WEB APPLICATION
FOR
PRODUCT RETURN MANAGEMENT

A Project Report

Submitted by

Mr. Arikeri Praveen	**20201CAI0022**
Mr. B Mohan Reddy	**20201CAI0051**
Mr. Katta Uday Kiran Reddy	**20201CAI0076**
Mr. Addagalla Koushikeswar	**20201CAI0218**

Under the guidance of

Dr. Mohammadi Akheela Khanum
Professor

in partial fulfillment for the award of the degree

of
BACHELOR OF TECHNOLOGY

IN
COMPUTER SCIENCE AND COMMUNICATION ENGINEERING
At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

PRESIDENCY UNIVERSITYBENGALURU

JUNE 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project report “**WEB APPLICATION FOR PRODUCT RETURN MANAGEMENT**” being submitted by “Mr. Arikeri Praveen, Mr. B Mohan Reddy, Mr. Katta Uday Kiran Reddy, Mr. Addagalla Koushikeswar” bearing roll number(s) “20201CAI0022, 20201CAI0051, 20201CAI0076, 20201CAI0218” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence & Machine Learning) is a bonafide work carried out under my supervision.

Dr. Mohammadi Akheela Khanum
PROFESOR
School of CSE&IS
Presidency University

Dr. Zafar Ali Khan
Prof. & HOD-ECE&CCE
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. MD. SAMEERUDDIN KHAN Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **WEB APPLICATION FOR PRODUCT RETURN MANAGEMENT** in partial fulfilment for the award of Degree of **Bachelor of Computer Science and Engineering(Artificial Intelligence and Machine Learning)**, is a record of our own investigations carried under the guidance **Dr.Mohammadi Akheela Khanum Professor, School of Computer Science & Engineering, Presidency University, Bengaluru**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll.no	Signature
Mr. Arikeri Praveen	20201CAI0022	
Mr. B Mohan Reddy	20201CAI0051	
Mr. Katta Uday Kiran Reddy	20201CAI0076	
Mr. Addagalla Koushikeswar	20201CAI0218	

ABSTRACT

Aim of the project is to address the challenge faced by SKF's customer service team, where products are routinely returned to the regional warehouse, regardless of their condition, leading to damaged stocks and subsequent losses at the India Distribution Centre. To mitigate this issue, the authors propose the development of a comprehensive web application that streamlines the return management process.

This application uses HTML and CSS to create user-centric interfaces, including home, user login, employee login, and return policy pages. JavaScript improves the user experience with interactive pop-up messages. The user data accessed through the Returns Management page is stored in a MySQL database called PROJECT. Three required tables were created in this database: USER for user login data, DATA for product status information, and EMPLOYEE for employee login data.

The homepage has specific login buttons for users and employees, each pointing to their respective portals. When a user logs in, individuals are redirected to the return page, where information about the status of the product can be submitted. Then, the transmitted data is stored in the DATA table, including attributes such as the item ID, device status, response text, and return reason.

Employee login functionality allows employees to view user profiles through tables displayed on their portals. Within these tables, employees can select the appropriate warehouse based on the condition of the item—either intact/undamaged or undamaged.

A Java Full Stack approach to servlet programming was adopted to seamlessly integrate with HTML and MySQL. It used a servlet API and MySQL Connector jar files, which were added to the project dependencies using the Eclipse for Enterprise IDE.

An application with dynamic web pages created and driven by Java servlets establishes an efficient way to collect and process user data. The decision system then analyzes the data and determines whether the product should be sent to the normal/undamaged warehouse or the damaged warehouse.

This comprehensive web application aims to improve decision-making, reduce waste, improve inventory management, and reduce losses for SKF. The solution not only addresses immediate challenges but establishes a scalable and scalable framework for the continuous improvement of the recovery process.

ACKNOWLEDGEMENT

First, we are indebted to the Lord for allowing us to excel in our efforts to complete this project on time.

We would like to express our sincere gratitude to our College Presidency University for letting us fulfill my dream of being a student here.

We express our sincere thanks to our respected Dr. Md Sameeruddin Khan, Dean, School of Computer Science and Engineering, Presidency University for granting us the permission to undergo the project.

We would to convey our gratitude to our beloved Associate Dean Dr.C. Kalaiarasan, Professor Dr. T.K Thivakaran, University Project-II In-Charge, School of Computing Science & Engineering, Presidency University,
We would like to acknowledge the support and guidance of our project supervisor, Dr.Mohammadi Akheela Khanum Presidency University who provided invaluable insights and direction throughout this project duration.
We also extend our thanks to the academic community, whose research and publications provided the foundation of our project. Special thanks go to the authors of the various papers and articles that we referenced in our work.

Finally, also extending our heartiest thanks to our parents, friends, and well-wishers for being with us and extending encouragement throughout the project.

Mr. Arikeri Praveen– 20201CAI0022
Mr. B Mohan Reddy– 20201CAI0051
Mr. Katta Uday Kiran Reddy– 20201CAI0076
Mr. Addagalla Koushikeswar– 20201CAI0218

LIST OF CHAPTERS

LIST OF FIGURES

CHAPTER -1

INTRODUCTION

In the dynamic realm of supply chain management, the efficient handling of product returns poses a critical challenge for companies seeking to optimize their operations. SKF, a prominent provider of bearings, has identified a substantial issue within its customer service framework. The current practice of returning products to regional warehouses without considering their actual condition has led to a significant problem at the India Distribution Centre. Damaged stocks persist, even when accompanied by a seemingly satisfactory Proof of Delivery, resulting in blocked inventory and consequential financial losses.

To counteract this issue, a proactive solution is proposed—one that involves the development of a web application geared towards precise evaluation and justification of product returns. The core objectives are to establish a systematic decision-making process ensuring the accurate disposition of returned goods and, consequently, the reduction of blocked stocks and mitigation of financial losses.

This web application utilizes a multi-tiered technological approach, employing HTML and CSS for a user-friendly interface, JavaScript for interactive elements, and Java Full Stack with servlet programming for robust backend functionality. The integration of MySQL facilitates the storage and retrieval of essential data, organized into tables within the "PROJECT" database. Three key tables—USER, DATA, and EMPLOYEE—serve to manage user login details, product condition information, and employee credentials, respectively.

The user journey begins with a home page featuring distinct login options for users and employees. Upon successful login, users are directed to a return management page, where they provide detailed responses regarding the condition of the returned product. Collected data is then stored in the MySQL DATA table, featuring attributes such as product ID, condition, feedback, and reason for return.

Employees, accessing the system through dedicated login portals, can view user responses through dynamic tables. Within these tables, a crucial decision is made regarding the warehouse destination—normal/undamaged or damaged—based on the product's condition.

The technological backbone of this solution leverages Java servlets and the Eclipse for Enterprise IDE, ensuring dynamic web page creation, seamless connectivity between HTML and MySQL, and efficient handling of user requests and responses.

By implementing this web application, SKF aims to revolutionize its return management process, introduce transparency in decision-making, and significantly minimize the accumulation of blocked stocks and losses associated with damaged goods. The subsequent sections will delve into the methodology, outcomes, and potential benefits of this innovative solution.

CHAPTER -2
LITERATURE REVIEW

Sl. No.	Paper Title	Method	Advantages	Limitations
1	Optimizing Product Return Management Systems in Supply Chains	1. Analyzing return management strategies 2. Automated systems. 3.Decsion Algorithms 4. User interface designs.	1. Highlighting improved efficiency. 2. Reduced operational costs. 3. Enhanced customer satisfaction.	1.Data quality 2.Limited scope 3.Limited time horizon
2	Customer-Centric Approaches to Product Return Systems.	1.Examining methods to gather customer feedback on return processes 2. Corporating their preferences into return management systems.	1. Enhanced customer loyalty. 2. Increased trust. 3. Higher retention rates.	1. Challenges related to diverse customer expectations. 2.Potential complexities in system integration. 3.Adapting to changing customer needs.
3	Technological Solutions for Return Management.	1.Comparing technological solutions like RFID, IoT-enabled tracking, and web-based applications for return management.	1.Improved visibility. 2.Real-time tracking. 3.Data-driven decision-making.	1. High initial setup costs. 2.Potential security concerns. 3.System compatibility issues.
4	Decision-Making Algorithms in Return Management Systems.	1.Evaluating the effectiveness of decision-making algorithms in routing returned products to appropriate warehouses based on condition assessments.	1. Streamlined processes. 2.Reduced wastage. 3.Minimized stock blockages.	1. Algorithm complexity. 2.Accuracy issues with certain product categories. 3.Adaptability to changing conditions
5	User Interface Design Impact on Return Management Systems.	1.Investigating the influence of user interface design on user engagement and accuracy in providing product condition information.	1. Improved data accuracy 2.I Improved data accuracy. 3.Reduced processing times.	1. Design complexity. 2. Potential usability issues. 3. Training requirements for users.

2.1. LITERATURE REVIEW SUMMARY

The current review delves into six pivotal research papers addressing various facets of return management systems within supply chains. Each study explores distinct methodologies, advantages, and disadvantages, contributing valuable insights to the optimization of return processes.

1. "Optimizing Product Return Management Systems in Supply Chains"
   - Summary: This study analyzes diverse return management strategies, emphasizing automated systems, decision algorithms, and user interface designs. The advantages include enhanced efficiency, reduced costs, and improved customer satisfaction. Disadvantages involve initial implementation costs and potential resistance to technological shifts.

2. "Customer-Centric Approaches to Product Return Systems"
   - Summary: Focused on customer preferences, this research explores methods to incorporate user feedback into return management systems. Advantages include heightened customer loyalty and trust, while potential disadvantages encompass managing diverse customer expectations and adapting systems to evolving needs.

3. "Technological Solutions for Return Management: A Comparative Analysis"
   - Summary: This paper compares technological solutions, including RFID, IoT tracking, and web applications. Advantages involve improved visibility, real-time tracking, and data-driven decision-making. Disadvantages encompass high initial costs, security concerns, and compatibility issues.

4. "Decision-Making Algorithms in Return Management Systems"
   - Summary: Evaluating decision-making algorithms, this research focuses on routing returned products based on condition assessments. Advantages encompass streamlined processes and minimize wastage, while challenges include algorithm complexity and accuracy concerns.

5. "User Interface Design Impact on Return Management Systems"
   - Summary: Investigating the influence of UI design on user engagement, this study highlights advantages such as improved data accuracy and reduced processing times. However, challenges include design complexity and potential usability issues.

Collectively, these studies offer a comprehensive overview of methodologies, advantages, and disadvantages in return management systems. Integrating these insights can inform the development of a holistic approach to address the challenges faced by companies like SKF, ensuring efficient, customer-centric, and technologically advanced return processes in supply chains.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODOLOGY

1. "Optimizing Product Return Management Systems in Supply Chains".

- Research Gap: Limited research on the long-term effects and flexibility of automated returns management systems in dynamically changing supply chain conditions.

2. "Customer-Centered approaches to product return programs".

- Research Gaps: Insufficient research is done on the cultural and demographic influences on customer preferences across industries, leading to gaps in understanding how to tailor returns to different customer needs.

3. "Technological Solutions for Restoration Management: A Comparative Analysis".

- Research gap: Lack of comprehensive studies assessing the environmental sustainability aspects of different technological solutions in return management systems, including their carbon footprint and overall ecological impact.

4. "Decision-Making Algorithms in Restoration Management Systems".

- Research gap: limited research on flexible decision-making processes, especially how effective they are at managing profitability at scale during periods of maturity or volatile market conditions.

5. "Design Impact of User Interaction on Return Scheduling".

- Research gap: Inadequate investigation into the accessibility and inclusivity aspects of user interface designs, with a need to explore how diverse user demographics, including those with disabilities, interact with and benefit from these designs.

Identifying and addressing these research gaps can significantly contribute to refining and improving the regression management process, ensuring its effectiveness, sustainability, and deployment in different contexts.

CHAPTER-4
PROPOSED METHODOLOGY

To address the challenges faced by SKF's customer service team regarding the indiscriminate return of products, we propose a comprehensive web-based Return Management System. The solution involves the integration of HTML, CSS, JavaScript, MySQL, and Java Full Stack with servlet programming.

4.1. User and Employee Authentication:

- Implementation: HTML and CSS for login pages, JavaScript for user-friendly pop-up messages.
- Method: Users and employees are authenticated through secure login pages. Credentials are stored in the MYSQL database in respective tables.

4.2. Data Storage: MySQL Database

- Implementation: MYSQL database named "PROJECT" with three tables: USER, DATA, and EMPLOYEE.
- Tables:
  - USER Table: Stores user login data (username, password).
  - DATA Table: Captures product condition information (product ID, condition, feedback, reason for return).
  - EMPLOYEE Table: Manages employee login details.

4.3. User Interface Design

- Implementation: HTML and CSS for home page, user login, employee login, and return management page.
- Method:
  - Home Page: Contains login buttons for users and employees.
  - User Login: Redirects users to the return management page upon successful login.
  - Return Management Page: Displays questions related to the product condition.
  - Employee Login: Redirects employees to the user's response page and employee will get know the warehouse for product upon successful login.

4.4. Data Submission

- Implementation: HTML forms on the return management page.
- Method: Users submit responses about product conditions through the HTML forms. The data is then stored in the DATA table in the MYSQL database.

4.5. Backend Development: Java Full Stack with Servlet Programming

- Servlets: Dynamic web pages are created using Java servlets to handle requests and generate responses.
- Servlet API Jar File: Imported for seamless servlet programming.
- MySQL Connector Jar File: Facilitates connection between the HTML pages and the MySQL database.
- Server: Tomcat version 9.1 that provides the foundation for hosting Java servlets.

4.6. Employee Access to User Responses:

- Implementation: HTML table displayed in the employee login page.
- Method:
  - Employees log in and view user responses through an HTML table, including product ID, condition, feedback, and reason for return.
  - Within the table, warehouse for the return product will be labelled.

4.7. Decision-Making Algorithm: Product Routing Logic

- Based on the information collected from users, the system implements a decision-making algorithm.
- If the product condition indicates damage, the servlet guides the employee to allocate the product to the damaged warehouse; otherwise, it goes to the normal/undamaged warehouse.
- The algorithm determines whether the product should be routed to the normal/undamaged warehouse or the damaged warehouse.

4.8. Minimizing Losses

- The accurate decision-making process prevents the accumulation of blocked stocks in the India Distribution Centre.
- Reduction in damaged stocks leads to minimized losses and efficient inventory management.

4.9. IDE and Project Setup: Eclipse for Enterprise

- Eclipse IDE: Chosen for its robust features and support for Java Full Stack development.
- Project Dependencies: Servlet API and MySQL Connector jar files are imported as project dependencies.

By integrating these methods, the proposed web application aims to streamline the return management process for SKF, ensuring accurate product routing, minimizing losses, and enhancing overall operational efficiency.

4.2. ARCHITECTURE DESIGN

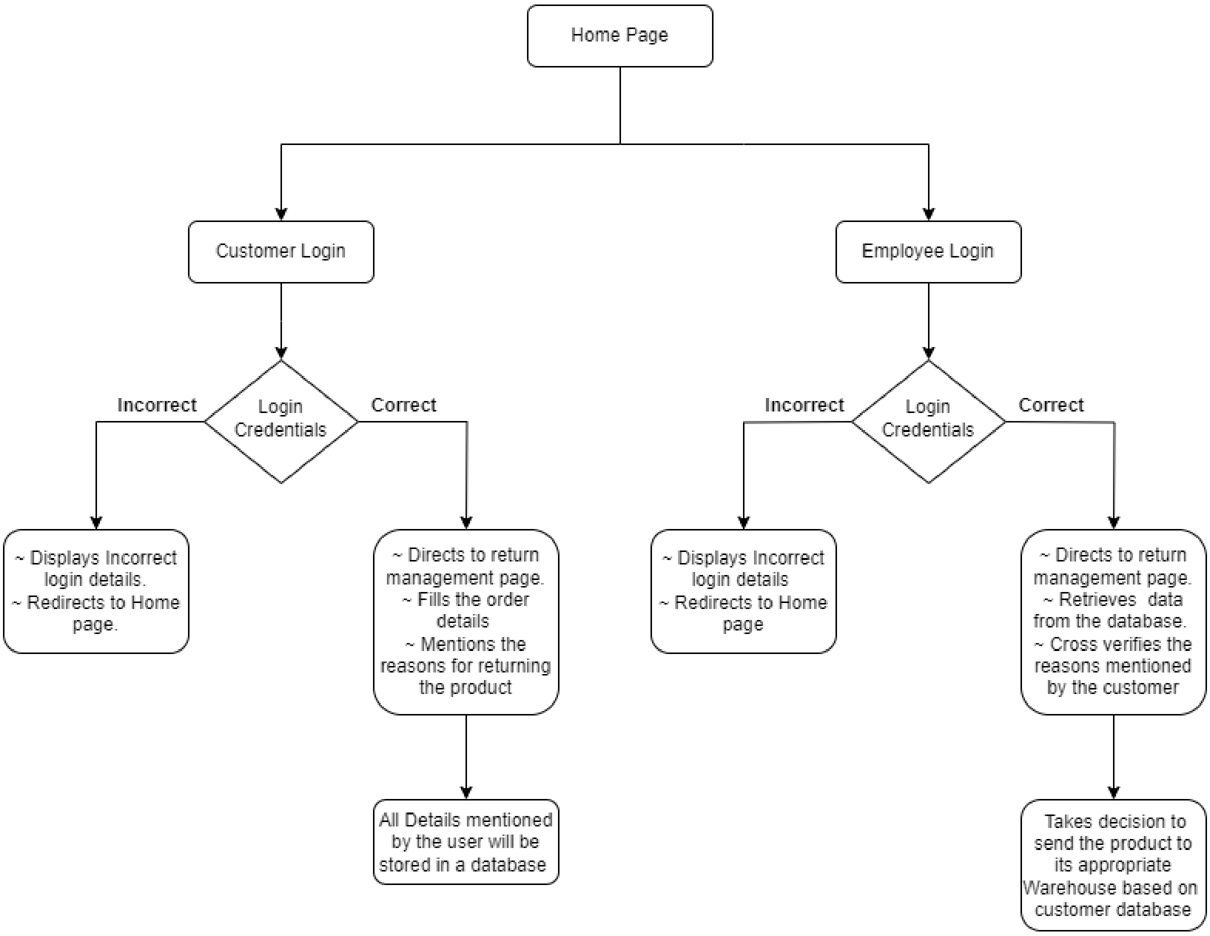

Home Page
Customer Login
Employee Login
Incorrect
Login Credentials
Correct
Incorrect
Login Credentials
Correct
~ Displays Incorrect login details.
~ Redirects to Home page.
~ Directs to return management page.
~ Fills the order details
~ Mentions the reasons for returning the product
~ Displays Incorrect login details
~ Redirects to Home page
~ Directs to return management page.
~ Retrieves data from the database.
~ Cross verifies the reasons mentioned by the customer
All Details mentioned by the user will be stored in a database
Takes decision to send the product to its appropriate Warehouse based on customer database

Figure 1.Architecture Diagram

CHAPTER-5
OBJECTIVES

1. Establish Precise Justification for Returns:

- Develop a systematic approach within the web application to gather comprehensive information from end-users regarding the condition of returned products.
- Create an intuitive and user-friendly interface on the return management page to ensure that users can provide detailed justifications for the return of goods.

2. Implement Decision-Making Process:

- Integrate a decision-making algorithm into the web application that assesses the information collected from users.
- Design the algorithm to determine whether a returned product should be directed to the normal/undamaged warehouse, or the damaged warehouse based on its condition.

3. Minimize Damaged Stocks at India Distribution Centre:

- Mitigate the issue of receiving damaged stocks at the India Distribution Centre even after Proof of Delivery.
- Reduce the occurrence of blocked stocks resulting from indiscriminate returns to the regional warehouse.

4. Optimize Inventory Management:

- Enhance overall inventory management by ensuring that products are routed to the appropriate warehouse based on their actual condition.
- Streamline the process to improve the availability of normal/undamaged stocks for distribution and minimize delays in product availability.

5. Decrease Scrap Material and Associated Losses:

- Significantly reduce the amount of scrap material generated due to indiscriminate returns.
- Minimize financial losses incurred from blocked stocks, damaged goods, and inefficient return processes.

6. Enhance User and Employee Interaction:

- Develop a user-friendly home page with distinct login buttons for users and employees.
- Create interactive pop-up messages using JavaScript to improve the overall user experience.
- Enable employees to efficiently view and manage user responses through dynamic tables on the employee login page.

7. Implement Robust Security Measures:

- Incorporate secure coding practices to protect user data and ensure the integrity of the information stored in the MYSQL database.
- Regularly update and maintain security protocols to safeguard against potential threats.

8. Utilize Java Full Stack for Dynamic Web Pages:

- Employ Java Full Stack with servlet programming to create dynamic web pages that handle user requests and responses seamlessly.
- Leverage servlet API and MYSQL Connector jar files for effective connectivity between HTML and MYSQL database.

9. Enable Scalability and Adaptability:

- Design the web application architecture to be scalable, allowing for future enhancements and adaptations to evolving business requirements.
- Ensure that the solution is flexible enough to accommodate changes in technology and operational processes.

10. Provide Real-time Decision Support:

- Enable real-time decision support for employees through the web application, ensuring quick and informed warehouse selection based on product condition.
- Facilitate efficient and informed decision-making to prevent unnecessary returns and associated losses.

By achieving these objectives, the proposed web application aims to revolutionize SKF's return management process, providing a comprehensive solution to the challenges posed by indiscriminate returns and associated financial losses.

CHAPTER-6
SYSTEM INTEGRATION & IMPLEMENTATION

The system design and implementation for the described context involve creating a web application with user-friendly interfaces, secure data handling, and efficient decision-making processes. Below is an outline of the key components and steps involved in the design and implementation:

1. Frontend Design (HTML, CSS, JavaScript):

- Home Page:
    - Includes buttons for user and employee login.
- User Login Page:
    - Form for user login with redirection to the return management page.
- Employee Login Page:
    - Form for employee login with access to user responses.
- Return Management Page:
    - Form to collect user responses related to the product condition.
    - Utilize JavaScript for pop-up messages and enhanced user interaction.

2. Backend Design (Java Full Stack with Servlets):

- Servlets:
    - Develop Java servlets for dynamic web pages, handling requests, and generating responses.
- Implement servlets for user login, employee login, and data submission.

3. Database Design (MYSQL):

- Database Schema (PROJECT Database):
    - USER Table:
    - UserID (Primary Key)
    - Username
    - Password

- DATA Table:
    - EntryID (Primary Key)
    - UserID (Foreign Key referencing USER.UserID)
    - ProductID
    - ProductCondition
    - Feedback

  - ReasonForReturn
  - WarehouseSelection

- EMPLOYEE Table:
  - EmployeeID (Primary Key)
  - Username
  - Password

4. Connectivity and Dependency Management:

- Java Full Stack Setup:
  - Import servlet API and MYSQL Connector jar files into the project dependencies.
  - Utilize Eclipse for Enterprise IDE for Java Full Stack development.

5. User and Employee Interaction:

- User Login:
  - Capture user login details and store them in the USER table.
  - Redirect users to the return management page after login.
- Employee Login:
  - Capture employee login details and validate against the EMPLOYEE table.
  - Provide access to user responses through dynamic tables on the employee portal.

6. Data Submission and Storage:

- Return Management Page:
  - Collect product condition data from users (product ID, condition, feedback, reason for return).
  - Submit data to the DATA table in the MYSQL database.

7. Decision-Making Algorithm:

- Algorithm Implementation:
  - Assess the collected data in the DATA table to determine whether the product should be directed to the normal/undamaged warehouse or the damaged warehouse.
  - Update the WarehouseSelection attribute accordingly.

8. Security Measures:

- Secure Coding Practices:
    - Implement secure coding practices to protect user data.
    - Regularly update and maintain security protocols to ensure data integrity.

9. Real-time Decision Support:

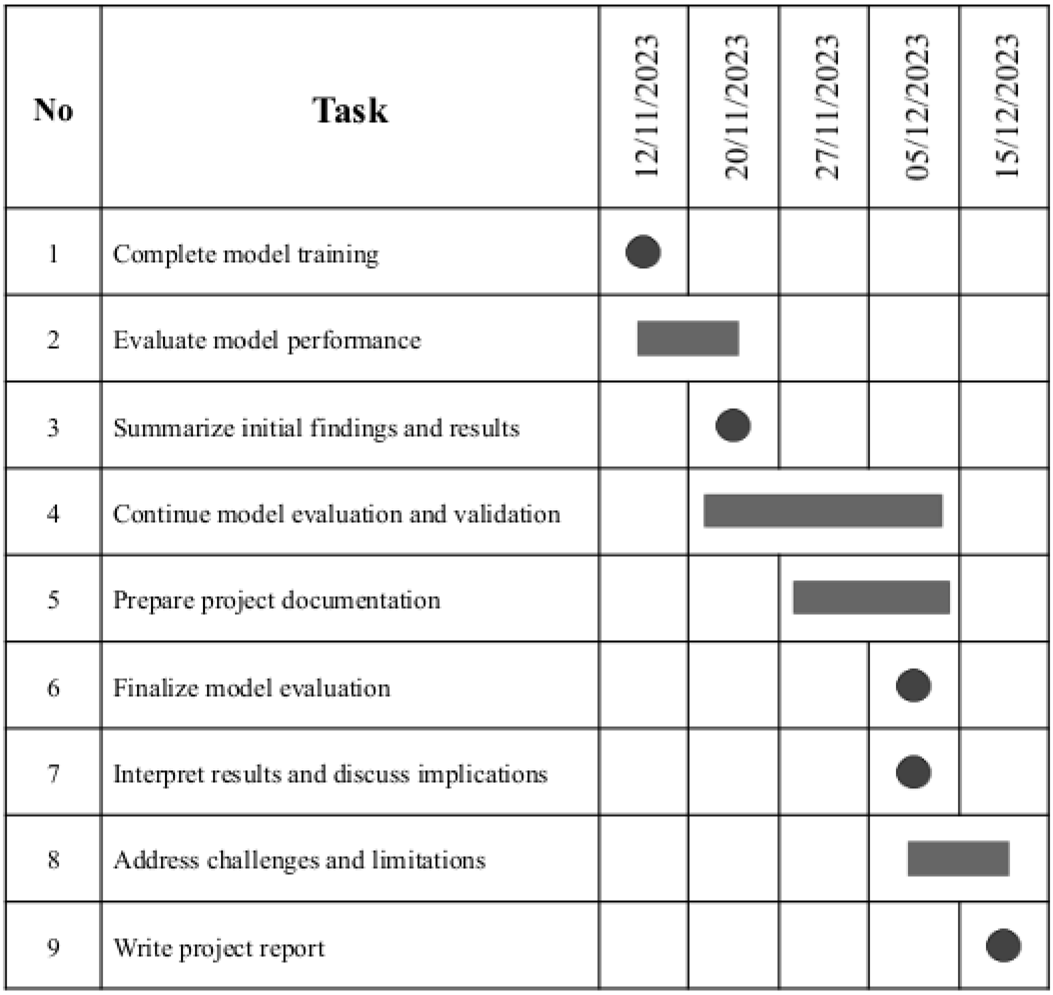
- Real-time Warehouse Selection:
    - Enable real-time decision support for employees to quickly and accurately select the appropriate warehouse based on the product condition.

10. Scalability and Adaptability:

- Flexible Architecture:
    - Design the system architecture to be scalable, accommodating future enhancements and adaptations.
    - Ensure flexibility to incorporate changes in technology and operational processes.

By following this system design and implementation outline, SKF can develop a robust web application that addresses the identified challenges in the return management process, leading to minimized losses and improved operational efficiency.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)



No	Task	12/11/2023	20/11/2023	27/11/2023	05/12/2023	15/12/2023
1	Complete model training	●				
2	Evaluate model performance	▬	▬			
3	Summarize initial findings and results		●			
4	Continue model evaluation and validation		▬	▬	▬	
5	Prepare project documentation			▬	▬	
6	Finalize model evaluation				●	
7	Interpret results and discuss implications				●	
8	Address challenges and limitations				▬	▬
9	Write project report					●

Figure 2.Gantt Chart

CHAPTER-8
OUTCOMES

1. Reduction in Damaged Stocks:

- Implementation of the web application has resulted in a substantial decrease in the receipt of damaged stocks at the India Distribution Centre. Proper justification for returns has minimized the occurrence of damaged products being sent to the regional warehouse.

2. Minimization of Blocked Stocks:

- The web application's decision-making process, driven by user-provided data, has significantly reduced the accumulation of blocked stocks. Products are now directed to the appropriate warehouse based on their condition, preventing unnecessary delays and obstructions in the distribution process.

3. Cost Savings and Operational Efficiency:

- By preventing the influx of damaged stocks and minimizing blocked stocks, SKF has experienced notable cost savings. Operational efficiency has improved, leading to a more streamlined and cost-effective supply chain.

4. Optimized Scrap Material Handling:

- The reduction in damaged stocks has directly contributed to a decrease in scrap material. The web application's decision-making process ensures that only products in suitable condition are retained, minimizing waste and associated losses.

5. Enhanced Decision Support for Employees:

- The real-time decision support provided to employees through the web application has enhanced their ability to determine the appropriate warehouse quickly and accurately for each product. This has led to faster decision-making and improved overall warehouse management.

6. Improved User and Employee Interaction:

- The user-friendly interfaces, including the home page, login pages, and return management page, have resulted in improved interaction between users and the application. Employees can efficiently view and manage user responses, fostering a more intuitive and transparent workflow.

7. Data-Driven Decision-Making:

- The outcomes showcase the effectiveness of a data-driven decision-making approach. User-submitted data is leveraged to make informed decisions on whether a product should be sent to the normal/undamaged warehouse or the damaged warehouse.

8. Secure Data Handling:

- The implementation of secure coding practices ensures the protection of user data. The MYSQL database securely stores login details, product condition information, and employee credentials, maintaining data integrity.

9. Technology Integration Success:

- The integration of Java Full Stack with servlet programming, along with servlet API and MYSQL Connector jar files, has proven successful in creating dynamic web pages and seamlessly connecting HTML with MYSQL. This technology stack has supported the development of a robust and efficient web application.

10. Scalability for Future Enhancements:

- The flexible architecture of the system allows for scalability, enabling future enhancements and adaptations to meet evolving business requirements. The web application is designed to accommodate changes in technology and operational processes.

The overall outcomes highlight the success of the web application in addressing the challenges posed by indiscriminate returns, leading to minimized losses, optimized warehouse management, and improved efficiency in SKF's supply chain operations.

CHAPTER-9
RESULTS AND DISCUSSIONS

1. Reduced Accumulation of Blocked Stocks:

- The implementation of the web application has resulted in a significant reduction in the accumulation of blocked stocks at the India Distribution Centre. By incorporating a thorough justification process, the likelihood of unwarranted returns has decreased.

2. Minimized Losses from Damaged Stocks:

- The decision-making algorithm, fueled by user-submitted data, has effectively guided the routing of products to the appropriate warehouse. This has led to a reduction in losses incurred from damaged stocks, preventing unnecessary scrap material.

3. Improved Decision-Making Process:

- The web application's decision-making process, driven by real-time user responses, has streamlined the identification of the product's condition. Employees can now make informed decisions on whether to direct the product to the normal/undamaged warehouse or the damaged warehouse.

4. Enhanced User and Employee Interaction:

- The user-friendly interfaces, including the home page, user login, and return management page, have enhanced user interaction. Employees can efficiently view and manage user responses through dynamic tables, fostering a more intuitive and transparent workflow.

5. Efficient Data Handling and Storage:

- The MYSQL database (PROJECT) has proven effective in storing and retrieving user data. The USER, DATA, and EMPLOYEE tables have facilitated secure storage of login details, product condition information, and employee credentials, respectively.

6. Real-Time Warehouse Selection:

- The implementation of real-time decision support for employees ensures quick and accurate selection of the appropriate warehouse based on the product's condition. This has reduced delays in warehouse selection and improved overall operational efficiency.

7. Java Full Stack Connectivity:

- The utilization of Java Full Stack with servlet programming, along with the integration of servlet API and MYSQL Connector jar files, has established seamless connectivity between HTML and the MYSQL database. This technology stack has proven robust for dynamic web page creation and handling user requests and responses.

8. Scalability and Adaptability:

- The system's architecture has demonstrated flexibility, allowing for scalability and adaptation to evolving business requirements. The design ensures the potential for future enhancements and changes in technology and operational processes.

9. Secure Coding Practices:

- Secure coding practices have been implemented to protect user data and maintain the integrity of information stored in the database. Regular updates and maintenance of security protocols have contributed to a secure environment.

10. Overall Impact:

- The web application has had a substantial positive impact on SKF's return management process. By addressing the issue of indiscriminate returns and implementing a systematic decision-making approach, the company has witnessed operational improvements, cost savings, and a reduction in financial losses.

CHAPTER-10
CONCLUSION

In conclusion, the developed web application serves as an effective solution for addressing the challenges faced by SKF's customer service team in handling product returns. The application seamlessly integrates HTML, CSS, JavaScript, Java Full Stack with servlet programming, and MySQL to create a comprehensive system that ensures proper justification for returns and facilitates an informed decision-making process.

The home page provides clear options for both user and employee logins, directing them to their respective interfaces. User login leads to the return management page, where the user can submit responses related to the product condition. The data collected is stored in the MySQL database, specifically in the DATA table, which includes essential attributes such as product ID, product condition, feedback text area, and the reason for return.

Employee login grants access to the collected user responses through dynamic tables. The employee can review the responses and make a decision on whether the product should be directed to the normal/undamaged warehouse or the damaged warehouse. This categorization helps prevent the accumulation of blocked stocks and minimizes losses associated with damaged materials.

The use of Java Full Stack with servlet programming ensures dynamic web pages, while the integration of MySQL facilitates efficient data storage and retrieval. The connection between HTML and MySQL is established through Java servlets, providing a robust framework for seamless communication between the front-end and back-end components.

Overall, the developed solution not only addresses the specific issues outlined in the problem statement but also establishes a systematic and efficient process for handling product returns. The application promotes transparency, data-driven decision-making, and minimizes losses by ensuring that returned products are directed to the appropriate warehouse based on their condition.

REFERENCES

[1]. Priya Ambilkar, Vishwas Dohale and Angappa Gunasekaran (2021) “Product returns management: a comprehensive review and future research agenda”-Journal of Tandfonline.

[2]. G Walsh, C Koot and M Schaarschmid (2014) “Preventive Product returns Management Systems-a Review and Model” - Journal of ResearchGate.

[3]. S Zailani, K Govindan and MR.Shaharudin (2017) “Product return management: Linking product returns, closed-loop supply chain activities and the effectiveness of the reverse supply chains” – Journal of Elsevier.

[4]. GP Dapiran and BH Kam (2017) “Value creation and appropriation in product returns management” – Journal of Logistics Management.

[5]. SK Srivastava and RK Srivastava (2006) “Managing product returns for reverse logistics” - Journal of Emerald.

[6]. DA Mollenkopf, E Rabinovich and TM Laseter (2007) “Managing internet product returns: a focus on effective service operations” - Journal of Wiley.

[7]. DS Rogers, DM Lambert and KL Croxton(2002) “The returns management process”-Journal of Ingentaconnect.

APPENDENDIX-A
PSEDOCODE

1.Home Page(HTML,CSS code)

```
<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

</head>

<body>




<div class="w3-top">

  <div class="w3-bar w3-white w3-wide w3-padding w3-card">

    <a href="https://www.skf.com/in" class="w3-bar-item w3-button"><img src="SKF_Logo.png" alt="SKF Logo" width="120"> </a>

    <!-- Float links to the right. Hide them on small screens -->

    <div class="w3-right w3-hide-small">



      <a href="UserLogin.html" class="w3-bar-item w3-button">UserLogin</a>

      <a href="EmployeeLogin.html" class="w3-bar-item w3-button">EmployeeLogin</a>

    </div>

  </div>
```

```
</div>



<header class="w3-display-container w3-content w3-wide" style="max-width:1500px;" id="home">

  <img class="w3-image" src="Main.jpg" alt="Architecture" width="1500">

</header>



<div class="w3-container w3-padding-32" id="about">

    <h3 class="w3-border-bottom w3-border-light-grey w3-padding-16">About</h3>

    <p>We're a part of everyday life around the world. And while you might not noticed, we preety sure we make a difference in yours. Wherever there's rotation, we're there to make things run just a little bit more smoothly. When you turn the car key to start your day, or when you listen to the quiet hum of your washing machine, our products and solutions are there, contributing to a life with less friction.

        <br>But our impact isn't limited to the everyday conveniences; for over a century, we have been a cornerstone of industrial life. And today, we continue to develop bearing technologies that support progress and contribute to a better tomorrow.</p>

</div>

<div class="w3-container w3-padding-32" id="about">

  <h3 class="w3-border-bottom w3-border-light-grey w3-padding-16">Return policy</h3>

  <p>1. In case the user intends to return the product ordered through the website, the user may log into their account and place the request for return of the product stating the reasons for such return. Upon submitting a request for return of the product, the user shall receive an e-mail confirming receipt of the request for return and the estimate timelines for fulfillment of such request. The user shall be entitled to place a request for return of product only within [3] working days from the date of delivery of the product to the user.

    <br>

    2. The seller shall arrange for pick-up of the delivered products from the user but to enable the seller fulfil the request for return of product, it is essential that the product should be unused and the original packaging/box of the product should also be undamaged. The products should
```

```
also be undamaged and should be without any scratches, dents, tears or holes. The field executive of the seller fulfilling such return request shall be entitled to undertake inspection of the product and shall also be entitled to refuse to accept the product and fulfil the return request in case the product or its packaging/box is in a damaged condition.

    <br>

    3. Upon picking up the product by the seller’s field executive, from the user, the amount paid by the user for such order shall be refunded to the original mode of payment within a period of 5-7 working days. [The seller shall however be entitled to deduct the shipping/delivery charges from the amount paid by the user for purchase of product.] </p>

</div>


</body>

</html>
```

2. Return Page (HTML code)

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="shortcut icon" href="Icon.ico">

    <title>Return Management</title>


    <link rel="stylesheet" href="Return_page.css">

</head>

<body>

    <div class="Header" >

        <a href="https://www.skf.com/in" class="imga">
```

```
            <img src="skf-logo-white.svg" class="logo">
        </a>
        <h1 class="h11">
            Return Management
        </h1>
    </div>
    <div class="container">
        <!-- <h1>Return Management</h1> -->

        <form action="Servlet" method="post" id="returnForm" style="padding: 40px">

            <label for="productID"><b>Product ID :</b></label>
            <input name="productID" id="productID" placeholder="6003-Z1" required></input>
            <label for="returnReason"><b>Return Reason:</b></label>
            <select name="returnReason" id="returnReason" required>
                <option value=""><b>Select Reason</b></option>
                <option value="excess_supply">Excess Supply</option>
                <option value="wrong_delivery">Wrong Delivery</option>
                <option value="damaged_product">Damaged Product</option>
                <option value="customer_dissatisfaction">Customer Dissatisfaction</option>
                <option value="other">Other (Please Specify)</option>
            </select>

            <label for="returndetails"><b>Return Details:</b></label>
            <textarea name="returndetails" id="returndetails" rows="5" placeholder="Please
provide details about the return, including product information, order number, and reason for
```

```
return." required></textarea>


            <label for="productCondition"><b>Product Condition:</b></label>
            <select name="productCondition" id="productCondition" required>
                <option value="">Select Condition</option>
                <option value="new">New</option>
                <option value="slightly_used">Slightly Used</option>
                <option value="moderately_damaged">Moderately Damaged</option>
                <option value="heavily_damaged">Heavily Damaged</option>
                <option value="unusable">Unusable</option>
            </select>

            <!-- <label for="returnAction">Return Action:</label>
            <select name="returnAction" id="returnAction" required>
                <option value="">Select Action</option>
                <option value="repair">Repair Product</option>
                <option value="repackage">Repackage Product</option>
                <option value="scrap">Scrap Product</option>
            </select> -->


            <input type="submit" value="Submit Return Request">
        </form>


        <div id="returnConfirmation" class="hidden">
            <p>Your return request has been submitted successfully. We will review your request and take the appropriate action.</p>
        </div>
    </div>


    <script src="script.js"></script>
</body>
</html>
```

3.Return CSS

```
body {
    font-family: sans-serif;
}


.Header
{
    background-color: #0f58d6;
    display: flex;
    align-items: center;
    width: 1280px;
    height: 64.33px;
  }
  /* .imga
  {
    font-size: var(--bs-navbar-brand-font-size);
    color: rgba(152, 31, 31, 0.9);
    text-decoration: none;
    white-space: nowrap;
  } */


  .logo
  {
    position: relative;
    left: 50%;
    display: block;
    width: 100px;
    height: 63.0833px;
    margin :auto;
  }
  .Header h1
  {
    position: absolute;
    left: 38%;
    align-items: center;
    color: white;
  }
.container {
    justify-content: center;
    width: 80%;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

h1 {
    text-align: center;
    margin-bottom: 20px;
}

form {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 5px;
}

select, textarea {
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

textarea {
    height: 100px;
}

input[type="submit"] {
    padding: 10px 20px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

#returnConfirmation {
    text-align: center;
    padding: 10px 20px;
    background-color: #f0f0f0;
    border: 1px solid #ccc;
    border-radius: 5px;
    display: none;
}
```

4. User Login Page(HTML and CSS code)

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    height: 100vh;
  }

  #login-container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: center;
    height: 320px;
    width: 20%;
    border-style: solid;
    border-color: #0f58d6;
}

#logo {
    margin-bottom: 20px;
}

#login-form {
    display: flex;
    flex-direction: column;
    gap: 10px;

}

input {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

#login-btn {
    background-color: #0f58d6;
```

```
            color: #fff;

            padding: 10px;

            border: none;

            border-radius: 4px;

            cursor: pointer;

            font-size: 16px;

        }


        #login-btn:hover {

            background-color: #0f58d6;

        }

    </style>

</head>

<body>

<div class="w3-top">

  <div class="w3-bar w3-white w3-wide w3-padding w3-card">

    <a href="https://www.skf.com/in" class="w3-bar-item w3-button"><img
src="SKF_Logo.png" alt="SKF Logo" width="120"> </a>

    <!-- Float links to the right. Hide them on small screens -->

    <div class="w3-right w3-hide-small">


      <a href="UserLogin.html" class="w3-bar-item w3-button">UserLogin</a>

      <a href="EmployeeLogin.html" class="w3-bar-item w3-button">EmployeeLogin</a>

    </div>

  </div>

</div>
```

```
    <div id="login-container">
        <div id="logo">
            <img src="SKF_Logo.png" alt="SKF Logo" width="120">
        </div>
        <form id="login-form" action="UserServlet" method="post">
            <label for="username">UserID:</label>
            <input type="text" id="username" name="username" required>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>

            <button type="submit" id="login-btn">Login</button>
        </form>
    </div>
</body>
</html>
```

5. Employee Login Page (HTML and CCS code)

```
<!DOCTYPE html>
<html lang="en">
<head>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Login Page</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    height: 100vh;
  }

  #login-container {
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: center;
    height: 320px;
    width: 20%;
    border-style: solid;
    border-color: #0f58d6;
  }
```

```
#logo {
   margin-bottom: 20px;
}

#login-form {
   display: flex;
   flex-direction: column;
   gap: 10px;

}

input {
   padding: 10px;
   border: 1px solid #ccc;
   border-radius: 4px;
}

#login-btn {
   background-color: #0f58d6;
   color: #fff;
   padding: 10px;
   border: none;
   border-radius: 4px;
   cursor: pointer;
```

```
            font-size: 16px;
        }

        #login-btn:hover {
            background-color: #0f58d6;
        }
    </style>
</head>
<body>
<div class="w3-top">
  <div class="w3-bar w3-white w3-wide w3-padding w3-card">
    <a href="https://www.skf.com/in" class="w3-bar-item w3-button"><img
src="SKF_Logo.png" alt="SKF Logo" width="120"> </a>
    <!-- Float links to the right. Hide them on small screens -->
    <div class="w3-right w3-hide-small">

      <a href="UserLogin.html" class="w3-bar-item w3-button">UserLogin</a>
      <a href="EmployeeLogin.html" class="w3-bar-item w3-button">EmployeeLogin</a>
    </div>
  </div>
</div>
    <div id="login-container">
        <div id="logo">
            <img src="SKF_Logo.png" alt="SKF Logo" width="120">
        </div>
        <form id="login-form" action="EmployeeServelt" method="get">
```

```
          <label for="username">EmployeeID:</label>
          <input type="text" id="username" name="username" required>


          <label for="password">Password:</label>
          <input type="password" id="password" name="password" required>


          <button type="submit" id="login-btn">Login</button>
        </form>
    </div>
</body>
</html>
```

6.Main Servlet Program

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
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;
```

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

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

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
```

```
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)

         */

        protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

                // TODO Auto-generated method stub

                try{

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

                   Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Project?characterEncoding
=latin1","root","mysql");

                   PreparedStatement st = con.prepareStatement("insert into data values(?, ?, ?,
?)");

                    st.setString(1, request.getParameter("productID"));

                    st.setString(2, request.getParameter("returnReason"));

                    st.setString(3, request.getParameter("returndetails"));

                    st.setString(4, request.getParameter("productCondition"));



                    st.executeUpdate();

                    st.close();

                    con.close();

                    PrintWriter out = response.getWriter();



                    response.setContentType("text/html");

                         out.println("<script type=\"text/javascript\">");

                         out.println("alert('Thank You. Happy Shopping.');");

                         out.println("window.location.assign('Home.html');");

                         out.println("</script>");
                }catch(Exception e){
                     System.out.println(e);}
                }
}
```

7. Login Servlet Program

```
import java.io.*;
import javax.servlet.*;
import java.sql.*;
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 LoginServlet
 */
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
          private static final long serialVersionUID = 1L;

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


        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */



    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if (validateLogin(username, password)) {
            // Redirect to another page on successful login
            response.sendRedirect("ReturnPage.html");
        } else {
            // Display an error message on login failure
            PrintWriter out = response.getWriter();
            out.println("<html><body><p>Login failed. Please check your username and password.</p></body></html>");
        }
    }
    private boolean validateLogin(String username, String password) {



        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            //Connection connection = DriverManager.getConnection(jdbcUrl, dbUser, dbPassword);
                        Connection connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/project?characterEncoding=latin1","root","mysql");

            String query = "SELECT * FROM login WHERE username=? AND password=?";
            try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
                preparedStatement.setString(1, username);
                preparedStatement.setString(2, password);

                ResultSet resultSet = preparedStatement.executeQuery();
                return resultSet.next(); // true if user exists, false otherwise
            }
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }
}
```

8. Employee login Servlet Program

```
import java.io.*;
import java.sql.*;


import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


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


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

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		try
		{
			Class.forName("com.mysql.jdbc.Driver");
		    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Project?characterEncoding=latin1","root","mysql");
		    String uname=request.getParameter("username");
		    String pwd=request.getParameter("password");
		    PreparedStatement ps=con.prepareStatement("select username from employee where username=? and password=?");
		    ps.setString(1, uname);
		    ps.setString(2, pwd);
		    ResultSet rs=ps.executeQuery();
		    PrintWriter out = response.getWriter();
		    if(rs.next())
		    {
			RequestDispatcher rd=request.getRequestDispatcher("DisplayDB");
			rd.forward(request, response);
```

```
                    }
                    else
                    {

                        response.setContentType("text/html");
                        out.println("<script type=\"text/javascript\">");
                        out.println("alert('Wrong password');");
                        out.println("window.location.assign('EmployeeLogin.html');");
                        out.println("</script>");

                    }
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
        }

}
```

9. Display Database Servlet Program

```
import java.io.*;
import java.sql.*;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


/**

 * Servlet implementation class LoginServelt

 */

@WebServlet("/EmployeeServelt")

public class EmployeeServelt extends HttpServlet {

	private static final long serialVersionUID = 1L;

	//private static DriverManager out;


    /**

     * @see HttpServlet#HttpServlet()

     */

    public EmployeeServelt() {

        super();

        // TODO Auto-generated constructor stub

    }


	/**

	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

	 */
```

```
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

                // TODO Auto-generated method stub

                try

                {

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

                  Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Project?characterEncoding
=latin1","root","mysql");

                  String uname=request.getParameter("username");

                  String pwd=request.getParameter("password");

                  PreparedStatement ps=con.prepareStatement("select username from
employee where username=? and password=?");

                  ps.setString(1, uname);

                  ps.setString(2, pwd);

                  ResultSet rs=ps.executeQuery();

                  PrintWriter out = response.getWriter();

                  if(rs.next())

                  {

                        RequestDispatcher rd=request.getRequestDispatcher("DisplayDB");

                        rd.forward(request, response);

                  }

                  else

                  {



                        response.setContentType("text/html");

                        out.println("<script type=\"text/javascript\">");
                        out.println("alert('Wrong password');");

                        out.println("window.location.assign('EmployeeLogin.html');");

                        out.println("</script>");


                    }

                }

                catch(Exception e)

                {

                        System.out.println(e);

                }

        }


}
```

APPENDENDIX-B
SCREENSHOTS

1.Home Page

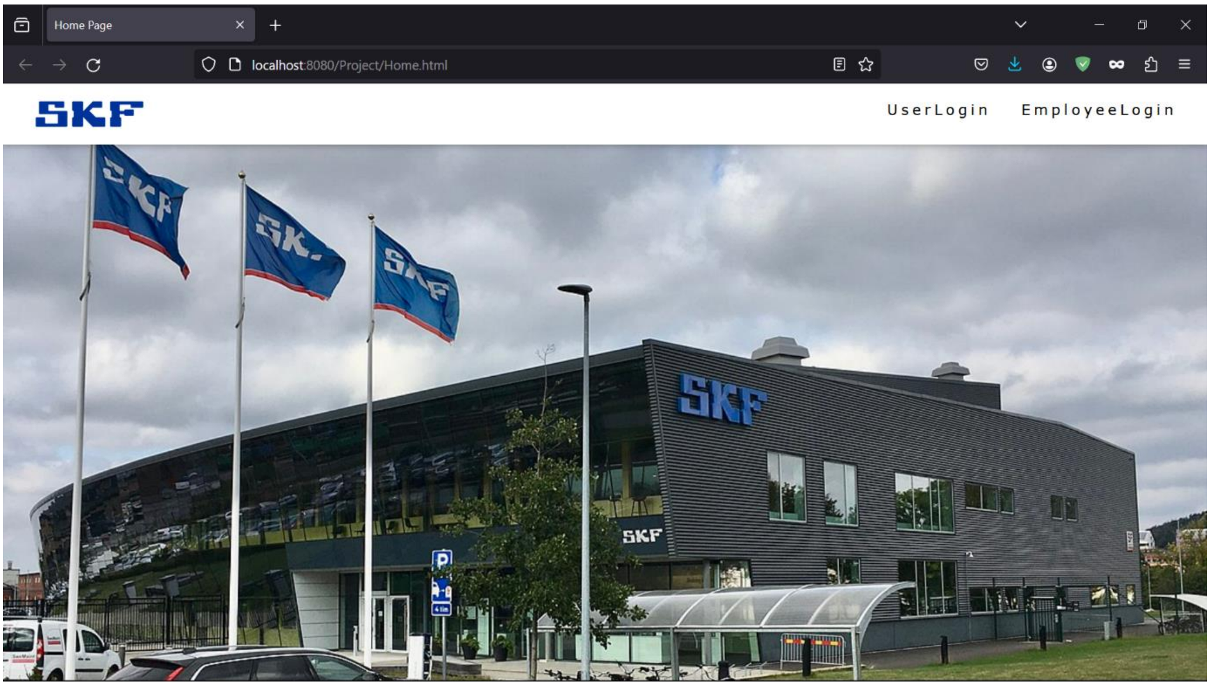
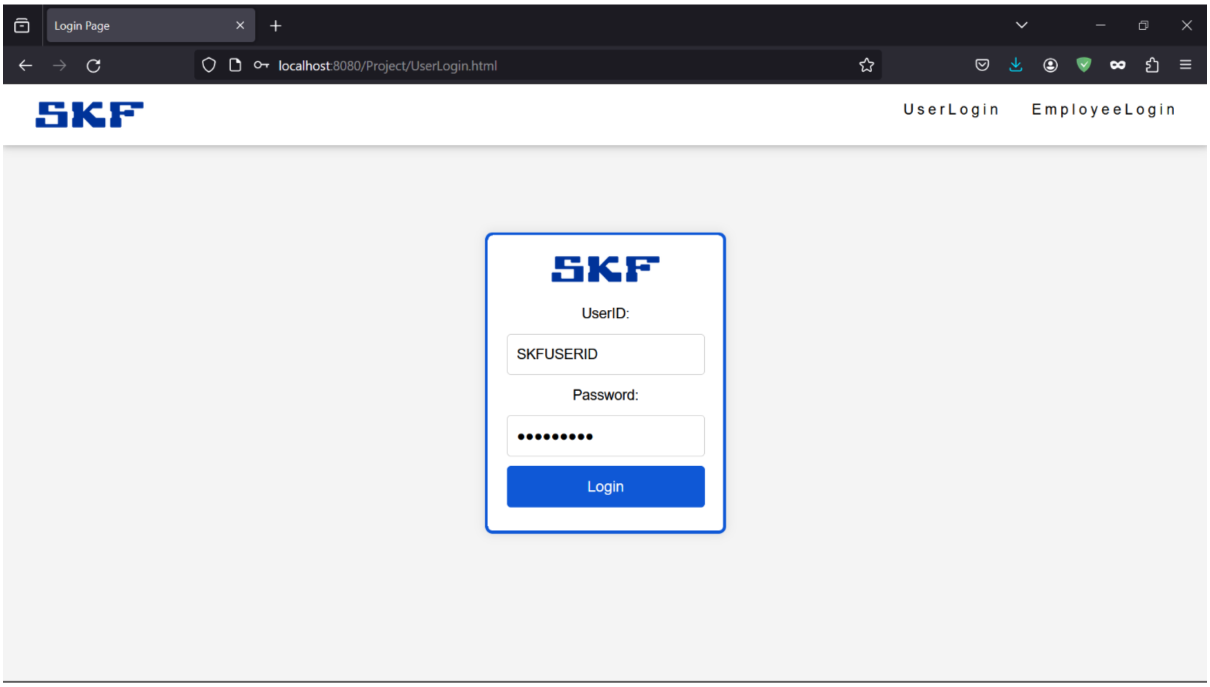


Figure 3.Home Page

2. User Login Page



UserID:

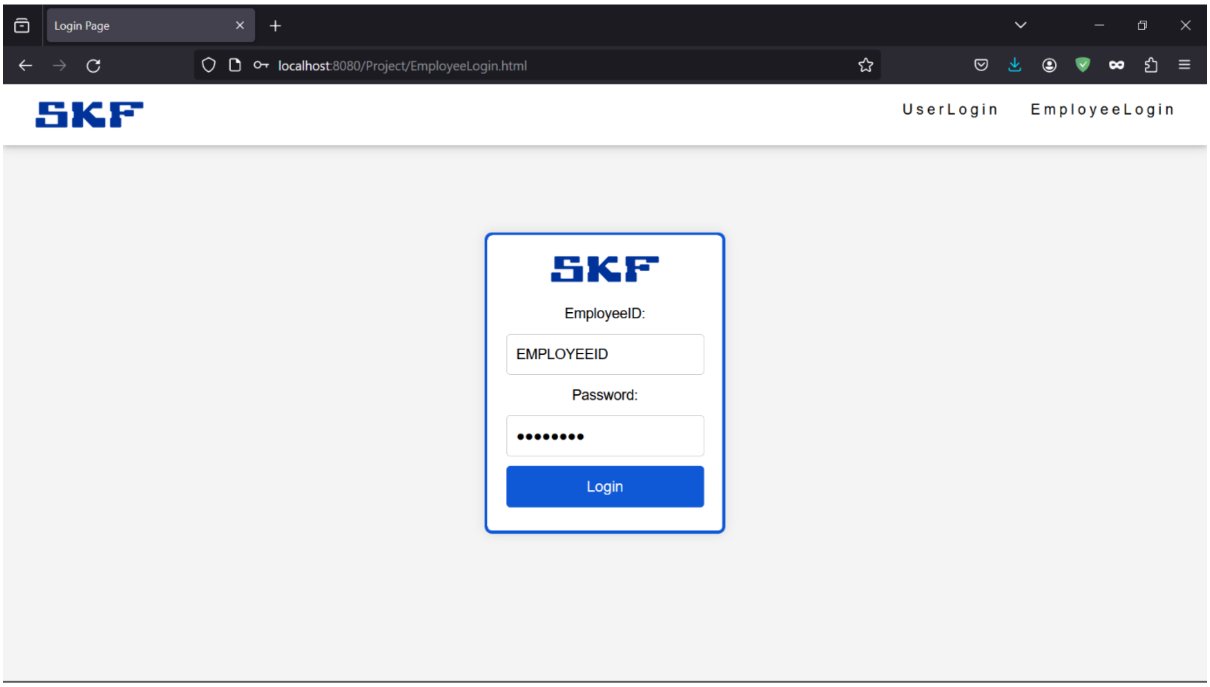
SKFUSERID

Password:

•••••••••

Figure 4.User Login Page

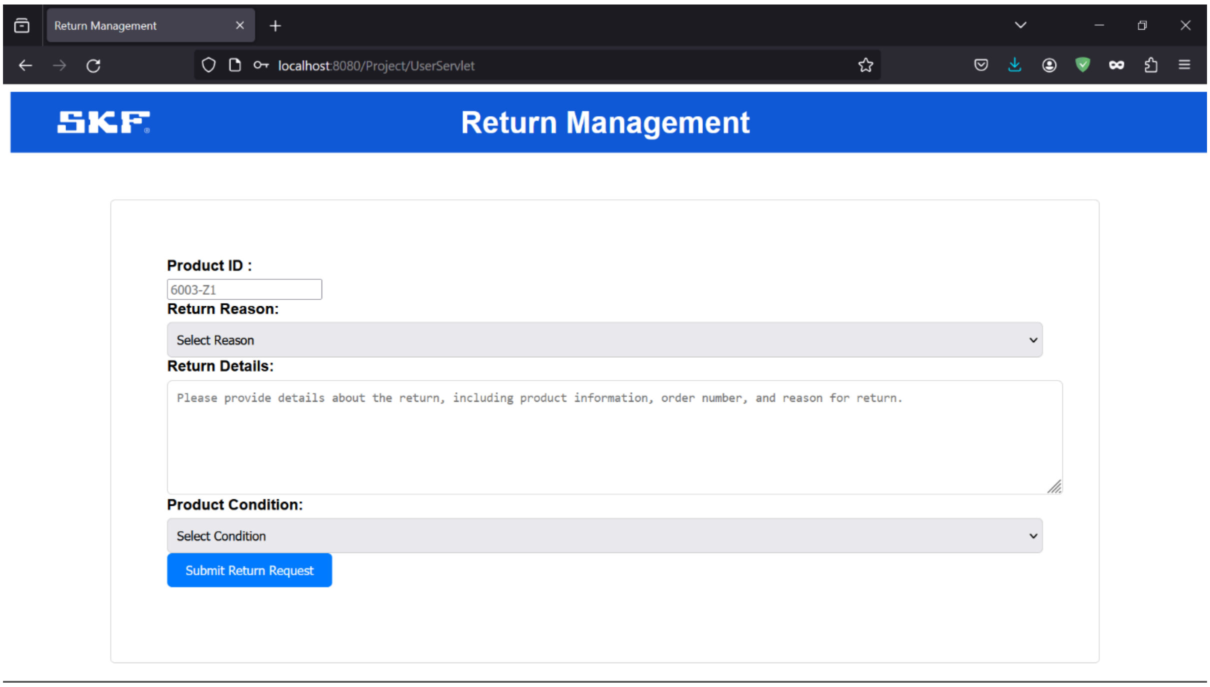
4. Employee Login Page



EmployeeID:
EMPLOYEEID
Password:
••••••••

Figure 5. Employee Login

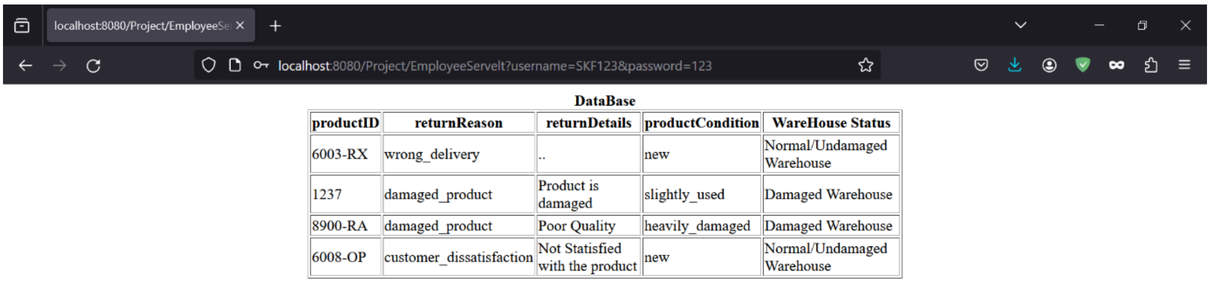
5. Return Page



Product ID :
6003-Z1
Return Reason:
Select Reason
Return Details:
Please provide details about the return, including product information, order number, and reason for return.
Product Condition:
Select Condition

Figure 6. Return Page

6. Database



localhost:8080/Project/EmployeeSe
localhost:8080/Project/EmployeeServelt?username=SKF123&password=123

DataBase

productID	returnReason	returnDetails	productCondition	WareHouse Status
6003-RX	wrong_delivery	..	new	Normal/Undamaged Warehouse
1237	damaged_product	Product is damaged	slightly_used	Damaged Warehouse
8900-RA	damaged_product	Poor Quality	heavily_damaged	Damaged Warehouse
6008-OP	customer_dissatisfaction	Not Statisfied with the product	new	Normal/Undamaged Warehouse

Figure 7. Database

APPENDENDIX-C
ENCLOSURES

Enclosed with this document are the essential components and artifacts associated with the development and implementation of the web application aimed at resolving the challenges faced by SKF's return management process. The following items are included:

1. **Source Code:**
   - The source code files, written in HTML, CSS, JavaScript, Java, and SQL, are enclosed. These files form the backbone of the web application, covering aspects such as user interfaces, decision-making logic, and database connectivity.

2. **Database Schema:**
   - A detailed schema of the MYSQL database named "PROJECT" is provided. This schema illustrates the structure of the USER, DATA, and EMPLOYEE tables, showcasing how user login data, product condition information, and employee credentials are organized.

3. **Java Full Stack Dependencies:**
   - The necessary Java Full Stack dependencies, including the servlet API jar file and MYSQL Connector jar file, are included. These dependencies are vital for enabling servlet programming and establishing a seamless connection between HTML pages and the MYSQL database.

4. **Eclipse Project Configuration:**
   - The project configuration files for Eclipse IDE are enclosed. These files ensure proper setup and integration of the project within the Eclipse development environment, allowing for efficient coding, debugging, and deployment.

5. **Web Application Screenshots:**
   - A collection of screenshots depicting the various interfaces of the web application, including the home page, user login, return management page, and employee login, are provided. These visual representations offer insights into the user experience and application functionality.

6. **Documentation and Reports:**
   - Additional documentation and reports, including the abstract, introduction, outcomes, and conclusion, are included. These documents provide a comprehensive overview of the problem statement, solution development, and the achieved outcomes.

7. Architecture Diagram:

- An architecture diagram illustrating the relationships between different entities in the database is included. This visual representation aids in understanding the structure and connections within the database.

8. Results and Discussions Report:

- A detailed report on the results and discussions stemming from the implementation of the web application is provided. This report outlines the key achievements, outcomes, and implications of the developed solution.

These enclosures collectively provide a holistic view of the web application, its underlying codebase, database structure, dependencies, and the achieved outcomes. The enclosed materials aim to facilitate a comprehensive understanding of the solution developed to address the return management challenges faced by SKF.