** A key component of cybersecurity is safeguarding the availability, confidentiality, and integrity of data. This entails safeguarding data integrity, limiting access, and encrypting important information.

- **Application security** is the process of protecting software applications from potential attacks by locating and fixing vulnerabilities.
- **Management of Identity and Access (IAM):** Controlling access to systems and data, including devices and users, usually by using multi-factor authentication (MFA) or more sophisticated authentication approaches like usernames and passwords.
- **Security Awareness and Training:** Providing users and staff with information on security best practices and possible risks will help to lower the likelihood of human mistakes and make them less vulnerable to social engineering attacks. Investigation, containment, and recovery are all part of incident response and recovery, which involves getting ready for and handling security events, breaches, or data losses.

SHA

Websites frequently employ SHA-256 cryptographic hashing to ensure data security and integrity. Passwords or files are used as input data, creating a unique 256-bit hash value from them making them computationally impossible to decipher the original input. Websites utilize SHA-256 to hash passwords, safely storing user information in databases. It is also used in digital signatures to confirm the legitimacy and consistency of material on websites, increasing user and server confidence. SHA-256 is also essential to SSL/TLS protocols because it generates cryptographic keys that secure data transfer and establish secure connections, protecting private data from unwanted access.

AES

Advanced Encryption Standard, or AES, is frequently used in MySQL for both data encryption and decryption. It has built-in methods that let users encrypt and decrypt data inside the database, such as `AES_ENCRYPT()` and `AES_DECRYPT()`. MySQL's AES encryption offers variable security levels by supporting key lengths of 128, 192, or 256 bits. This feature improves data safety in MySQL applications by enabling the safe transfer and storage of sensitive data. MySQL secures confidentiality and integrity by using AES encryption, protecting sensitive data from alteration or unauthorised access.

BLOCKCHAIN

The term "blockchain immutable ledger concept" refers to the inability to erase or change data after it has been recorded. Consensus procedures and cryptographic hashing provide the data's immutability while guaranteeing its integrity and reliability. A chain of unchangeable records is created when a transaction or record is added to the blockchain and cryptographically connected to earlier blocks. Across dispersed networks, this capability offers a transparent and impenetrable method for storing data and its verification.

2.3 ECLIPSE

For Java web development, the Eclipse IDE is an adaptable tool that provides thorough support for Java EE technologies such as servlets, JSP, and JSF. The development process is improved by its integrated debugger, allowing for real-time code inspection and debugging. Making use of Eclipse's vast plugin ecosystem, well-known web frameworks like Spring and Hibernate can be integrated with ease, giving developers access to more features and resources. Additionally, Eclipse has built-in support for code management and collaborative development for version control systems like Git. Because of its intuitive UI and strong technical support, Eclipse is still the go-to tool for Java developers creating complex and scalable websites.

2.4 HTML

The World Wide Web (WWW) uses Hypertext Markup Language (HTML) to enable users to create web pages with text, images, and hyperlinks to other websites. HTML is an application of SGML (Standard Generalized Markup Language), an ISO standard 8879 modified for the Web and specifically tailored for hypertext. It is not a programming language. Hypertext's one point to another point concept. We can browse the content according to our preferences and areas of interest. All that is needed to show a markup language is a collection of things encased in elements. Links that lead to different documents or specific sections of the same content are called hyperlinks and are indicated works.

Any kind of document may be shown using HTML on the host computer, which may be located anywhere in the world. This language is flexible and may be utilized on any desktop.

HTML is platform-neutral and offers tags, or special codes, to enhance the appearance of the content. HTML tags don't care about the case.

Utilizing images, typefaces, varying sizes, colors, and other elements can improve how the material is presented. Everything that isn't a tag is a component of the document.

2.5 CSS

Cascading Style Sheets, or CSS, is a stylistic language that manages how web publications appear visually. It specifies the arrangement, hues, typefaces, and additional aesthetic features of HTML components. With CSS, developers may better maintain and increase flexibility by separating a webpage's display from its content. Designers may use CSS to produce visually appealing and responsive websites that work on a range of screens and devices. Because of its cascading structure, styles may be inherited and overridden, enabling effective and modular code organization.

2.6 JAVASCRIPT

JavaScript is a small, object-oriented programming language used to create web applications for clients and servers. Statements inserted directly into an HTML page using JavaScript are interpreted by Netscape Navigator 2.0. Additionally, Livewire lets you write server-based applications that resemble CGI (common gateway interface) programs.

For instance, a JavaScript function can confirm that users fill out forms that ask for phone numbers or zip codes with accurate information. An HTML page having embedded JavaScript can interpret text entered by the user and notify them via a message dialog if the input is invalid, or it can use JavaScript to trigger an action (like playing an audio file, launching an applet, or interacting with a plug-in) when the user opens or closes a page. All of this functionality can be done without requiring any network transmission.

2.7 APACHE TOMCAT SERVER

Java web apps are deployed using Apache Tomcat, an open-source web server and servlet container. It offers a stable environment for executing dynamic online content by implementing the Java Servlet, JavaServer Pages (JSP), WebSocket, and Java Expression Language (EL) requirements. Because of its lightweight architecture, which offers options for performance optimization and scalability, Tomcat is appropriate for use in

both development and production contexts. Java-based web applications are hosted by many using Apache Tomcat, which has a large documentation base and vibrant community support. It is frequently used as a reverse proxy and to handle static material, when combined with Nginx or Apache HTTP Server.

2.8 MySQL AND HEIDI SQL

MySQL is a relational database management system (RDBMS) that is available for free and is renowned for its performance, scalability, and dependability. Because it can handle and manipulate data using SQL (Structured Query Language), it may be used for multiple purposes, from small-scale initiatives to enterprise-level solutions. MySQL is a popular database used in web development that powers dynamic webpages and online apps. In contrast, HeidiSQL is a well-liked open-source client for MySQL, PostgreSQL, and Microsoft SQL Server database management. It offers an easy-to-use interface for managing data, viewing database architecture, and running SQL queries. With features like code completion, syntax highlighting, and database export/import capabilities, HeidiSQL is a useful tool for both database administrators and developers.

Together, MySQL and HeidiSQL form a powerful combination for building, managing, and maintaining relational databases. Developers leverage MySQL for its robust database engine, while HeidiSQL provides a convenient interface for interacting with MySQL databases efficiently.

2.9 JSP AND SERVLETS

A general server extension is called a Servlet. Java classes are dynamically loaded to increase a server's capacity. Web servers are frequently used with servlets, where CGI scripts can be replaced by them. A servlet is a portable and secure alternative to a proprietary server extension since it operates inside the server's Java Virtual Machine (JVM). Servlets function only inside the server's domain. Individual threads inside the web server process handle each servlet, in contrast to CGI and Fast CGI, which employs distinct processes to handle individual programs or requests. This indicates that all servlets are scalable and efficient. Servlets are cross-platform, meaning they may be used on different web servers and operating systems. The finest platform available for developing web applications is provided by Java Servlets. CGI scripts are replaced with servlets on a web server to increase the functionality of any type of server. A mail server

with servlet support, for instance, might have its capabilities expanded to include virus scanning of all attached documents or mail filtering tasks.

Servlets offer a Java-based answer to the issues that come with server-side development nowadays, such as missing interfaces, platform-specific APIs, and inextensible scripting solutions.

Objects that connect to a Java-based server and follow a certain interface are called servlets. Servlets are to the server side, if applets are to the client-side object byte codes that might be dynamically loaded from the internet. They vary from applets in that they are faceless and do not have a graphical user interface. They work as pluggable, platform-independent, dynamically loadable auxiliary byte code objects on the server side, which may be used to dynamically extend server-side functionality.

2.10 JDBC

A Java API called JDBC makes it possible to create a single application that runs SQL statements. It is made up of many Java programming language interfaces and classes. Because JDBC provides a standard API, database developers may use it to construct database applications with a pure Java API.

By utilizing JDBC, sending SQL statements to practically any software that can send SQL statements to the correct database is easy. Combining Java and JDBC enables programmers to write code once and execute it anywhere.

The `java.sql.Driver` interface is implemented by a particular JDBC driver that provides access to a single database system. Almost all common RDBMS systems provide drivers, although most are not free. To provide access to conventional ODBC data sources, such as a Microsoft Access database, Sun includes a free JDBC-ODBC bridge driver with the JDK. However, Sun cautions against utilizing the bridge driver for anything beyond development and extremely restricted development.

CHAPTER 3

SYSTEM DESIGN

The proposed blockchain-based system represents a transformative shift in cosmetic manufacturing, addressing key limitations of the traditional system while introducing innovative solutions to enhance transparency, data security, and allergen disclosure. By leveraging cutting-edge blockchain technology fortified by SHA-512 and AES encryption standards, the system ensures unparalleled data security and integrity throughout the production cycle. It meticulously records and securely stores data on ingredient sourcing, formulation, quality control, testing, and container supply, offering complete transparency and traceability. Rigorous microbial testing processes provide an enhancement in the proposed system, which identifies and discloses potential allergens. This not only enhances consumer safety and trust but also aligns the cosmetic industry with modern demands for allergen disclosure and safety assurance. The technology also provides previously unheard-of benefits in terms of safety, security, and openness. For the system to be successfully used in the cosmetics sector, there are a few issues that must be resolved.

These include implementation costs, data migration, regulatory compliance, and resistance to change. Careful consideration of these challenges and stakeholder resistance is essential for the successful implementation and adoption of the proposed system in the cosmetic industry.

3.1 MODULES

- Admin
- Manufacture
- Testing
- Management
- Blockchain

Admin Module:-

Allows login to the page. The user logs in to the module after registering, following which they are sent to the home page. There are menu options like view and verify the client from Status. This module provides permission as well as keys for handling operations in the other modules.

Manufacture Module:-

In the manufacturing module, manufacturers upload CSV files containing product details, including raw materials and quantities. These files are forwarded to testers who verify compliance with consumer standards and conduct microbial testing to ensure allergen safety. If products fail the testing phase, they are returned to the manufacturer for remanufacturing.

Management Module:-

Register their Management client Name, and password within this module, then confirm the password to log in for the send request for admin. If the admin approves the user, they will receive the password, and the page will be redirected to the home page when they log into the module. The first step is to upload the product and send it to the testing module, the updation on different module manufacturers will be updated on the management module. Users can view the results of the report in each process.

Testing Module:-

Register your client's name, email address, and password in this module, then validate your password to access the page. If the user has registered, the page will redirect to the home page when they login to the module. There are menus to view the manufactured product, get the key from the management module, upload the testing details of the product, and send it to the management module if any of the given testing products fail.

Blockchain Module:-

Additionally, the servlet generates a unique batch ID for each manufacturing process and updates the database accordingly. It also encrypts sensitive data using AES encryption before storing it in the database table "encry_manuf". The servlet also handles the blockchain integration aspect. It computes the hash of each block of data using the SHA-256 hashing algorithm and stores this information in a table named "blc_chain". This process involves generating a previous hash, calculating the hash of the current block, and updating the database with this information. The servlet also updates the status of the process based on whether it's pending or completed. Furthermore, the servlet manages file

uploads, such as test reports or data files, associating them with specific processes and storing them in the database. Depending on type of data uploaded and the associated process (manufacturing, testing, internal data, or external data), the servlet redirects users to different web pages to provide feedback on the success or failure of the operation.

3.2 SYSTEM ARCHITECTURE

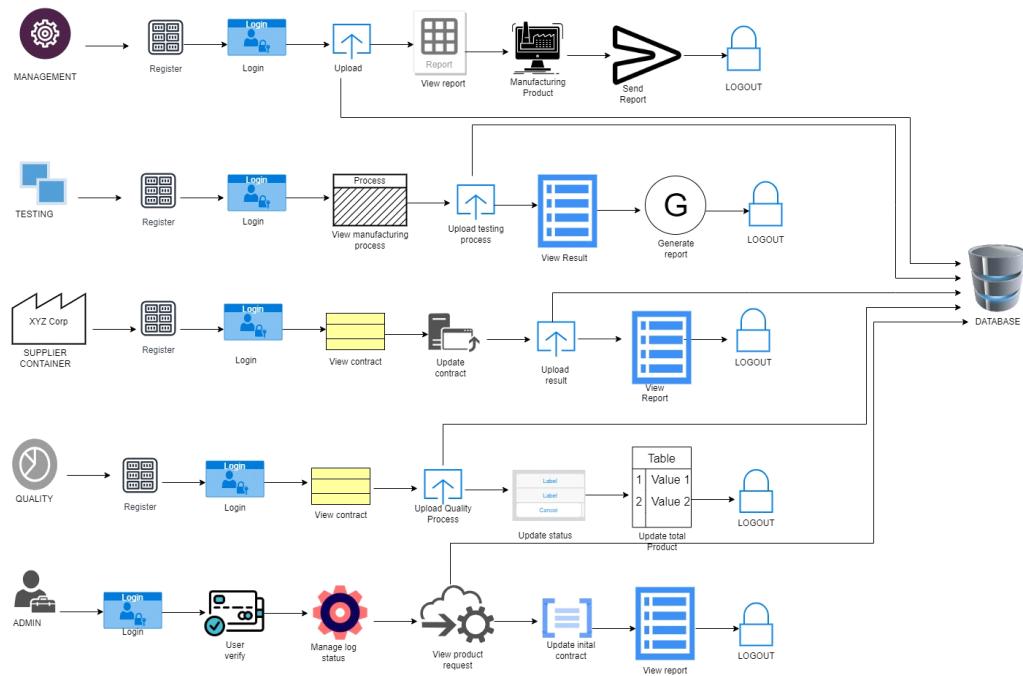


Fig. 3.1. System Architecture

3.3 E – R DIAGRAMS

A conceptual ER diagram forms the framework for the relationship within system, specifying entities already in place, the standard relations that sustain the system and the cardinalities required to maintain its current state. The Entity Relationship Diagram (ERD) displays the relationship between data items, the explanation of which can be found in the description of a data item. The ERD is the notation that is used to carry out the date modeling activity and the properties of each data object documented. The main objective of ERD is the representation of data items and their relationships.

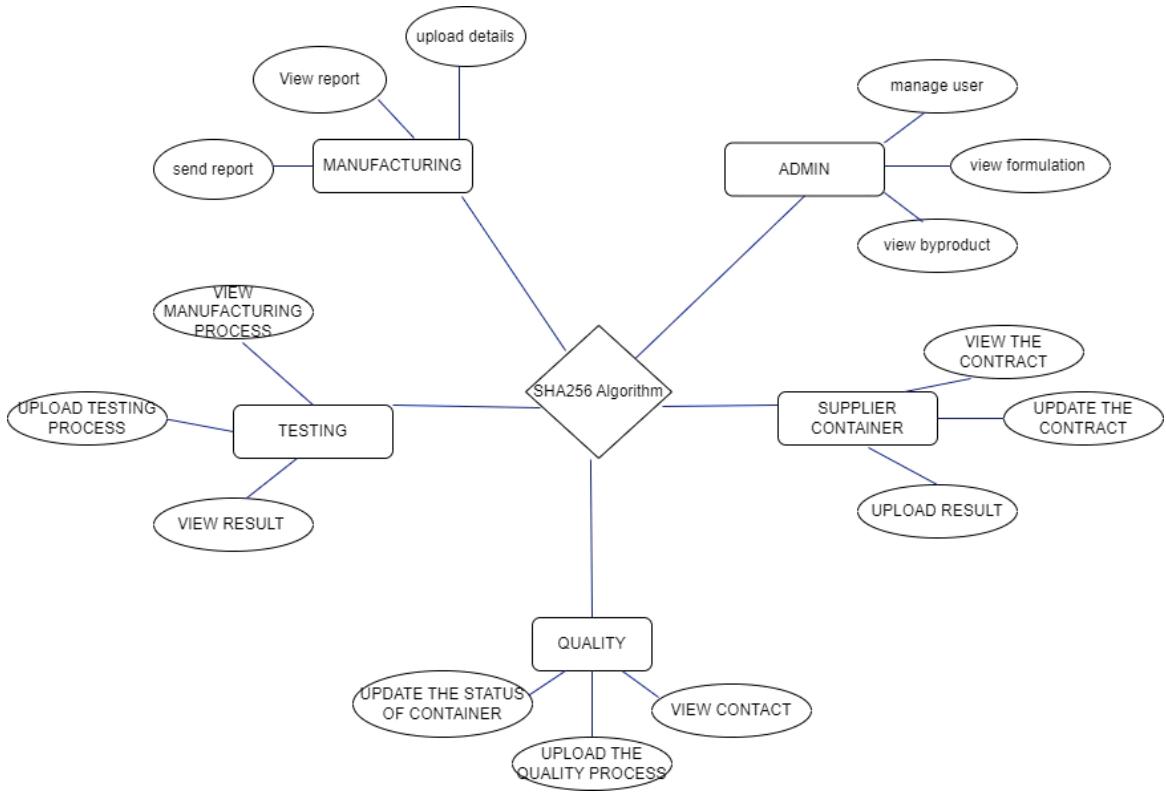


Fig. 3.2. Entity Relationship Diagram of Project

3.4 FLOW DIAGRAMS

A data flow diagram is a tool used for describing data movement across a system. Processing changes data from input to output. The term "logical data flow diagrams" is used to refer to them. The physical data flow diagrams show how actual tools and data move between people, departments, and workstations. A collection of data flow diagrams fully explains a system. Every procedure in the lower-level diagrams is deconstructed into a more intricate DFD at a higher level. The context diagram is also called a top-level diagram. A DFD's motive is to clarify system requirements and highlight significant changes that will be coded into the system design.

There are four symbols in the DFD listed below:-

- Square indicates where system data originates or ends.
- Data flow is shown by an arrow. It is the conduit by which data is sent.

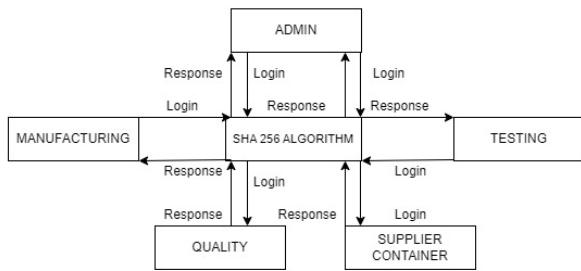
- A process that converts incoming data flows into exiting data flows is represented by a circle.
- An open rectangle can be used as a temporary data repository, data storage, or data at rest.

DATA FLOW

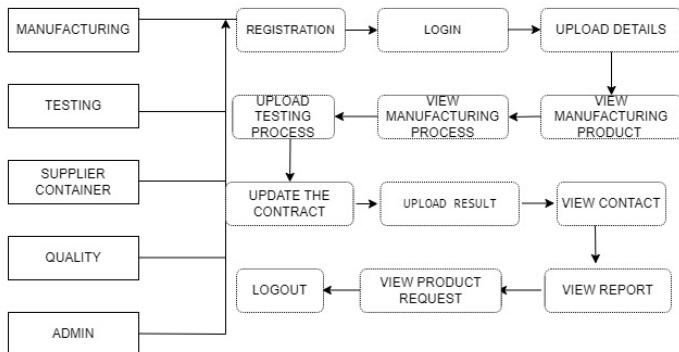
- In a data flow, there is only a single path of flow between symbols. However, process and data storage may allow a flow in both ways, showing a read before an update, depending on the workflow and data caching.
- A connection in DFD denotes that the same data arises via two or more different processes and is sunk or stored at a common location.
- A stream of data cannot instantly resume its original process. At least one additional process handling the data flow and creating another data flow has to return the original information from the first process.
- A data flow signals an update (removal or change) to a data store.
- A data flow can then be used to access or retrieve information from a data store.

Several data flows with the same noun phrase label can appear on the same arrow as long as they all travel together as a single bundle.

LEVEL0



LEVEL1



LEVEL 2

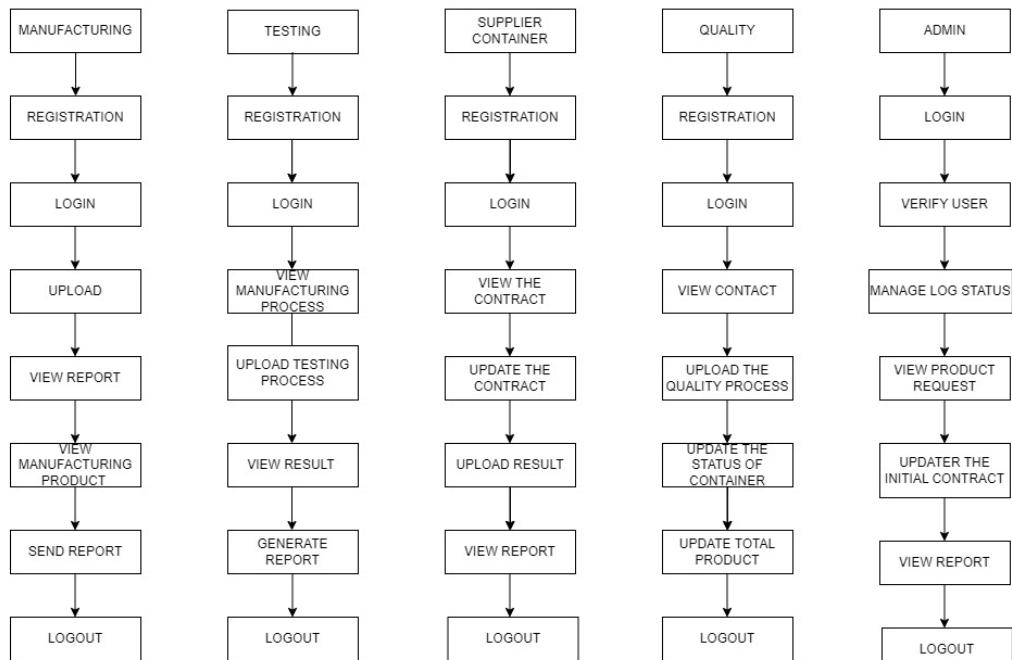


Fig. 3.3. Data Flow Diagram of Implementation Idea

3.5 USE CASE DIAGRAM

A use case diagram aids in the visualization of user-system interactions, the definition of system boundaries, and the identification of functional needs. By showing user objectives and system functioning, it facilitates the development process and offers a clear picture of how users interact with the system. It also acts as a communication tool for stakeholders, supporting requirements analysis and guaranteeing that user demands and system design are in sync.

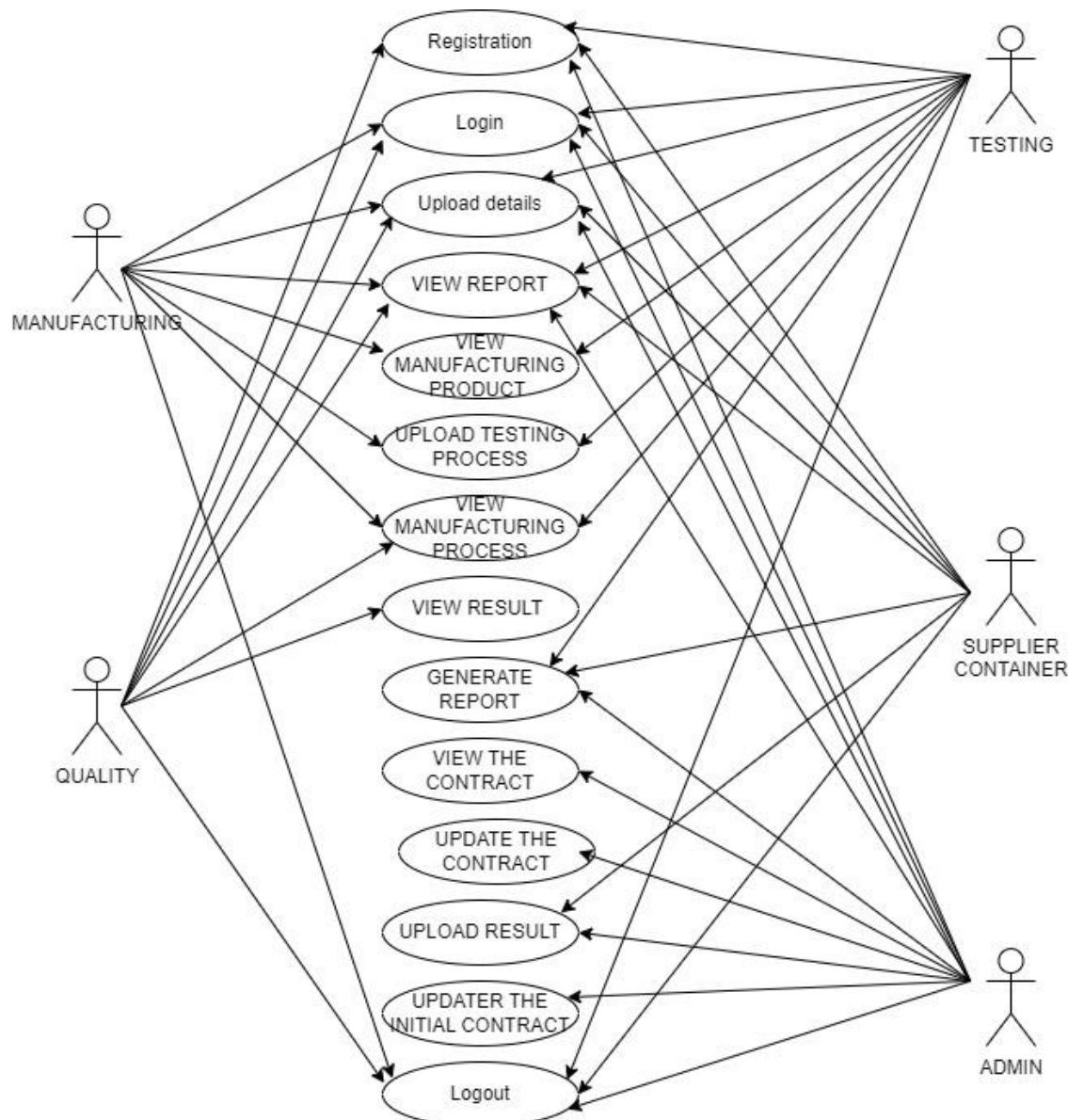


Fig. 3.4. Use Case Diagram of Flow

3.6 CLASS DIAGRAM

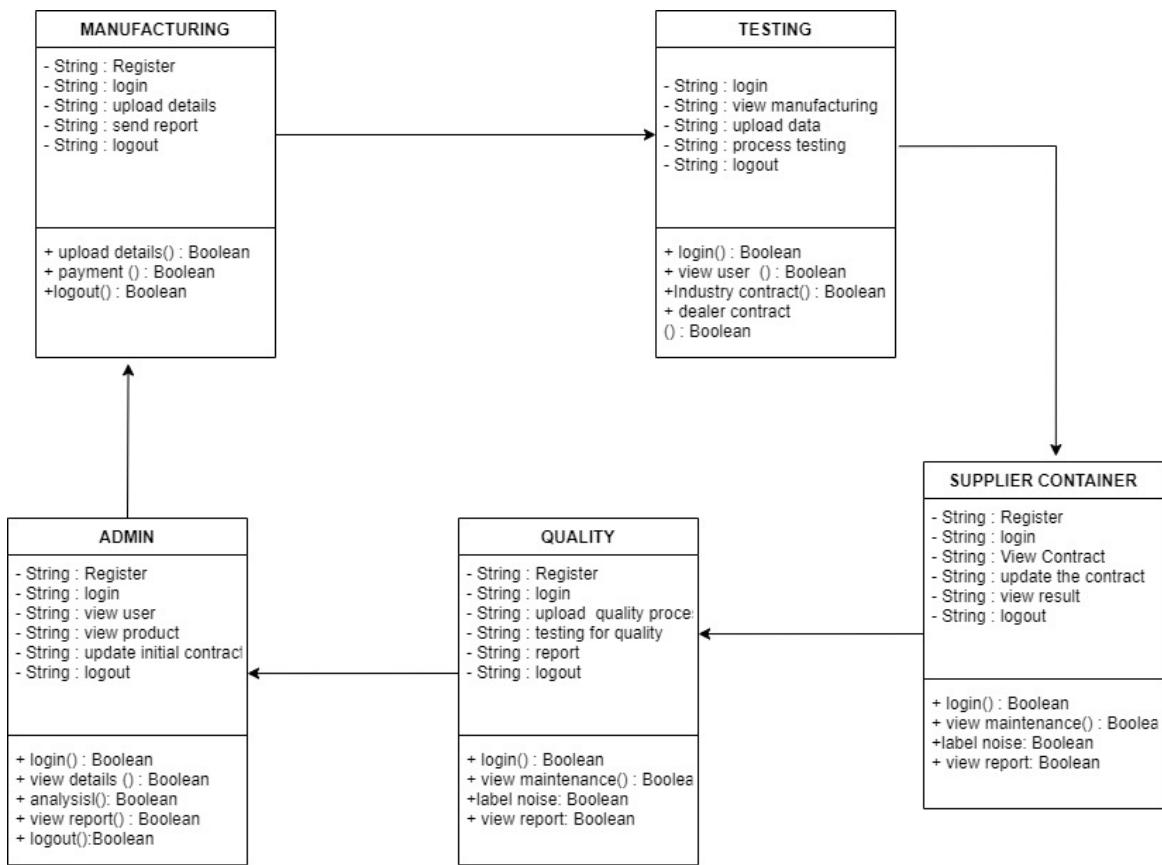


Fig. 3.5. Class Diagram of Project

CHAPTER 4

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, HTML,CSS, JS, Servlet
- **Web application** : J2EE Frameworks
- **Back end** : MySQL 5.1

CHAPTER 5

SOURCE CODE

Ad_log.java

```
① Ad_log.java ✘
1 package admin;
2
3*import java.io.IOException;□
11
12 /**
13 * Servlet implementation class Ad_log
14 */
15 @WebServlet("/Ad_log")
16 public class Ad_log extends HttpServlet {
17     private static final long serialVersionUID = 1L;
18     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         String mail=request.getParameter("email");
20         String pass=request.getParameter("pass");
21
22         if(mail.equalsIgnoreCase("admin@gmail.com") && pass.equalsIgnoreCase("admin"))
23         {
24             RequestDispatcher rd=request.getRequestDispatcher("ad_home.html");
25             rd.include(request, response);
26         }
27         else
28         {
29             RequestDispatcher rd=request.getRequestDispatcher("ad_log.jsp");
30             rd.include(request, response);
31         }
32     }
33
34
35
36 }
37
38 }
```

AdAccept.java

```
① AdAccept.java ✘
19 @WebServlet("/AdAccept")
20 public class AdAccept extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     /**
24      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
25      */
26     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27         // TODO Auto-generated method stub
28         PrintWriter pp=response.getWriter();
29         try
30         {
31             Class.forName("com.mysql.jdbc.Driver");
32             Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/cosmetic","root","root");
33             System.out.println("*****$$$$$$$$$11100000000000000000");
34             String str="update manuf set ad_status='All Completed'";
35             PreparedStatement st =cn.prepareStatement(str);
36             int i=st.executeUpdate();
37             if(i>0)
38             {
39                 pp.print("<html><body><script>alert('Update Sucessfull');</script></body></html>");
40                 RequestDispatcher rd=request.getRequestDispatcher("ad_home.html");
41                 rd.include(request, response);
42             }
43             else
44             {
45                 pp.print("<html><body><script>alert('Update Unsucessfull');</script></body></html>");
46                 RequestDispatcher rd=request.getRequestDispatcher("ad_home.html");
47                 rd.include(request, response);
48             }
49
50
51     }
```

Manuf_blc.java


```

String q = "INSERT INTO encry_manuf (Product_Name,Raw_Material,Qut_kg,batch_cost,Unit_Price,Cost_estim, batch_te
/* String qry2 = "insert into dataset2 select * from dataset1 ";*/
try {
    int ps2 = con1.prepareStatement(q).executeUpdate();
    if(ps2>0)
    {
        ses.setAttribute("id",id);
        ses.setAttribute("cname1",cname);
        RequestDispatcher rsd=request.getRequestDispatcher("manuf_hm.jsp");
        rsd.include(request, response);
    }
    else
    {
        RequestDispatcher rsd=request.getRequestDispatcher("manuf_hm.jsp");
        rsd.include(request, response);
    }
}

System.out.println("*****33");
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

231 String pre = "00000000000000000000000000000000";
232     // data
233     try{
234         Connection con = DB.getconnection();
235
236         Statement st = con.createStatement();
237         ResultSet rs = st.executeQuery("select * from blc_chain where id='"+id+"' and prehash ='"+ pre +
238
239         if(rs.next())
240         {
241             System.out.println("1111111111111111111111111111222222222222");
242             Statement st1 = con.createStatement();
243             ResultSet rs1 = st1.executeQuery("select * from blc_chain where id='"+id+"' order by count d
244
245             if(rs1.next())
246             {
247
248                 System.out.println("11111111111111111111111111222222222223333333333");
249                 String B = rs1.getString("count");
250                 Statement st2 = con.createStatement();
251                 ResultSet rs2 = st2.executeQuery("select * from blc_chain where id='"+id+"' and count='"
252
253                 if(rs2.next())
254                 {
255
256                     System.out.println("11111111111111111111111111222222222223333333333444");
257                     String previoushash = rs2.getString("nexthash");
258
259                     String tohash = previoushash;
260
261                     //hashing
262                     MessageDigest md = MessageDigest.getInstance("SHA-256");
263                     byte[] digest = md.digest(tohash.getBytes(StandardCharsets.UTF_8));
264                     String sha256 = DatatypeConverter.printHexBinary(digest).toLowerCase();
265                     String sts = "Report Generated";

```

```

266 int B2 = Integer.parseInt(B);
267 int B3 = B2+1;
268 int ids1=Integer.parseInt(id);
269 /*String s1 = "update sampledetails1 set status='"+sts+"' where id='"+uid+"'";
270     int r1 = DB.getConnection().prepareStatement(s1).executeUpdate();*/
271 PreparedStatement ps = con.prepareStatement("insert into blc_chain (id,prehash,nexthash,count,date,data,use
272     ps.setInt(1, ids1);
273     ps.setString(2, previoushash);
274     ps.setString(3, sha256);
275     ps.setString(4, "11");
276     ps.setString(5, strDateNew);
277     ps.setBinaryStream(6, inputStream,(int)filePart.getSize());
278     ps.setString(7, cname);
279     ps.setString(8, "pending");
280     ps.setString(9, batch);
281
282     int i = ps.executeUpdate();
283     if(i>0 && cname.equalsIgnoreCase("manuf")) {
284
285         String redirectURL = "manuf_hm.jsp";
286         out.println("<script>");
287         out.println("alert('Report uploaded successfully');");
288         out.println("window.location.href = '" + redirectURL + "'");
289         out.println("</script>");
290
291         out.print("<html><body><script>alert('Report uploaded successfully')</script></body></html>");
292         request.getRequestDispatcher("manuf_hm.jsp").include(request, response);
293
294     }
295     else if (i>0 && cname.equalsIgnoreCase("ManufTesting")){
296
297
298         String redirectURL = "testing_hm.html?p=" + id ;
299         out.println("<script>");
300         out.println("alert('Report uploaded successfully');");
301
302         out.println("alert('Report uploaded successfully');");
303         out.println("window.location.href = '" + redirectURL + "'");
304         out.println("</script>");
305
306         out.print("<html><body><script>alert('Report uploaded successfully')</script></body></html>");
307         request.getRequestDispatcher("testing_hm.html").include(request, response);
308
309     else if (i>0 && cname.equalsIgnoreCase("Internal_data")){
310         Connection con6=DB.getConnection();
311         String sts6 = "Completed";
312         String u = "UPDATE cl SET req_st= '"+sts6+"'";
313         PreparedStatement ps6 = con6.prepareStatement(u);
314         int rs6 = ps6.executeUpdate();
315
316         String redirectURL = "in_home.jsp?p=" + id ;
317         out.println("<script>");
318         out.println("alert('Report uploaded successfully');");
319         out.println("window.location.href = '" + redirectURL + "'");
320         out.println("</script>");
321
322         out.print("<html><body><script>alert('Report uploaded successfully')</script></body></html>");
323         request.getRequestDispatcher("in_home.jsp").include(request, response);
324
325     }
326
327 }
328
329

```

DB.java

```

DB.java
1 import java.io.IOException;
2
3 /**
4  * Servlet implementation class Database
5  */
6 @WebServlet("/DB")
7 public class DB extends HttpServlet {
8
9
10
11     private static final String URL = "jdbc:mysql://localhost:3306/cosmetic";
12
13     private static final String DRIVER = "com.mysql.jdbc.Driver";
14
15     private static final String USERNAME = "root";
16
17     private static final String PASSWORD = "root";
18
19     private static Connection connection = null;
20
21
22     public static Connection getConnection() throws SQLException {
23         if (connection != null)
24             return connection;
25         else {
26             try {
27                 Class.forName(DRIVER);
28                 connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
29             } catch (ClassNotFoundException e)
30             {
31                 e.printStackTrace();
32             }
33             catch (SQLException e)
34             {
35                 e.printStackTrace();
36             }
37             return connection;
38         }
39     }
40
41 }

```

LoginUser.java

```

LoginUser.java
1 package Login;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class LoginUser
7  */
8 @WebServlet("/LoginUser")
9 public class LoginUser extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
14     */
15     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16
17         HttpSession session = request.getSession();
18         PrintWriter pp = response.getWriter();
19         String e = request.getParameter("role");
20         String m = request.getParameter("email");
21         System.out.print("role===="+e);
22         String n = request.getParameter("pss");
23
24
25         System.out.println("role"+e+ "          pss"+n+ "          mail"+m);
26         try
27         {
28             Class.forName("com.mysql.jdbc.Driver");
29             Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/cosmetic","root","root");
30             PreparedStatement p3 =cn.prepareStatement("select * from login where email='"+m+"'");
31             ResultSet rr3 = p3.executeQuery();
32             if(rr3.next())
33             {
34                 PreparedStatement p1 =cn.prepareStatement("select id from login where email='"+m+"' and ad_status='pending'");
35                 ResultSet rr1 = p1.executeQuery();
36             }
37         }
38     }
39 }

```



```
153         rd.include(request,response);
154     }
155     }
156   }
157 }
158 catch(Exception e1)
159 {
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 else
170 {
171 pp.print("<html><body><script>alert('Still User is Not Registered');</script></body></html>");
172 RequestDispatcher rd=request.getRequestDispatcher("LogReg.jsp?role="+e+"");
173 rd.forward(request,response);
174 }
175 }
176 }catch(Exception ex)
177 {
178 ex.printStackTrace();
179 pp.close();
180 }
181 }
182 }
183 }
184 }
185 }
186 }
```

Qr_scanner.java

```
Qr_scanner.java ✘
1 package manuf;
2
3 import java.awt.image.BufferedImage;
4
5 public class Qr_scanner {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         String data ="" ;
10
11         String filePath = "D://beauty_prod2//beauty_prod//WebContent//Qrcode//.png";
12         int width = 250;
13         int height = 250;
14         generateQRCode(data, filePath, width, height);
15         System.out.println("QR code generated successfully.");
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40         String decodedData = scanQRCode(filePath);
41         System.out.println("Decoded QR code data: " + decodedData);
42     }
43
44     public static void generateQRCode(String data, String filePath, int width, int height) {
45         try {
46             Map<EncodeHintType, Object> hints = new HashMap<>();
47             hints.put(EncodeHintType.CHARACTER_SET, "UTF-8");
48
49             QRCodeWriter qrCodeWriter = new QRCodeWriter();
50             BitMatrix bitMatrix = qrCodeWriter.encode(data, BarcodeFormat.QR_CODE, width, height, hints);
51
52             BufferedImage qrImage = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
53             for (int x = 0; x < width; x++) {
54                 for (int y = 0; y < height; y++) {
55                     qrImage.setRGB(x, y, bitMatrix.get(x, y) ? 0xFF000000 : 0xFFFFFFFF);
```

```

47         hints.put(EncodeHintType.CHARACTER_SET, "UTF-8");
48
49         QRCodeWriter qrCodeWriter = new QRCodeWriter();
50         BitMatrix bitMatrix = qrCodeWriter.encode(data, BarcodeFormat.QR_CODE, width, height, hints);
51
52         BufferedImage qrImage = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
53         for (int x = 0; x < width; x++) {
54             for (int y = 0; y < height; y++) {
55                 qrImage.setRGB(x, y, bitMatrix.get(x, y) ? 0xFF000000 : 0xFFFFFFFF);
56             }
57         }
58         File qrFile = new File(filePath);
59         ImageIO.write(qrImage, "png", qrFile);
60
61     } catch (WriterException | IOException e) {
62         e.printStackTrace();
63     }
64 }
65 public static String scanQRCode(String filePath) {
66     try {
67         BufferedImage qrImage = ImageIO.read(new File(filePath));
68         BinaryBitmap binaryBitmap = new BinaryBitmap(new HybridBinarizer(new BufferedImageLuminanceSource(qrImage)));
69         MultiFormatReader reader = new MultiFormatReader();
70         Result result = reader.decode(binaryBitmap);
71
72         return result.getText();
73     } catch (Exception e) {
74         e.printStackTrace();
75     }
76     return null;
77 }
78
79
80 }
81

```

Cal_Res.java

```

1 package testing;
2
3 import java.io.IOException;
4
5 /**
6  * Servlet implementation class Cal_Res
7  */
8 @WebServlet("/Cal_Res")
9 public class Cal_Res extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
14     */
15     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
16     {
17         // TODO Auto-generated method stub
18         int result=0;
19         PrintWriter ou=response.getWriter();
20         Random r=new Random();
21         int max=120;
22         int min=170;
23         int count=0;
24         try
25         {
26
27             Class.forName("com.mysql.jdbc.Driver");
28             Connection cn2.DriverManager.getConnection("jdbc:mysql://localhost:3306/cosmetic","root","root");
29             String str="select * from manuf where ad_status='testProcessing'";
30             PreparedStatement st3 =cn2.prepareStatement(str);
31             ResultSet rss=st3.executeQuery();
32             while(rss.next())
33             {
34                 String q="insert into microbial_test(prod_name) values('"+rss.getString(2)+"')";
35                 PreparedStatement st4 =cn2.prepareStatement(q);
36             }
37         }
38     }
39
40
41
42
43
44
45
46
47
48

```

```
Cal_Res.java
48     PreparedStatement st4 =cn2.prepareStatement(q);
49     result=st4.executeUpdate();
50
51 }
52 if(result>0)
53 {
54     String mt="select * from microbial_test";
55     PreparedStatement pp =cn2.prepareStatement(mt);
56     ResultSet rt=pp.executeQuery();
57
58     while(rt.next())
59     {
60         count++;
61         int rand1= (int)Math.floor(Math.random() * (max - min + 1) + min);
62         int rand2= (int)Math.floor(Math.random() * (max - min + 1) + min);
63         String upd="update microbial_test set id='"+count+"',start_range='"+rand1+"',critical_range='"+rand2+
64             PreparedStatement PP2 =cn2.prepareStatement(upd);
65             PP2.executeUpdate();
66     }
67     ou.print("<html><body><script>alert('Sucessfull ') </script></body></html>");
68     RequestDispatcher rd = request.getRequestDispatcher("testing_hm.html");
69     rd.include(request, response);
70
71 }
72 else
73 {
74     ou.print("<html><body><script>alert('Sucessfull ') </script></body></html>");
75     RequestDispatcher rd = request.getRequestDispatcher("testing_hm.html");
76     rd.include(request, response);
77
78 }
79
80 }
81
82 catch(Exception e)
```

Index.html

```
index.html ✘
22
23<header>
24<section class="header">
25<div class="banner">
26<div class="w3_banner">
27<div class="container">
28<nav class="navbar navbar-expand-lg navbar-light bg-light">
29    <h1><a class="navbar-brand" href="#">Cosmetics</a></h1>
30    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-cont
31        <span class="navbar-toggler-icon"></span>
32    </button>
33
34<div class="collapse navbar-collapse" id="navbarSupportedContent">
35    <ul class="navbar-nav ml-auto">
36        <li class="nav-item active">
37            <a class="nav-link" href="index.html">Home </a>
38        </li>
39        <li class="nav-item active">
40            <a class="nav-link" href="Login.jsp?role=manuf">Manufacture</a>
41        </li>
42        <li class="nav-item active">
43            <a class="nav-link" href="Login.jsp?role=quality">Quality</a>
44        </li>
45        <li class="nav-item active">
46            <a class="nav-link" href="Supp_Log.jsp?role=contsupp">Supplier</a>
47        </li>
48        <li class="nav-item active">
49            <a class="nav-link" href="Login.jsp?role=ManufTesting">Testing</a>
50        </li>
51        <li class="nav-item active">
52            <a class="nav-link" href="ad_Log.jsp">Admin</a>
53        </li>
54        <!-- <li class="nav-item">
55            <a class="nav-link scroll" href="#partners">Partners</a>
56        </li>
```

Ad_log.jsp

```
ad_log.jsp
110 </style>
111 </head>
112<body>
113
114 <a href="index.html"><button style="width:80px;height: 35px; border:5px; margin-top:70px; margin-left:70px; color: black; font-size: 16px; background-color: transparent; border: none; outline: none;" type="button">Admin Log In</button>
115<div id="login-form">
116    <div id="Login-head">
117        <h1>Admin Login</h1>
118    </div>
119    <div id="Login-details">
120        <form action="Ad_Log" method="post">
121            <div id="user">
122                <input type="email" name="email" placeholder="email@gmail.com">
123            </div>
124
125            <div id="pass">
126                <input type="password" name="pss" placeholder="Password">
127            </div>
128
129            <div id="submit">
130                <input type="submit" value="Log-In">
131            </div>
132        </form>
133    </div>
134
135</div>
136</div>
137</body>
138</html>
139
140
```

Login.jsp

```
 Login.jsp
115<%
116 String role=request.getParameter("role");
117 %
118 <a href="index.html"><button style="width:80px;height: 35px; border:5px; margin-top:70px; margin-left:70px; color: black; font-size: 16px; background-color: transparent; border: none; outline: none;" type="button">Login</button>
119<div id="login-form">
120    <div id="Login-head">
121        <h1><%=role%> Login</h1>
122    </div>
123    <div id="Login-details">
124        <form action="LoginUser" method="post">
125            <div id="user">
126                <input type="email" name="email" placeholder="email@gmail.com">
127            </div>
128            <input type="hidden" name="role" value=<%=role%>">
129            <div id="pass">
130                <input type="password" name="pss" placeholder="Password">
131            </div>
132
133            <div id="submit">
134                <input type="submit" value="Log-In">
135            </div>
136        </form>
137    </div>
138
139</div>
140</div>
141</body>
142</html>
143
```

LogReg.jsp

```
1 LogReg.jsp
2 210<form id="survey-form" action="RegUser" method="post">
3 211<div class="form-group">
4 212<label id="email-label" for="email">Email</label>
5 213<input
6 214    type="email"
7 215    name="email"
8 216    id="email"
9 217    class="form-control"
10 218    placeholder="Enter your Email"
11 219    required
12 220  />
13 221</div>
14 222
15 223<div class="form-group">
16 224<p>Question for the security key?</p>
17 225<select id="dropdown" name="qun" class="form-control" required>
18 226<option disabled selected value>Select current role</option>
19 227<option value="What is the priority for manufacture">What is the priority for manufacture</option>
20 228<option value="what is the main purpose of testing">what is the main purpose of testing</option>
21 229<option value="how the customer satisfy with the product mostly">How the customer satisfy with the product mo
22 230<option value="which gender are like the product most">Which gender are like the product most</option>
23 231<option value="which product women like ">which product women like </option>
24 232</select>
25 233</div>
26 234
27 235<div class="form-group">
28 236<label id="email-label" for="email">Answer</label>
29 237<input
30 238    type="Text"
31 239    name="ans"
32 240    id="email"
33 241    class="form-control"
34 242    placeholder="Enter your Answer"
35 243    required
36 244
```

smtpKey.jsp

```
1 smtpKey.jsp
2 1<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
3 2    pageEncoding="ISO-8859-1"%>
4
5 3<%@ page import="java.sql.* ,java.util.* ,java.io.* "%>
6 4<%@ page import="java.io.* ,java.util.* ,javax.mail.* "%>
7 5<%@ page import="javax.mail.internet.* ,javax.activation.* "%>
8 6<%@ page import="javax.servlet.http.* ,javax.servlet.* "%>
9 7<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"
10 8<html>
11 9<head>
12 10<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
13 11<title>Insert title here</title>
14 12</head>
15 13<body>
16 14<%
17 15        Multipart multipart = new MimeMultipart();
18 16        MimeBodyPart textPart = new MimeBodyPart();
19 17        String id = request.getParameter("id");
20 18        String role=request.getParameter("role");
21 19        String email=request.getParameter("mail");
22 20        int max = 99999;
23 21        int min=100;
24 22        int ss = (int) Math.floor(Math.random() * (max - min + 1) + min);
25 23        String pass = "PS-"+ss+"";
26 24
27 25        try {
28 26            Class.forName("com.mysql.jdbc.Driver");
29 27            Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/cosmetic","root","root");
30 28
31 29            String sql="update login set ad_status='accepted',req_key='"+pass+"' where ad_status='requested'";
32 30            PreparedStatement ps=cn.prepareStatement(sql);
33 31            int status = ps.executeUpdate();
34 32
35 33
```

```

smtpKey.jsp
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/cosmetic","root","root");
    String sql="update login set ad_status='accepted',req_key='"+pass+"' where ad_status='requested' ";
    PreparedStatement ps=cn.prepareStatement(sql);
    int status = ps.executeUpdate();

    String nn="provided";
    textPart.setText("<html><head>" +
    "<style type=\"text/css\">" +
    "</style>" +
    "</head>" +
    "<p>"+Hi <b>"+email+"</b>,Your profile has been accepted successfully for <b></b>.Your pas" +
    "</p>" +
    "</html>",
    "ascii", "html");
    multipart.addBodyPart(textPart);
    String message1="Your Profile Accepted Successfully.Your password is "+pass+"";
    String host="", user="", passw="";
    host ="smtp.gmail.com";
    user ="vd1264833@gmail.com";
    passw ="hjmvshsfjcmrbny";
    String SSL_FACTORY = "javax.net.ssl.SSLSocketFactory";
    String to ="vd1264833@gmail.com";
    String from ="vd1264833@gmail.com";
    String subject="User Acceptance Information";
    String messageText = message1;
    boolean sessionDebug = true;

    Properties props = System.getProperties();
    props.put("mail.host",host);
}

```

```

smtpKey.jsp
try {
    transport.sendMessage(msg, msg.getAllRecipients());
}
catch (Exception err) {
    out.println("message not successfully sended");
}
transport.close();
String pstatus="Accepted";

if(status>0){
    %
    <script>
        alert("Acceptance details successfully sent to mail");
        window.location="viewLogReq.jsp";
    </script>
    %
}
else{
    %
    <script>
        alert("Acceptance details not successfully sent");
        window.location="viewLogReq.jsp";
    </script>
    %
}
}catch(Exception ex){
    ex.printStackTrace();
}
%
</body>
</html>

```

view_encData.jsp

```
view_encData.jsp
157 }
158 else
159 {
160     String qry = "SELECT * FROM manuf where ad_status='testcompleted' and cl_id='"+id+"' ";
161     PreparedStatement ps = cn.prepareStatement(qry);
162     ResultSet q1=ps.executeQuery();
163     System.out.println("elseeeeeeeeeeeeeeeeeeeee");
164     while (q1.next()) {
165
166
167
168     %>
169
170
171
172
173
174
175
176
177
178
179
180
181             <tr>
182             <td><%=q1.getString(1) %></td>
183             <td><%=q1.getString(2) %></td>
184             <td><%=q1.getString(3) %></td>
185             <td><%=q1.getString(4) %></td>
186             <td><%=q1.getString(5) %></td>
187             <td><%=q1.getString(6) %></td>
188             <td><%=q1.getString(7) %></td>
189             <td><%=q1.getString(8) %></td>
190             <td><%=q1.getString(9) %></td>
191             <td><%=q1.getString(10) %></td>
192             <td><%=q1.getString(11) %></td>
193         </tr>
194     <%
195
196
197     }
198
199
200
201 }
```

CHAPTER 6

DATABASE TABLE STRUCTURES

Fig. 6.1. blc_chain

cosmetic.login: 3 rows total (approximately)					» Next	Show all	Sorting	Columns (9/9)	Filter
id	role	email	ad_status	req_key	req_question	ans	viewReq	ad_view	
1	manuf	manuf@gmail.com	accepted	PS-69501	What is the priority for manufacturer	good quality product	accepted	accepted	
2	ManufTesting	test@gmail.com	accepted	PS-99707	what is the main purpose of testing	no allergens	accepted	accepted	
3	quality	analyser@gmail.com	accepted	PS-47506	which gender are like the product most	female	pending	pending	

Fig. 6.2. login

Fig. 6.3. encry_manuf

cosmetic.microbial_test: 24 rows total (approximately)					» Next		Show all		Sorting		Columns (10/10) F	
id	prod_name	Test_Name	Test_Method	start_range	critical_range	deviation	cl_id	test_result	tot_process			
1	Skin Moisturizer	Microbial Testing	Agar Plate	158	134	-24	1	fail	pending			
2	Lipstick	Microbial Testing	Plate Count	140	134	-6	1	fail	pending			
3	Shampoo	Microbial Testing	Agar Plate	159	160	1	1	pass	completed			
4	Sunscreen	Microbial Testing	Plate Count	153	150	-3	1	fail	pending			
5	Mascara	Microbial Testing	Agar Plate	133	138	5	1	pass	completed			
6	Foundation	Microbial Testing	Plate Count	132	130	-2	1	fail	pending			
7	Lip Balm	Microbial Testing	Agar Plate	134	148	14	1	pass	completed			
8	Hair Conditioner	Microbial Testing	Plate Count	151	135	-16	1	fail	pending			
9	Perfume	Microbial Testing	Agar Plate	147	154	7	1	pass	completed			
10	Nail Polish	Microbial Testing	Plate Count	124	137	13	1	pass	completed			
11	Face Mask	Microbial Testing	Agar Plate	159	145	-14	1	fail	pending			
12	Eye Shadow Palette	Microbial Testing	Plate Count	141	160	19	1	fail	pending			
13	Hand Cream	Microbial Testing	Agar Plate	128	128	0	1	fail	pending			
14	Body Scrub	Microbial Testing	Plate Count	143	123	-20	1	fail	pending			
15	Nail Serum	Microbial Testing	Agar Plate	164	136	-28	1	fail	pending			
16	Tinted Lip Balm	Microbial Testing	Plate Count	132	166	34	1	fail	pending			
17	Hair Serum	Microbial Testing	Agar Plate	166	153	-13	1	fail	pending			
18	Highlighter Palette	Microbial Testing	Plate Count	157	139	-18	1	fail	pending			
19	Hair Spray	Microbial Testing	Agar Plate	144	143	-1	1	fail	pending			
20	Face Serum	Microbial Testing	Plate Count	152	139	-13	1	fail	pending			

Fig. 6.4. microbial_test

cosmetic.supplprod_finalres: 24 rows total (approximately)								
Date_of_Inspection	No_visible_cracks	Lid/Cap	Label_adheres	No_visible_dirt	Inside_is_clean	No_foreign_part	Label_is_leg	Print_qual_good
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	NO
pending	Yes	Yes	Yes	NO	Yes	Yes	NO	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	NO	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	NO	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	NO	Yes	Yes	Yes	Yes
pending	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

cosmetic.supplprod_finalres: 24 rows total (approximately)								
Container_type	Quantity	containers_alloys	Batch_Number	Date_of_Inspection	No_visible_cracks	Lid/Cap	Label_adheres	No_visible_dirt
1 Skin Moisturizer	400	GlassJar	bat-11830	pending	Yes	Yes	Yes	Yes
2 Lipstick	400	PlasticBottle	bat-11830	pending	Yes	Yes	Yes	Yes
3 Shampoo	400	Tube	bat-11830	pending	Yes	Yes	Yes	Yes
4 Sunscreen	400	GlassBottle	bat-11830	pending	Yes	Yes	Yes	NO
5 Mascara	400	PumpBottle	bat-11830	pending	Yes	Yes	Yes	Yes
6 Foundation	400	CompactCase	bat-11830	pending	Yes	Yes	Yes	Yes
7 Lip Balm	400	Tub	bat-11830	pending	Yes	Yes	Yes	Yes
8 Hair Conditioner	400	PumpDispenser	bat-11830	pending	Yes	Yes	Yes	Yes
9 Perfume	400	Tube	bat-11830	pending	Yes	Yes	Yes	Yes
10 Nail Polish	400	PlasticBottle	bat-11830	pending	Yes	Yes	Yes	Yes
11 Face Mask	400	Tube	bat-11830	pending	Yes	NO	Yes	Yes
12 Eye Shadow Palette	400	GlassBottle	bat-11830	pending	Yes	Yes	Yes	Yes
13 Hand Cream	400	SqueezeBottle	bat-11830	pending	Yes	Yes	Yes	Yes
14 Body Scrub	400	PlasticBottle	bat-11830	pending	Yes	Yes	Yes	Yes
15 Nail Serum	400	PumpDispenser	bat-11830	pending	Yes	Yes	Yes	Yes
16 Tinted Lip Balm	400	CompactCase	bat-11830	pending	Yes	Yes	Yes	Yes
17 Hair Serum	400	PlasticBottle	bat-11830	pending	Yes	Yes	Yes	Yes
18 Highlighter Palette	400	GlassDropperBottle	bat-11830	pending	Yes	Yes	Yes	NO
19 Hair Spray	400	Roll-OnBottle	bat-11830	pending	Yes	Yes	Yes	Yes

Fig. 6.5. supplprod_finalres

cosmetic.supp_log: 9 rows total (approximately)				
id	company	companycode	email	password
1	small	cp_1014	godin88@gmail.com	din1541
2	small	cp_1015	leeach@gmail.com	din1542
3	small	cp_1016	fiya@gmail.com	din1543
4	mid	cp_1017	limsa@gmail.com	din1544
5	mid	cp_1018	zogo@gmail.com	din1545
6	mid	cp_1019	mante@gmail.com	din1546
7	high	cp_1020	vigel@gmail.com	din1547
8	high	cp_1021	mosca@gmail.com	din1548
9	high	cp_1022	nanami@gmail.com	din1549

Fig. 6.6. supp_log

cosmetic.ssupplier_cont: 2 rows total (approximately)											
	Supplier Code	tot_qun	min_prc	max_prc	comp	grac_pen	termination_cause	reg	batch_id	status	ad_stDate
1	cp_1017	600	97,367	400,000	cp_1017	on	on	2024-03-06	bat-11830	assigned	2024-03-19
2	cp_1019	600	97,367	35,000	cp_1019	on	on	2024-03-05	bat-11830	closed	2024-03-19

Fig. 6.7. ssupplier_cont

cosmetic.proto_supplierres: 1 rows total (approximately)							
	No_visiblecracks	Cap_seals_properly	Labeladhereswithoutwrinkles	No_visible_dirt_or_residue	Inside_is_clean_and_dry	No_foreign_particles	Label_is_le
1	passed	passed	passed	passed	passed	passed	pending

Fig. 6.8. supplierres

cosmetic.testing_res: 24 rows total (approximately)					
	test_name	test_method	deviation	prod_name	
1	Microbial Testing	Agar Plate	15	Skin Moisturizer	
2	Microbial Testing	Plate Count	15	Lipstick	
3	Microbial Testing	Agar Plate	32	Shampoo	
4	Microbial Testing	Plate Count	25	Sunscreen	
5	Microbial Testing	Agar Plate	28	Mascara	
6	Microbial Testing	Plate Count	15	Foundation	
7	Microbial Testing	Agar Plate	25	Lip Balm	
8	Microbial Testing	Plate Count	30	Hair Conditioner	
9	Microbial Testing	Agar Plate	11	Perfume	
10	Microbial Testing	Plate Count	15	Nail Polish	
11	Microbial Testing	Agar Plate	12	Face Mask	
12	Microbial Testing	Plate Count	15	Eye Shadow Palette	
13	Microbial Testing	Agar Plate	26	Hand Cream	
14	Microbial Testing	Plate Count	18	Body Scrub	
15	Microbial Testing	Agar Plate	14	Nail Serum	
16	Microbial Testing	Plate Count	11	Tinted Lip Balm	
17	Microbial Testing	Agar Plate	12	Hair Serum	
18	Microbial Testing	Plate Count	26	Highlighter Palette	
19	Microbial Testing	Agar Plate	16	Hair Spray	
20	Microbial Testing	Plate Count	16	Face Serum	

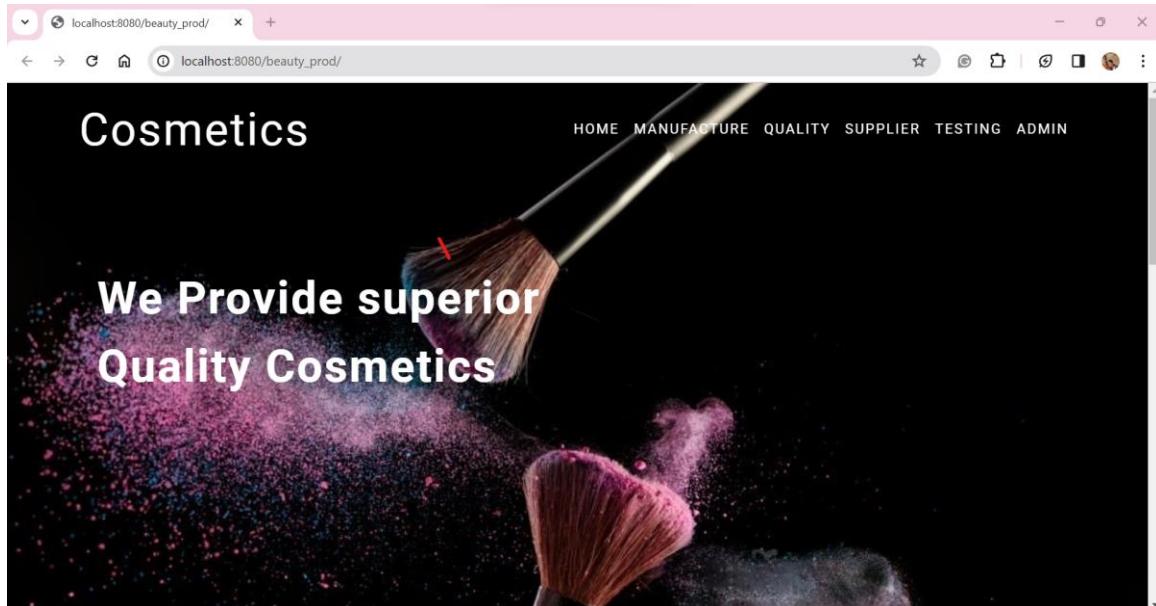
Fig. 6.9. testing_res

cosmetic.cont_qualityinsp: 24 rows total (approximately)										» Next	Show all	Sorting	Columns (15/15)	Filter
id	Container_type	Quantity	containers_alloys	Batch_Number	No_visible_cracks	Lid/Cap	Label_adheres	No_visible_dirt	Inside_is_clean					
1	Skin Moisturizer	400	GlassJar	bat-11830	Yes	Yes	Yes	Yes	Yes					
2	Lipstick	400	PlasticBottle	bat-11830	Yes	Yes	Yes	Yes	Yes					
3	Shampoo	400	Tube	bat-11830	Yes	Yes	Yes	Yes	Yes					
4	Sunscreen	400	GlassBottle	bat-11830	Yes	Yes	Yes	Yes	Yes					
5	Mascara	400	PumpBottle	bat-11830	Yes	Yes	No	Yes	No					
6	Foundation	400	CompactCase	bat-11830	Yes	Yes	Yes	Yes	Yes					
7	Lip Balm	400	Tub	bat-11830	Yes	Yes	Yes	Yes	Yes					
8	Hair Conditioner	400	PumpDispenser	bat-11830	Yes	Yes	Yes	Yes	Yes					
9	Perfume	400	Tube	bat-11830	Yes	No	Yes	Yes	Yes					
10	Nail Polish	400	PlasticBottle	bat-11830	Yes	Yes	Yes	Yes	Yes					
11	Face Mask	400	Tube	bat-11830	Yes	Yes	Yes	No	Yes					
12	Eye Shadow Palette	400	GlassBottle	bat-11830	Yes	No	Yes	Yes	No					
13	Hand Cream	400	SqueezeBottle	bat-11830	Yes	Yes	Yes	Yes	Yes					
14	Body Scrub	400	PlasticBottle	bat-11830	Yes	Yes	Yes	No	Yes					
15	Nail Serum	400	PumpDispenser	bat-11830	Yes	No	Yes	Yes	Yes					
16	Tinted Lip Balm	400	CompactCase	bat-11830	Yes	Yes	Yes	Yes	Yes					
17	Hair Serum	400	PlasticBottle	bat-11830	Yes	Yes	No	Yes	Yes					
18	Highlighter Palette	400	GlassDropperBottle	bat-11830	Yes	Yes	Yes	Yes	Yes					
19	Hair Spray	400	Roll-OnBottle	bat-11830	Yes	No	Yes	Yes	No					

Fig. 6.10. cont_qualityinsp

CHAPTER 7

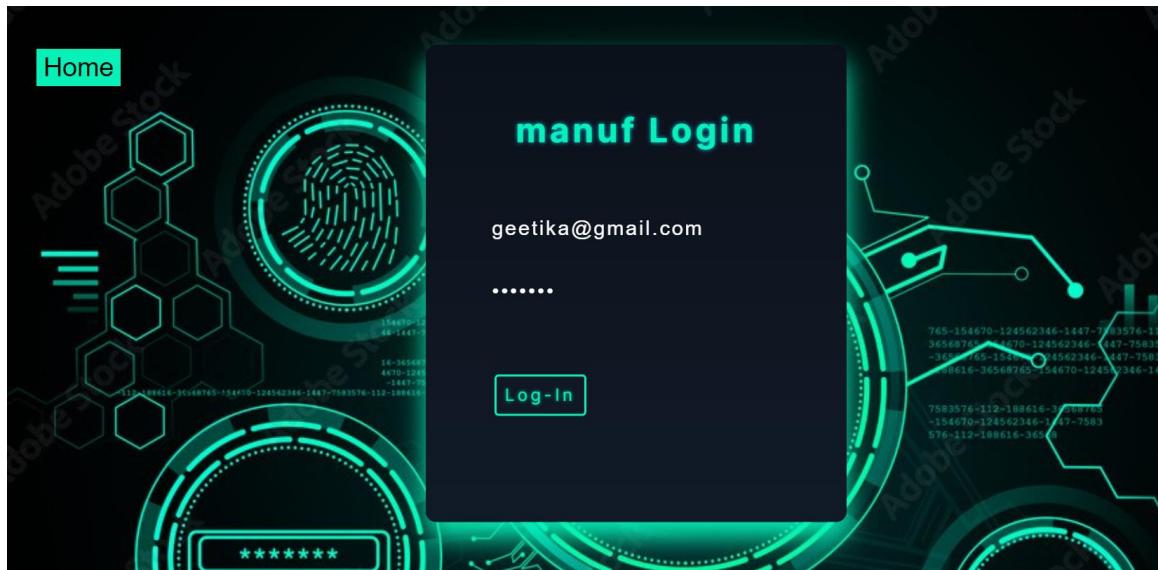
OUTPUT SNAPSHOTS



The screenshot shows a section of a website with four circular icons and their corresponding headings and descriptions:

- Consumer Trends**: Consumer preferences play a significant role in shaping the industry. Recent years have seen a shift towards natural and organic products, cruelty-free formulations, and inclusivity in shade ranges to cater to diverse skin tones.
- Innovation**: Continuous innovation in product development, packaging, and marketing is crucial to staying competitive. Companies invest heavily in research and development to create new formulations, technologies, and delivery systems.
- E-commerce**: The rise of e-commerce has transformed the way cosmetics are bought and sold. Online platforms and social media have become essential marketing and sales channels, enabling direct-to-consumer brands to thrive.
- Sustainability**: Increasing awareness of environmental concerns has led to a greater emphasis on sustainability in the industry. Brands are working to reduce packaging waste, carbon footprints, and the use of harmful chemicals.

At the bottom, there is a copyright notice: © 2018 Timber. All rights reserved | Design by W3layouts.



Insert title here

localhost:8080/beauty_prod/LoginUser

Register Sign-Up form

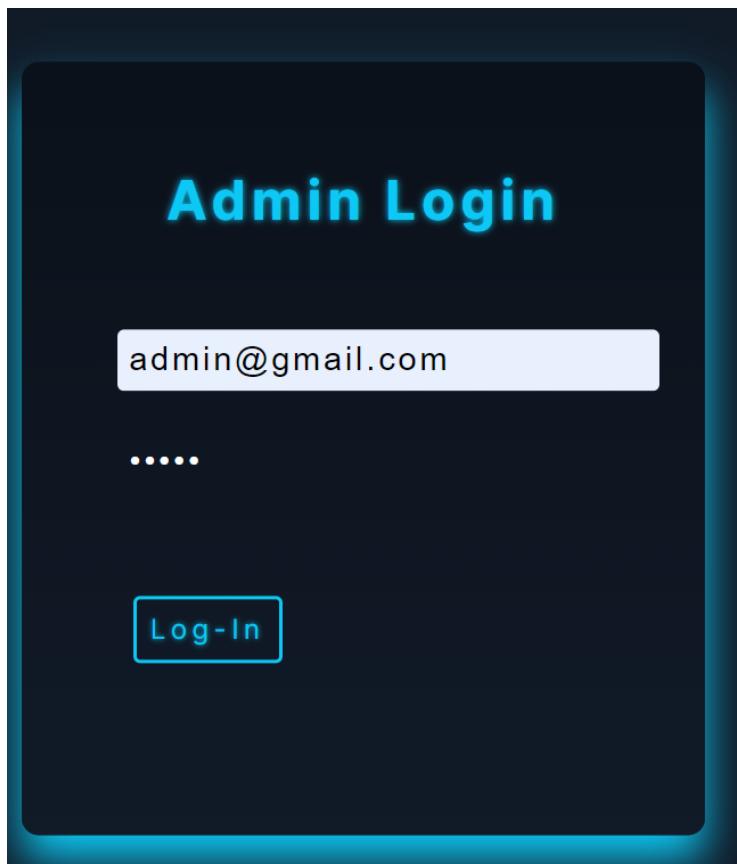
Email
geetika@gmail.com

Question for the security key?
What is the priority for manufacture

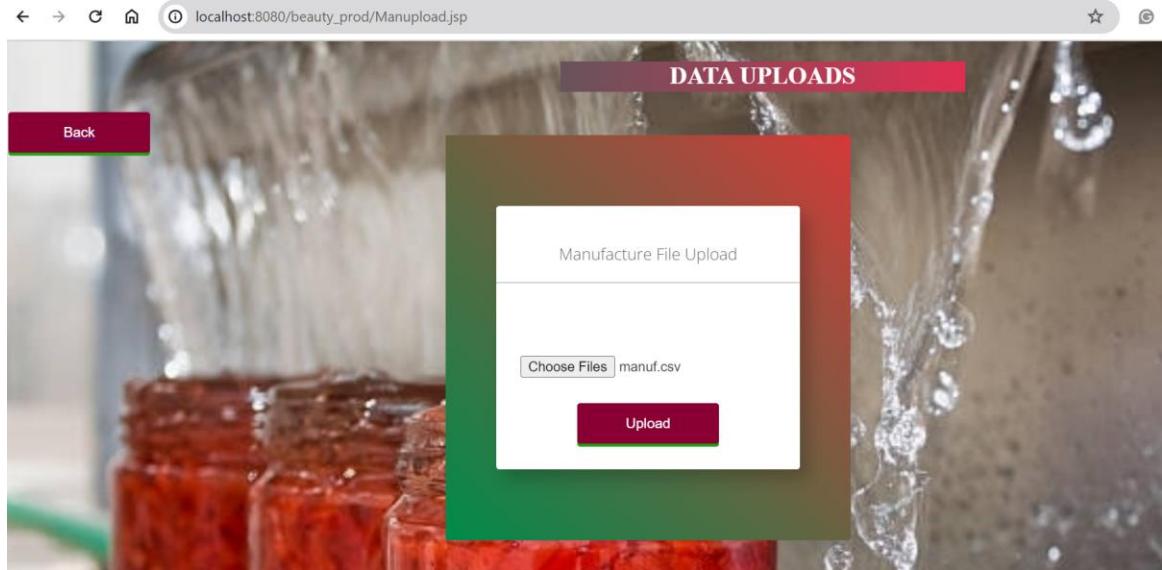
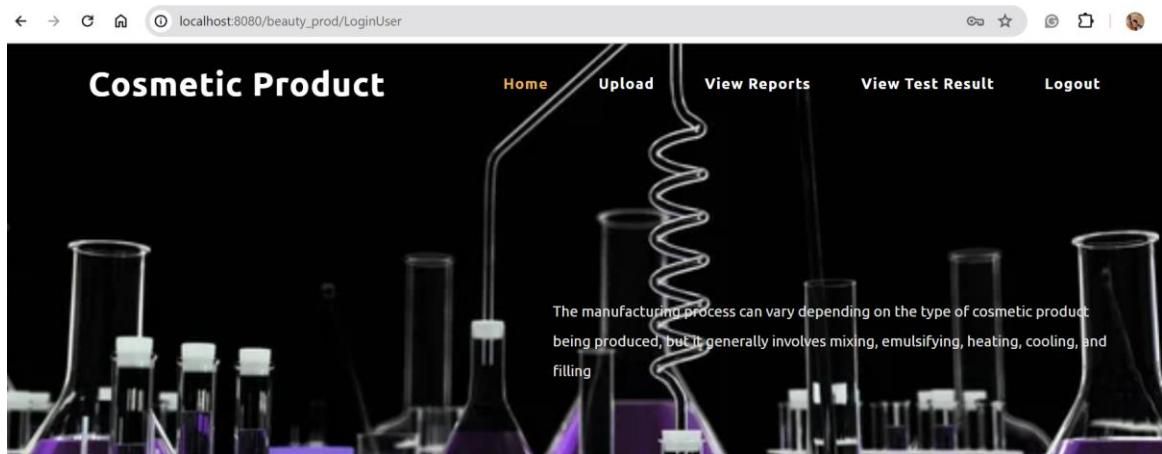
Answer
Hitting the niche market with product look.

submit

ID	ROLE	EMAIL	ACTION
2	manuf	geetika@gmail.com	REQUEST



Id	Quantity	Status	Action
2	manuf	geetika@gmail.com	Accept

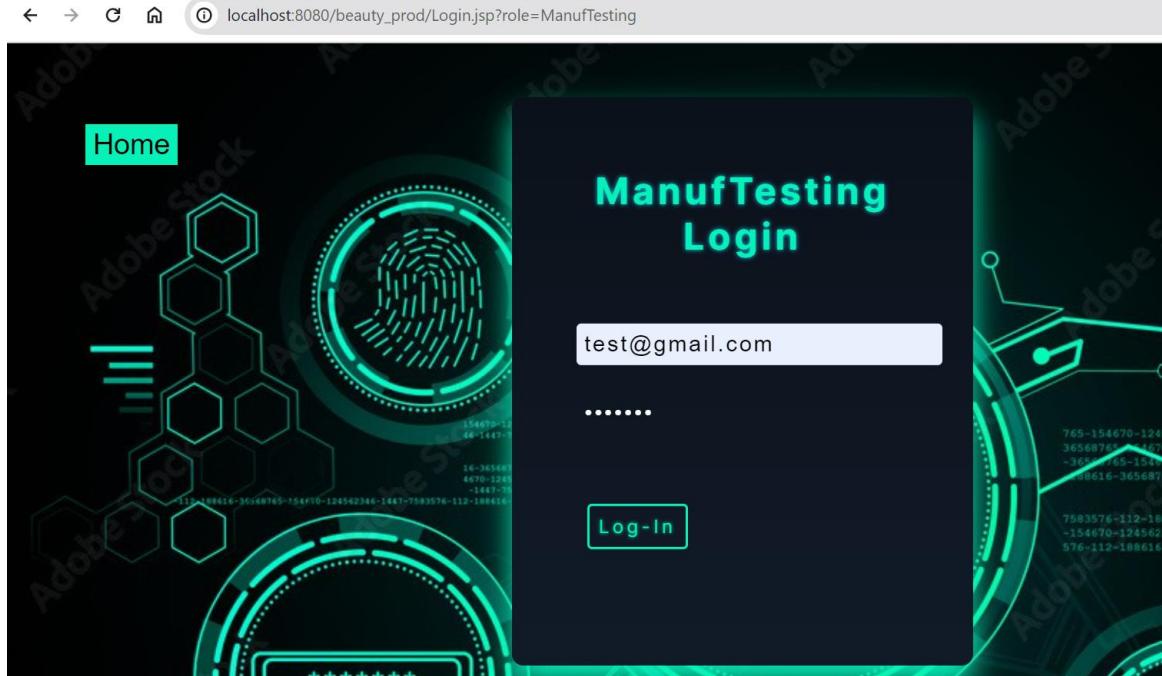


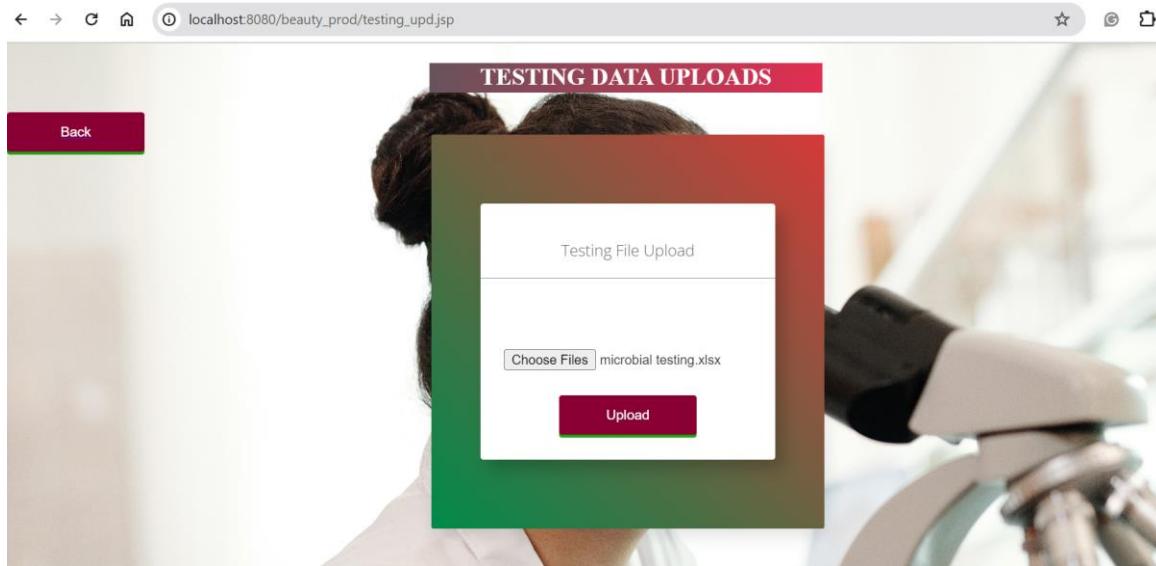
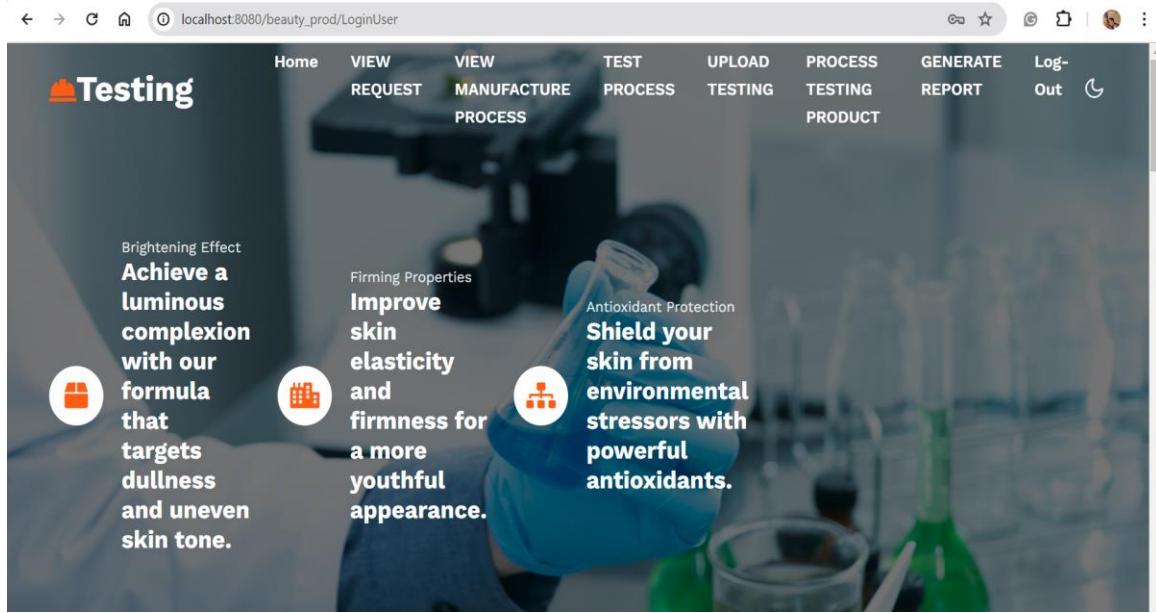
localhost:8080/beauty_prod/viewtesmanufprod.jsp?id=1

Process					
Id	Product_Total	status		action	
1	24	mfcompleted		<button>submit</button>	

localhost:8080/beauty_prod/sndtes_procss.jsp?cl_id=1

ID	Product Name	Raw Material	Quantity	Batch Cost	Unit Price	Cost estim	batch	test_result	quality_result	status
1	Skin Moisturizer	Glycerin,Water	100	19260	1	343	bat-20088	pending	pending	mfcompleted
2	Lipstick	Mica,OliveOil	100	16776	1	354	bat-20088	pending	pending	mfcompleted
3	Shampoo	Sulfates,CoconutOil	100	48275	1	265	bat-20088	pending	pending	mfcompleted
4	Sunscreen	Avobenzone,ZincOxide	100	16424	1	205	bat-20088	pending	pending	mfcompleted
5	Mascara	CarbonBlackPowder,Water	100	37332	1	389	bat-20088	pending	pending	mfcompleted
6	Foundation	Parabens,AloeVeraGel	100	41258	1	451	bat-20088	pending	pending	mfcompleted
7	Lip Balm	Petrolatum,Beeswax	100	48824	1	171	bat-20088	pending	pending	mfcompleted
8	Hair Conditioner	Silicone,ArganOil	100	27004	1	105	bat-20088	pending	pending	mfcompleted
9	Perfume	SyntheticFragrances,EssentialOils	100	16042	1	175	bat-20088	pending	pending	mfcompleted
10	Nail Polish	Acetone,NailLacquer	100	26965	1	117	bat-20088	pending	pending	mfcompleted
11	Face Mask	Preservatives,Clay	100	19437	1	419	bat-20088	pending	pending	mfcompleted
12	Eye Shadow Palette	Talc,PigmentedPowders	100	14107	1	374	bat-20088	pending	pending	mfcompleted
13	Hand Cream	ParaffinWax,SheaButter	100	13957	1	212	bat-20088	pending	pending	mfcompleted
14	Body Scrub	Microplastics,Sugar,CoconutOil	100	14786	1	451	bat-20088	pending	pending	mfcompleted
15	Nail Serum	Formaldehyde,VitaminE,Biotin	100	13291	1	385	bat-20088	pending	pending	mfcompleted





VIEW MANUFACTURE REQUEST					
Id	prehash	nexthash	count	date	
1	00000000000000000000000000000000	8b5a639c4e0c0a3d2ca79313bf8dd0220cbc53a9e6982db6bcff66789927f1ec	44	15-04-2024	

VIEW MANUFACTURE REQUEST							
nexthash	count	date	data	User	status	Question	Key
bf8dd0220cbc53a9e6982db6bcf66789927f1ec	44	15-04-2024	[B@4e6f6fd5	manuf	pending	what is the main purpose of testing	<button>view</button>



Manufacture Requests								
Id	role	email	ad_status	req_key	req_question	viewReq	Action	View
1	manuf	manuf@gmail.com	accepted	PS-15427	What is the priority for manufacture	good quality product	<button>Process</button>	<button>View</button>
2	ManuTesting	test@gmail.com	accepted	PS-11423	what is the main purpose of testing	no allergens	<button>Process</button>	<button>View</button>

VIEW MANUFACTURE REQUEST

hash	count	status	Question	Key
?0cbc53a9e6982db6bcf66789927f1ec	44	requested	what is the main purpose of testing	<button>Download</button>

CLIENT VERIFICATION

what is the main purpose of testing |

SUBMIT

VIEW MANUFACTURE REQUEST

count	date	data	User	status	Question	Key
6789927f1ec	44	15-04-2024 [B@522c44a2]	manuf	accepted	what is the main purpose of testing	<input type="button" value="Choose File"/> ManufTesting-779166.png <input type="button" value="UPLOAD"/>



VIEW PRODUCT FOR QR GENERATOR

Process

ID	Product Name	Raw Material	Quantity	Batch Cost	Unit Price	Cost estim	batch	test_result	quality_result	status
1	Skin Moisturizer	Glycerin,Water	100	19260	1	343	bat-20088	pending	pending	testrequested
2	Lipstick	Mica,OliveOil	100	16776	1	354	bat-20088	pending	pending	testrequested
3	Shampoo	Sulfates,CoconutOil	100	48275	1	265	bat-20088	pending	pending	testrequested
4	Sunscreen	Avobenzone,ZincOxide	100	16424	1	205	bat-20088	pending	pending	testrequested
5	Mascara	CarbonBlackPowder,Water	100	37332	1	389	bat-20088	pending	pending	testrequested
6	Foundation	Parabens,AloeVeraGel	100	41258	1	451	bat-20088	pending	pending	testrequested
7	Lip Balm	Petrolatum,Beeswax	100	48824	1	171	bat-20088	pending	pending	testrequested
8	Hair Conditioner	Silicone,ArganOil	100	27004	1	105	bat-20088	pending	pending	testrequested
9	Perfume	SyntheticFragrances,EssentialOils	100	16042	1	175	bat-20088	pending	pending	testrequested
10	Nail Polish	Acetone,NailLacquer	100	26965	1	117	bat-20088	pending	pending	testrequested
11	Face Mask	Preservatives,Clay	100	19437	1	419	bat-20088	pending	pending	testrequested
12	Eye Shadow Palette	Talc,PigmentedPowders	100	14107	1	374	bat-20088	pending	pending	testrequested
13	Hand Cream	ParaffinWax,SheaButter	100	13657	1	212	bat-20088	pending	pending	testrequested

View Testing Product

ID	PRODUCT_NAME	RAW MATERIAL	QUANTITY	BATCH COST	UNIT_PRICE	COST_ESTIM	BATCH
1	Skin Moisturizer	Glycerin,Water	100	19260	1	343	bat-20088
2	Lipstick	Mica,OliveOil	100	16776	1	354	bat-20088
3	Shampoo	Sulfates,CoconutOil	100	48275	1	265	bat-20088

[BACK](#)

VIEW RE MANUFACTURE TEST

Id	Total_Product	Test_Name	Process	Action
1	24	pending	pending	Process

[BACK](#)

VIEW TEST RESULT

Id	Total_Product	Status	Action
1	24	Microbial Testing	View

localhost:8080/beauty_prod/view_encData.jsp

View Testing Product

ID	PRODUCT_NAME	RAW MATERIAL	QUANTITY	BATCH COST	UNIT_PRICE
1	'h á/a?Y';éöcl?fÉéúS-?ójý! áimí5	s□□□Q□?ÜAr□e[ç	5Å□1Ú~I□/□"? <øwL	~ééC^Nvh\$üca.~o?	Bø? Ažæ.úOç? ±ÍYD
2	½'Ø¶?{qe?q i'e?ÆÉéúS-? ðý!áimí5	ú«íl^tHÖ~9Fa??yú	5Å□1Ú~I□/□"? <øwL	□[9AExñ□!1åØ□?^k]	Bø? Ažæ.úOç? ±ÍYD
3	:s}º=ü□Å £R??Ø/□ÉéúS-? ðý!áimí5	VNG□□jat□y?@?□?~¾#□Ö□m;? ¥?4æ	5Å□1Ú~I□/□"? <øwL	Ø'øLD□M?9?E&a□ña	Bø? Ažæ.úOç? ±ÍYD

CHAPTER 8

CONCLUSION AND FUTURE PLANS

The proposed system marks a transformative shift in cosmetic manufacturing, addressing the limitations of the existing system. By leveraging blockchain technology with robust encryption standards, introducing allergen testing, and ensuring supply chain transparency, it sets new industry standards for transparency, safety, and ethical practices. This innovative approach instills trust and confidence in consumers also positions the cosmetic industry to meet the evolving demands of authenticity and safety. Additionally, advanced microbial testing methods are on the horizon for even more accurate allergen detection. These forward-looking enhancements will elevate cosmetic manufacturing to a new level, ensuring it remains a paragon of responsible, secure, and trustworthy practices in the cosmetic industry.

CHAPTER 9

REFERENCES

1. <https://dl.acm.org/doi/abs/10.1145/3447879.3447883>
2. <https://ieeexplore.ieee.org/document/8661522>
3. <https://ieeexplore.ieee.org/document/9031596>
4. <https://squ.elsevierpure.com/en/publications/i-nice-a-new-approach-for-identifying-the-number-of-clusters-and->
5. <https://research.aber.ac.uk/en/publications/a-link-based-approach-to-the-cluster-ensemble-problem>