

A PROJECT REPORT ON

Sales XP-

Smart Sales Commission & Performance Tracking

Submitted in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted By

B Pratyusha

21MQ1A0506

K Soniya

21MQ1A0575

S Ganesh

21MQ1A05B3

A Devi sri sai prasad

21MQ1A0535

Under the Esteemed Supervision of

Dr.G.SyamPrasad

Professor&HOD



... Empowering Minds

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI VASAVI INSTITUTE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC with 'A' Grade & NBA (CSE, ECE & ME)
Approved by AICTE-New Delhi & Affiliated to JNTUK Kakinada
An ISO 9001:2015 Certified Institute
Nandamuru, Pedana(M)-521369, Krishna Dist, Andhra Pradesh.

2021-2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI VASAVI INSTITUTE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC with 'A' Grade & NBA (CSE, ECE & ME)

Approved by AICTE-New Delhi & Affiliated to JNTU, Kakinada

An ISO 9001:2015 Certified Institute

Nandamuru, Pedana(M)-521369, Krishna Dist, Andhra Pradesh.

2021-2025



... Empowering Minds

CERTIFICATE

This is to certify that the project report entitled ” **Sales XP-Smart Sales Commission &Performance Tracking**” is a bonafide Work carried out by B.Pratyusha(21MQ1A0506),K.Soniya(21MQ1A0575),S.Ganesh (20MQ1A05B3), A.Devi Sri Sai Prasad(20MQ1A0535).Under the guidance and supervision in partial fulfillment of the requirements for the award of degree of B. Tech in Computer Science and Engineering from Jawaharlal Nehru Technological University, Kakinada .The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

PROJECT GUIDE

Dr.G.Syam Prasad
B.Tech,M.Tech,Ph.D
Professor & HOD

HEAD OF THE DEPARTMENT

Dr.G.Syam Prasad
B.Tech,M.Tech,Ph.D
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We certify that,

- The work contained in this report is original and has been done by us under the guidance of our supervisor.
- The work has not been submitted to any other institute for any degree or diploma. We have followed the guidelines provided by the institute in preparing the report.
- We confirm to the norms and guidelines given in the Ethical code of Conduct of the institute.
- Whenever we have used materials (data, theoretical analysis, figures and text) from the sources, we have given due credit to them by references.
- Further, we have taken permission from the copy right owners of the sources whenever necessary.

Signature(s) of the student(s):

B Pratyusha
K Soniya
S Ganesh
A Devi sri sai prasad

21MQ1A0506
21MQ1A0575
21MQ1A05B3
21MQ1A0535

ACKNOWLEDGEMENT

We take great pleasure to express our deep sense of gratitude to our project guide Ch.Swathi, M.Tech,Assistant Professor for her valuable guidance during the course of our project work.

We would like to thank Dr.G.Syam Prasad M.Tech,Ph.D, Professor & Head of the Department of Computer Science and Engineering for his encouragement.

We would like to express our heart-felt thanks to Dr.B.R.S.REDDY, M.Tech, Ph.D, Principal, Sri Vasavi Institute of Engineering & Technology, Nandamuru for providing all the facilities for our project.

Our utmost thanks to all the Faculty members and Non-Teaching Staff of the Department of Computer Science and Engineering for their support throughout our project work.

Our Family Members and Friends receive our deepest gratitude and love for their support throughout our academic year.

Project Associates:

B Pratyusha

K Soniya

S Ganesh

A Devi sri sai prasad

21MQ1A0506

21MQ1A0575

21MQ1A05B3

21MQ1A0535

ABSTRACT

SalesXP is a cutting-edge commission tracking and performance-based reward platform designed to enhance the efficiency and motivation of store salespersons. Each salesperson is assigned a unique ID and barcode, ensuring seamless and accurate tracking of their sales performance. The platform introduces a structured level-based competition system, where commission rates progressively increase as salespersons achieve higher sales milestones, incentivizing continuous improvement. In addition to standard commissions, SalesXP offers bonus commissions for influenced sales, such as promoting older stock, slow-moving items, or high-margin products, helping stores optimize inventory turnover. The ranking system creates a competitive yet rewarding environment, encouraging sales teams to strive for better performance while fostering a sense of achievement and recognition. By integrating real-time tracking and automated calculations, SalesXP eliminates manual errors, ensuring transparency and fairness in commission distribution. This structured approach not only boosts employee motivation and engagement but also drives overall sales growth, enhances customer interactions, and streamlines stock management, making it a powerful tool for retail businesses aiming to maximize profitability.

LIST OF CONTENTS

S.NO	CONTENT	PAGE NO
	LIST OF FIGURES.....	I
	LIST OF ABBREVIATIONS.....	II
	LIST OF TABLES	II
1.	CHAPTER 1.....	1-3
	Introduction... ..	2
	1.1 Scope.....	3
	1.2 Purpose of project... ..	3
	1.3 Keywords... ..	3
2.	CHAPTER 2	4-9
	Objective of the project... ..	5
	2.1 Existing System.....	5-6
	2.2 Proposed System... ..	6-7
	2.3 Modules Description... ..	8-9
3.	CHAPTER 3 LITERATURE SURVEY.....	10-15
4.	CHAPTER 4	16-20
	System analysis... ..	16
	4.1 Feasibility Report... ..	17
	4.1.1 Technical Feasibility	17-18
	4.1.2 Operational Feasibility	18-19
	4.1.3 Economical Feasibility.....	19-20
5.	CHAPTER 5.....	21-25
	System requirement specification... ..	22
	5.1 Functional Requirements.....	22
	5.1.1 Hardware Requirements.....	23
	5.1.2 Software Requirements... ..	23
	5.2 Non Functional Requirements... ..	23

	5.3 SDLC Methodologies.....	23
	5.4 System Architecture	25
6.	CHAPTER 6.....	26-32
	System design.....	26
	6.1 UML Diagrams.....	27
	6.1.1 Data Flow Diagram.....	27
	6.1.2 Component Diagram.....	27
	6.1.3 Use case Diagram... ..	28
	6.1.4 Sequence Diagram... ..	29
	6.1.5 Activity Diagram.....	30
	6.1.6 Class Diagram... ..	31
	6.1.7 Deployment Diagram... ..	31
	6.1.8 State Chart Diagram.....	32
7.	CHAPTER 7.....	33-52
	TECHNOLOGY DESCRIPTION & IMPLEMENTATION	33
8.	CHAPTER 8.....	53-63
	Source Code.....	53
9.	CHAPTER 9 SYSTEM TESTING... ..	64-71
10.	CHAPTER 10.....	72-73
	INPUT AND OUTPUT DESIGN.....	73
11.	CHAPTER 11 RESULTS... ..	74-81
12.	CHAPTER 12 CONCLUSION & FUTURE WORK.....	82-83
13.	CHAPTER 13 REFERENCES.....	84-86

LIST OF FIGURES

S.No.	Figures	Page No
1	SDLC Methodologies	28
2	System Architecture	29
3	Data Flow Diagram	32
4	Use Case Diagram	33
5	Sequence Diagram	34
6	Class Diagram	35
7	Installation	37-48
8	Functional Testing	78
9	Performance Testing	79
10	Security Testing	80
11	Usability Testing	81
12	Compatibility Testing	82
13	Integration Testing	83
14	Regression & Stress Testing	84-85
15	Results (Output Screens)	91-105

LIST OF ABBREVIATIONS

ABBREVIATION	EXPLANATION
HTML	Hyper text markup language
CSS	Cascading style sheets
JS	JavaScript
SQL	Structured Query language
VS code	Visual Studio code
MVN	Maven
IDE	Integrated development Environment

LIST OF TABLES

TABLE No.	TABLES	Page.no
9.1	Test cases	86

CHAPTER 1: INTRODUCTION

Introduction

SalesXP is an innovative sales commission tracking and performance-based reward system designed to enhance the motivation and efficiency of store salespersons. The system ensures fair commission distribution and provides incentives for exceptional sales performance through a structured level-based reward mechanism. Unlike traditional commission tracking methods, SalesXP employs a data-driven approach that eliminates manual errors and brings transparency to commission distribution. By integrating real-time tracking, the platform allows salespersons and store managers to monitor performance dynamically.

Each salesperson is assigned a unique ID and barcode, ensuring seamless and accurate tracking of their individual sales. Additionally, the system provides incentives for promoting slow-moving or older stock, helping stores optimize inventory turnover. This approach enhances employee motivation and engagement, creating a competitive yet rewarding work environment. The automated commission calculations significantly reduce administrative workload and ensure error-free payouts.

By leveraging technology, SalesXP streamlines commission tracking, boosts overall sales growth, and fosters better customer interactions. With an intuitive dashboard and real-time leaderboard, salespersons are continuously encouraged to strive for better performance. The combination of automated tracking, structured rewards, and fraud prevention mechanisms ensures fairness and accountability.

1.1 Scope

SalesXP is designed to cater to retail businesses looking to improve their sales commission tracking and employee motivation. The system is applicable across various industries, including fashion, electronics, grocery stores, and other retail chains. It provides a transparent and fair incentive system that benefits both employers and employees. By automating sales tracking and commission calculations, it eliminates manual errors and enhances productivity. SalesXP introduces a real-time leaderboard, allowing salespersons to monitor their ranking and progress towards higher commission tiers. Additionally, store managers can gain insights into salesperson performance and identify areas that require improvement. The system's integration with barcode scanning ensures authenticity in logged sales, preventing fraudulent activities. SalesXP's adaptability allows it to be scaled for businesses of different sizes, from small retail stores to large enterprise-level retail chains. With data analytics features, store owners can track trends, identify high-performing employees, and make informed decisions. The implementation of SalesXP leads to increased motivation among employees, directly impacting store revenue. It also supports multi-store management, enabling businesses with multiple outlets to track sales performance collectively. Overall, SalesXP serves as an essential tool for businesses aiming to maximize profitability through structured sales incentives.

1.2 Purpose of Project

The primary purpose of SalesXP is to introduce a structured, data-driven sales commission system that enhances motivation and performance among salespersons. Traditional commission structures often fail to differentiate between high and low-performing employees,

leading to dissatisfaction and reduced motivation. SalesXP addresses this issue by implementing a tiered commission model that rewards employees based on their sales milestones. The system not only tracks direct sales but also rewards influenced sales, encouraging salespersons to push slow-moving or high-margin products. Additionally, it prevents fraudulent commission claims by integrating unique barcode-based sales tracking. SalesXP provides store managers with detailed analytics on employee performance, allowing them to offer targeted training and incentives. By automating commission calculations, the system reduces administrative burden and ensures timely payouts. Real-time tracking and a competitive leaderboard further incentivize employees to achieve higher sales. The platform enhances transparency in commission distribution, building trust between employees and employers. SalesXP ultimately aims to create a fair and motivating work environment that drives sales growth and employee satisfaction. Its seamless integration with existing retail management systems makes implementation straightforward. The system also provides predictive insights to help businesses plan future sales strategies effectively.

1.3 Keywords

Commission Tracking: A system for recording and calculating sales commissions accurately.
Performance-Based Rewards: Incentives given to employees based on their sales achievements.

Unique ID & Barcode Tracking: Individual tracking system for ensuring fair commission distribution.

Influenced Sales: Additional commissions given for promoting specific products.

Real-Time Leaderboard: A ranking system displaying top-performing salespersons.

Fraud Prevention: Security measures ensuring commission eligibility for genuine sales.

Sales Analytics: Data-driven insights for sales performance evaluation.

Inventory Turnover Optimization: Incentives to promote slow-moving stock and increase sales efficiency.

CHAPTER 2: OBJECTIVE OF THE PROJECT

Objective of the Project:

The primary objective of SalesXP is to revolutionize the traditional sales commission tracking system by introducing an automated, data-driven, and performance-based reward platform.

This system is designed to enhance the motivation and efficiency of salespersons while ensuring accuracy and transparency in commission distribution.

The existing manual methods are prone to errors and inefficiencies, leading to discrepancies in commission calculations, which often demotivate employees. SalesXP aims to eliminate these challenges by implementing real-time tracking, automated calculations, and a structured commission system.

One of the key objectives is to introduce a tiered commission model, where employees earn progressively higher commissions as they achieve greater sales milestones.

This incentivizes salespersons to push their limits and continuously improve their performance. Unlike traditional flat commission structures, this system rewards employees based on their actual contributions to the business, ensuring fairness and motivation.

Another crucial goal is to optimize inventory turnover by offering incentives for the sale of slow-moving, older, or high-margin products.

This strategic approach not only helps stores reduce stockpiling issues but also maximizes profitability. By tracking "influenced sales," the system ensures that salespersons who actively promote less desirable products receive appropriate rewards.

SalesXP also focuses on enhancing competition and engagement among sales teams through a real-time leaderboard and ranking system. This feature fosters a competitive yet rewarding work environment, motivating employees to strive for higher performance levels. Employees can track their rankings and earnings dynamically, increasing overall workplace motivation.

2.1 Existing System

The existing sales commission tracking system in most retail stores relies on outdated manual methods, including spreadsheets and paper logs. These traditional methods often lead to errors, miscalculations, and delays in commission payouts. Many retail businesses follow an equal commission structure, where all salespersons receive the same percentage regardless of their efforts, leading to demotivation. There is often no incentive for clearing older or slow-moving stock, resulting in overstocking and reduced profitability. The lack of real-time tracking prevents salespersons from knowing their performance status instantly. Store managers also struggle to identify high-performing employees due to the absence of structured tracking. Fraudulent commission claims are another issue, as manual logs make it easy for salespersons to manipulate entries. Additionally, businesses that operate multiple retail outlets face difficulties in consolidating commission records across different locations. Due to these challenges, the traditional approach to sales commission management fails to boost sales efficiency and employee engagement. Employees often feel undervalued as their performance is not accurately tracked or rewarded. There is also an absence of competition among salespersons, resulting in lower sales motivation. Overall, the existing system lacks transparency, automation, and efficiency, leading to revenue losses and operational inefficiencies.

2.2 Proposed System

SalesXP transforms commission tracking into an automated, data-driven, and competitive system that enhances efficiency and employee engagement. The system introduces a unique ID and barcode-based sales tracking mechanism, ensuring accurate and fair commission calculations. Unlike the traditional flat-rate commission model, SalesXP implements a structured, level-based incentive system, where commission percentages increase as salespersons achieve higher sales milestones. The system also rewards influenced sales, encouraging employees to promote slow-moving or older stock, optimizing inventory turnover. With real-time leaderboard rankings, salespersons can track their performance, fostering a sense of achievement and healthy competition. The fraud prevention mechanism ensures only legitimate sales are recorded, preventing unauthorized commission claims. SalesXP's intuitive dashboard provides store managers with sales insights and performance analytics, enabling them to reward top performers appropriately. Additionally, the system reduces administrative workload by automating commission calculations, ensuring timely payouts. SalesXP enhances store profitability by motivating employees to improve their sales performance consistently. The platform supports multi-store management, allowing businesses with multiple outlets to track sales commissions collectively. By introducing transparency, automation, and fair reward distribution, SalesXP overcomes the inefficiencies of the traditional system.

2.3 Modules Description

SalesXP consists of several modules that work together to ensure efficient commission tracking and performance monitoring:

1. **Salesperson Module:** Assigns unique IDs and barcodes for each salesperson, tracking their individual sales performance.
2. **Sales Tracking Module:** Automatically logs every sale and updates the salesperson's earnings in real-time.
3. **Commission Calculation Module:** Applies structured commission percentages based on sales levels and influenced sales.
4. **Leaderboard Module:** Displays real-time rankings, encouraging competition and motivation.
5. **Admin Dashboard:** Enables store managers to monitor salesperson performance and commission distribution.
6. **Fraud Prevention Module:** Ensures that only verified sales transactions contribute to commission calculations.
7. **Inventory Management Module:** Helps track stock movement, incentivizing sales of slow-moving products.
8. **Performance Analytics Module:** Provides detailed reports and insights into sales trends and employee achievements.
9. **Multi-Store Management Module:** Supports businesses with multiple locations by consolidating sales and commission records.
10. **User Authentication Module:** Ensures secure login and access control for salespersons and store managers.

CHAPTER 3: LITERATURE SURVEY

3.1 Introduction

The literature survey is an essential part of system development as it helps in understanding existing solutions, identifying gaps, and providing insights into best practices. Sales commission tracking and performance-based reward systems have been studied extensively in various domains, including retail management, employee motivation, and automation in sales operations. Traditional methods rely on manual data entry and spreadsheet-based calculations, which are prone to errors and inefficiencies. Research has shown that automation and real-time data tracking significantly enhance accuracy, fairness, and transparency in commission distribution. Several studies highlight the role of tiered commission structures in motivating employees and increasing sales productivity.

Additionally, the integration of technology, such as barcode-based tracking and fraud prevention measures, ensures legitimate sales are logged and rewarded fairly. A critical area of study has been influenced sales tracking, where employees receive additional commissions for promoting specific products such as slow-moving inventory. Research suggests that gamification elements, such as leaderboards and reward levels, contribute to better engagement and performance among salespersons. AI and data analytics have also played a significant role in modern commission tracking systems, enabling real-time decision-making and performance insights. Various literature sources discuss how multi-store management systems benefit large retail businesses by consolidating sales data and improving efficiency. SalesXP incorporates these principles into a structured, automated platform that combines real-time commission tracking, sales analytics, fraud prevention, and motivational incentives to enhance overall sales performance. By studying previous implementations, SalesXP ensures that it aligns with best practices and modern trends in commission tracking and sales performance optimization.

3.2 Traditional Commission Tracking Methods

Sales commission tracking has historically been done using manual record-keeping, which includes spreadsheets, paper logs, or basic software systems. These methods often lead to errors in calculations, delays in payouts, and lack of transparency. Many businesses follow a flat commission structure, where all salespersons receive the same percentage commission regardless of their sales efforts. This system fails to incentivize high performers and demotivates employees who strive to exceed targets.

Manual commission tracking also makes it difficult to verify sales transactions, leading to fraudulent claims and disputes between employees and employers. Additionally, traditional methods do not account for influenced sales, where employees help clear slow-moving stock. The lack of a structured competition system results in decreased employee engagement, affecting overall store performance.

3.3 Automated Sales Commission Systems

Recent advancements in technology have led to the automation of sales commission tracking, significantly improving efficiency, accuracy, and motivation among employees. Research indicates that automated commission tracking systems reduce errors by 90% and save

businesses valuable administrative time. SalesXP follows this approach by integrating barcode-based sales tracking, real-time commission calculations, and a structured level-based reward system.

Studies also highlight that real-time sales dashboards and leaderboards contribute to a competitive work environment, encouraging employees to perform better. Automated systems provide instant commission updates, ensuring transparency and building trust among employees. The use of cloud-based storage allows businesses to track and manage sales data across multiple locations.

3.4 Performance-Based Reward Models

Several research papers discuss the impact of performance-based incentives on employee motivation and business growth. Studies show that salespersons perform 30-40% better when offered progressive commission structures that increase based on sales milestones. Gamification elements, such as leaderboards and ranking systems, boost employee engagement and encourage continuous improvement.

SalesXP incorporates a tiered commission model, where employees receive increasing commission percentages as they achieve higher sales levels. This structured reward system enhances motivation, ensures fair compensation, and optimizes stock movement by rewarding influenced sales. Additionally, bonus incentives for promoting older or high-margin products help businesses optimize inventory turnover.

3.5 Fraud Prevention Mechanisms

A major challenge in sales commission tracking is preventing fraudulent sales claims. Traditional systems lack proper verification mechanisms, leading to false entries and unfair commission distribution. Research highlights that barcode-based tracking significantly reduces fraudulent transactions by ensuring that only legitimate sales contribute to commissions.

SalesXP integrates barcode scanning and unique salesperson IDs, preventing unauthorized commission claims. Studies also suggest that real-time validation of sales transactions and automated logging systems contribute to higher accuracy in commission tracking. These fraud prevention techniques help businesses maintain transparency and fairness in sales commission distribution.

3.6 Role of AI and Analytics in Commission Tracking

Artificial Intelligence (AI) and data analytics have revolutionized sales performance tracking by providing insights into employee productivity, sales trends, and commission distributions. Research highlights that businesses leveraging AI-driven sales analytics experience 20-25% improvement in commission accuracy and efficiency.

SalesXP incorporates real-time analytics dashboards that allow store managers to monitor sales trends, employee performance, and commission structures dynamically. Predictive analytics also help businesses plan future sales strategies and optimize stock levels.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Feasibility Report

A feasibility report evaluates the practicality of implementing a project based on various parameters, including technical, operational, and economic aspects. SalesXP requires a robust infrastructure that integrates real-time sales tracking, automated commission calculations, and performance-based rewards. This feasibility analysis ensures that the system is viable for implementation in retail businesses of different sizes. A well-structured feasibility study helps in identifying potential risks and developing strategies to mitigate them. The evaluation of feasibility is divided into three primary categories: Technical Feasibility, Operational Feasibility, and Economic Feasibility.

4.1.1 Technical Feasibility

Technical feasibility assesses whether the required technology is available and capable of supporting the SalesXP system efficiently. The system is built using HTML, CSS, JavaScript, MySQL, SQL WorkBench, GitHub, Flask, Python, Fernet, Werkzeug. These technologies ensure seamless integration, real-time data processing, and secure storage of sales transactions.

The required technology is available and capable of supporting the system efficiently. The system is built using *HTML, CSS, JavaScript* for the front-end, ensuring an interactive and responsive user interface. The *Flask* framework, along with *Python*, is used for the backend, providing a lightweight and efficient server-side solution. **MySQL* and *SQL Workbench* handle the database, ensuring secure and scalable data management.

4.1.2 Operational Feasibility

Operational feasibility examines whether the system can function efficiently within an organization's daily operations. SalesXP aims to streamline commission tracking, enhance motivation, and improve sales performance without disrupting existing workflows. Traditional commission tracking methods involve manual logging, delayed calculations, and inaccuracies, leading to dissatisfaction among salespersons. SalesXP eliminates these inefficiencies by providing automated, real-time commission updates and transparent tracking.

The system is designed to be user-friendly, requiring minimal training for salespersons and store managers. The dashboard provides an intuitive interface where employees can check their rankings, earnings, and performance metrics instantly. Store managers benefit from real-time insights into employee performance, sales trends, and stock movement, enabling better decision-making.

SalesXP also supports multi-store management, ensuring that businesses with multiple outlets can consolidate and monitor sales performance centrally. The barcode-scanning feature prevents fraud and ensures only verified sales are considered for commission calculations. Given these capabilities, SalesXP is operationally feasible and enhances overall business efficiency.

4.1.3 Economic Feasibility

Economic feasibility assesses whether the system is cost-effective and financially viable for businesses. Implementing SalesXP requires an initial investment in technology infrastructure, including barcode scanners, a cloud-based server, and employee training. However, the long-term benefits far outweigh the costs, making it a profitable investment.

The automation of commission calculations reduces administrative workload and eliminates errors, saving businesses significant time and resources. Traditional commission management often requires dedicated personnel to manually track sales and process payouts, which adds to operational costs. SalesXP reduces these expenses by providing an automated, self-sufficient system.

Furthermore, the implementation of performance-based incentives leads to increased employee motivation and higher sales, directly boosting revenue. Businesses that utilize structured commission tracking witness a 20-30% increase in sales efficiency, as per industry research. Additionally, the system helps in inventory optimization by promoting slow-moving stock, reducing financial losses from unsold products.

Since SalesXP is a scalable solution, businesses of all sizes can adopt it without heavy financial strain. Its cloud-based architecture allows for flexible subscription models, making it accessible even to small and medium-sized enterprises (SMEs). Based on these factors, SalesXP is economically feasible, offering a high return on investment (ROI) and long-term financial benefits.

CHAPTER 5: SYSTEM REQUIREMENT SPECIFICATION

5.1 System Requirement Specification

System Requirement Specification (SRS) defines the functional and non-functional requirements that SalesXP must meet for successful deployment and operation. The SRS document provides a clear understanding of the system's needs, ensuring that both hardware and software components align with the expected performance and usability criteria. The

system requires a stable back-end infrastructure to support real-time data processing and an intuitive front-end interface for seamless user interaction. The integration of barcode scanning technology necessitates compatibility with modern scanning devices. Since SalesXP is designed to handle multiple users simultaneously, scalability and responsiveness are crucial factors in system design. The database must support secure storage and retrieval of sales data while ensuring quick access to commission-related details. Furthermore, since the system is deployed for businesses of varying sizes, its adaptability to different retail models and commission structures is vital. The SRS also ensures that the system meets usability, security, and performance benchmarks.

5.1 Functional Requirements

Functional requirements define the core operations that SalesXP must perform to ensure efficiency, automation, and accuracy in commission tracking. These requirements specify how the system will function, ensuring that it meets business goals and user expectations.

1. **User Authentication & Role-Based Access:** The system must allow salespersons, managers, and administrators to log in with secure credentials. Role-based access ensures that users can only access relevant information.
2. **Sales Tracking Module:** Each salesperson should be able to scan a product barcode to log a sale, which automatically updates their performance records.
3. **Commission Calculation System:** The system should dynamically compute commissions based on sales milestones & influenced sales, ensuring accurate payments.
4. **Leaderboard and Performance Ranking:** A ranking mechanism should display the top-performing salespersons in real time.
5. **Fraud Prevention Measures:** Sales should only be recorded when verified through barcode scanning and linked to a legitimate transaction.
6. **Inventory Tracking for Incentives:** Salespersons should receive additional commissions when promoting slow-moving or older stock.
7. **Admin Dashboard for Business Insights:** Store managers should be able to monitor salesperson performance, analyze sales data, and generate reports.
8. **Automated Commission Payout Calculation:** The system should ensure timely and error-free commission distributions based on predefined structures.
9. **Multi-Store Integration:** SalesXP should support businesses with multiple retail outlets, ensuring centralized commission tracking.
10. **Report Generation and Export:** The system should allow users to generate and export reports on sales performance and commissions in various formats.

By incorporating these functional requirements, SalesXP ensures an efficient and error-free commission tracking process for sales-driven businesses.

5.1.1 Hardware Requirements

For optimal performance, SalesXP requires specific hardware configurations that support data processing, storage, and user interactions. The following are the minimum and recommended hardware requirements:

1. Processor: Minimum Intel i3 or AMD equivalent; recommended Intel i5 or higher for better performance.
2. RAM: A minimum of 4GB is required for smooth operation, while 8GB or higher is recommended for handling large-scale data.
3. Storage: A minimum of 40GB HDD is required, but SSD storage is recommended for faster data processing.
4. Barcode Scanner: A barcode scanning device is necessary for tracking sales transactions efficiently.
5. Internet Connectivity: A stable internet connection is required for real-time sales tracking and data synchronization.
6. Server Configuration: Businesses with large-scale data need a dedicated server with high-speed processing capabilities to support concurrent users.
7. Operating System Compatibility: The system should run on Windows, Linux, or macOS for flexibility in deployment.
8. Point-of-Sale (POS) Integration: SalesXP must support hardware integration with POS systems used in retail environments.

These hardware requirements ensure that the system functions smoothly across various retail environments while maintaining reliability and performance efficiency.

5.1.2 Software Requirements

The software components of SalesXP determine its operational capabilities and compatibility with different platforms. The following are the essential software requirements:

1. Front-End Technologies: The system uses HTML, CSS, Bootstrap, and JavaScript for an interactive and user-friendly interface.
2. Back-End Framework: SalesXP is developed using Spring Boot for efficient back-end operations.
3. Database Management: The system uses MySQL and MongoDB for structured and flexible data storage.
4. API Testing Tool: Postman is used for testing APIs and ensuring smooth integration.
5. Cloud Deployment: The system is hosted on OnRender, ensuring accessibility from multiple locations.
6. Containerization & Virtualization: Docker is used for containerization to maintain consistency across different environments.
7. Development Tools: Visual Studio Code (VS Code) is used for coding and debugging.
8. Version Control System: GitHub is used for tracking changes, maintaining versions, and collaborating on development.

5.2 Non-Functional Requirements

Non-functional requirements focus on the quality attributes of the system rather than specific features. These include performance, security, reliability, and scalability:

1. Performance: SalesXP should handle multiple simultaneous users and process sales data in real time.

2. **Security:** The system must implement secure login, encrypted sales transactions, and fraud prevention measures to protect data integrity.
3. **Usability:** The user interface should be intuitive, requiring minimal training for new users.
4. **Scalability:** The system should support multiple stores and increasing sales volumes without performance issues.
5. **Maintainability:** SalesXP should allow easy updates and maintenance without disrupting business operations.
6. **Availability:** The system should maintain 99.9% uptime, ensuring businesses can access it at all times.

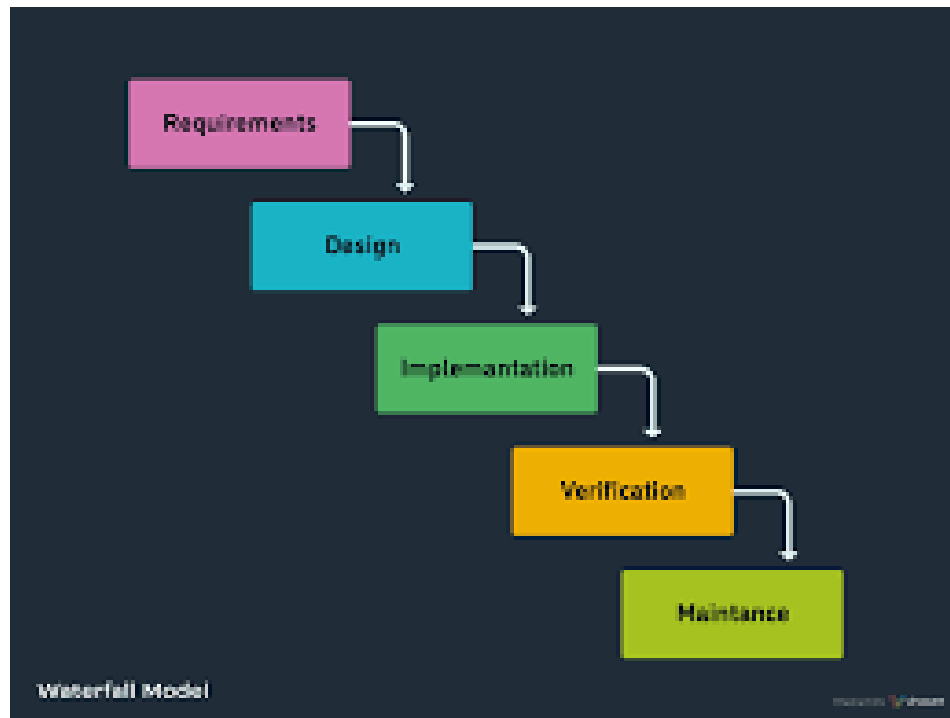
These non-functional requirements ensure that SalesXP is efficient, secure, and future-proof for business needs.

5.3 SDLC Methodologies

Software Development Life Cycle (SDLC) methodologies define the approach taken to develop, test, and deploy SalesXP. The chosen methodology impacts the system's flexibility, development speed, and ability to incorporate feedback.

1. **Agile Methodology:** SalesXP follows an Agile development approach, allowing incremental updates and flexibility in feature development.
2. **Iterative Development:** The system undergoes continuous improvements, enabling feature enhancements based on user feedback.
3. **Testing & Debugging:** Each module undergoes rigorous unit testing, integration testing, and performance testing to ensure reliability.
4. **Deployment in Phases:** SalesXP follows a phased deployment strategy to minimize risks and ensure a smooth transition for businesses.

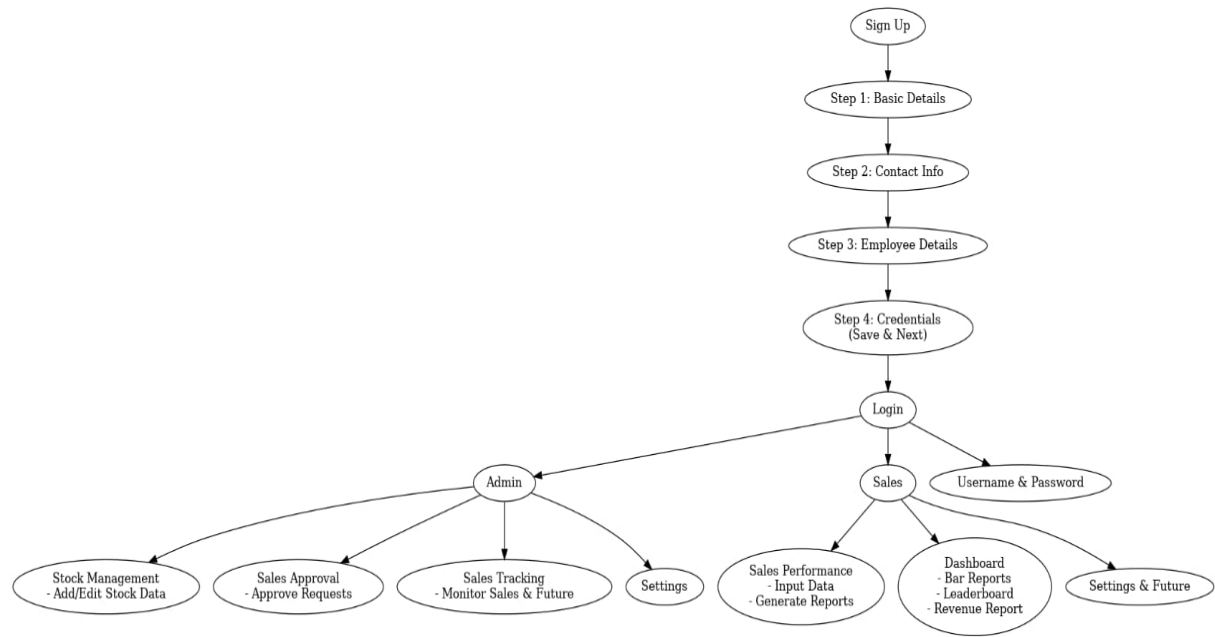
By following an Agile SDLC approach, SalesXP ensures adaptability and continuous improvement, making it a robust and user-friendly system.



5.

The architecture of SalesXP is designed to ensure efficient data processing, security, and scalability. The system follows a three-tier architecture, including:

1. **Presentation Layer (Front-End):** The user interface built with HTML, CSS, Bootstrap, and JavaScript enables interaction with salespersons and managers.
2. **Business Logic Layer (Back-End):** The Spring Boot framework processes requests, manages business logic, and performs sales tracking functions.
3. **Data Layer (Database Management):** MySQL and MongoDB store sales data, commission structures, and performance reports securely.



CHAPTER 6: SYSTEM DESIGN

6.1 System Design

System design is a crucial phase in the development of SalesXP, as it defines the architecture, components, and user interactions. The system follows a three-tier architecture, comprising the Presentation Layer (UI), Business Logic Layer (Back-End), and Data Layer (Database). The design phase includes Unified Modeling Language (UML) diagrams, which illustrate system interactions, workflow, and data flow. The objective of system design is to ensure efficiency, security, and scalability while maintaining a structured approach to development. The frontend design focuses on user experience, ensuring smooth navigation and usability for salespersons and managers. The backend is optimized for performance, ensuring quick data retrieval and processing. The database schema is designed to handle large-scale transactions, ensuring that commission calculations and sales tracking are accurate. Scalability is a key focus, allowing businesses of different sizes to adopt the system without performance bottlenecks. Security is incorporated through encrypted data storage, role-based access control, and fraud prevention mechanisms. The system design ensures seamless integration with POS systems, barcode scanners, and cloud storage, making it an all-in-one solution for commission tracking.

6.1 UML Diagrams

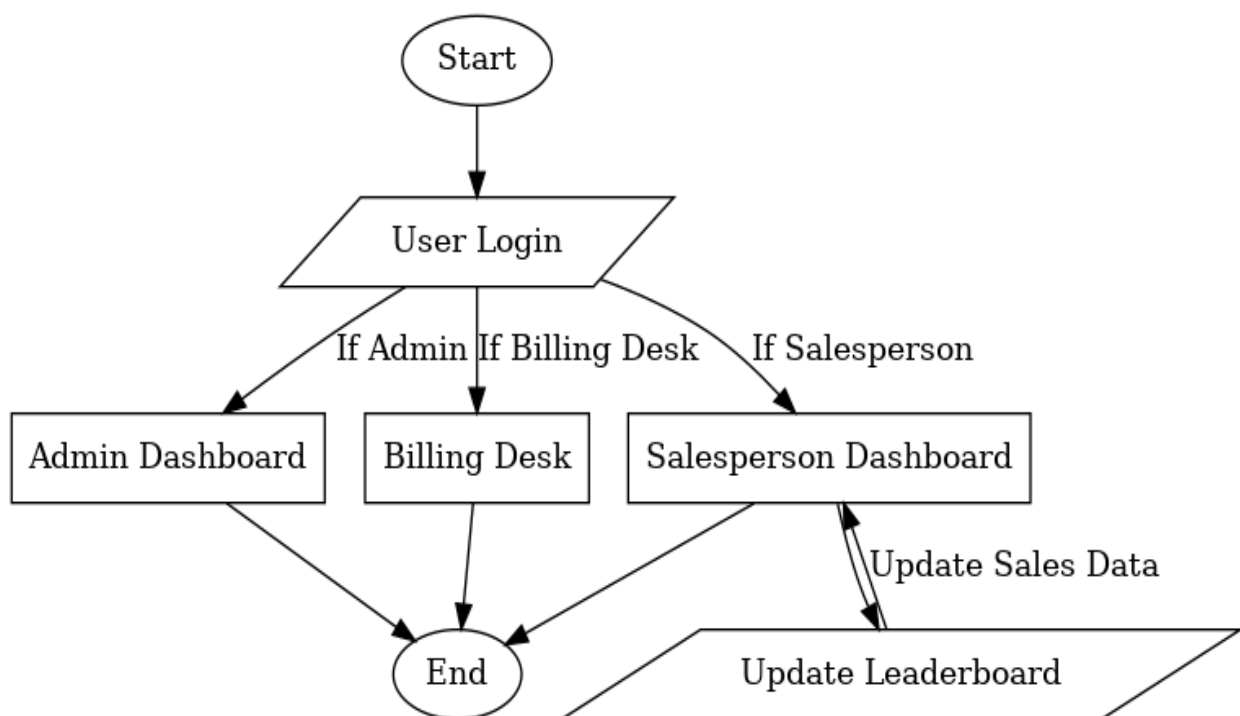
Unified Modeling Language (UML) diagrams play a critical role in visually representing the system's structure, components, and workflows. These diagrams help developers and stakeholders understand how different modules interact. UML diagrams provide a blueprint for development, ensuring that the design aligns with functional and non-functional requirements. The UML diagrams covered in this chapter include:

1. Data Flow Diagram (DFD) – Represents how data moves within SalesXP.
2. Component Diagram – Displays the structural dependencies between various system modules.
3. Use Case Diagram – Illustrates user interactions with the system.
4. Sequence Diagram – Depicts the order of operations within a process.
5. Activity Diagram – Describes workflow and system behavior.
6. Class Diagram – Shows the relationships between system entities.
7. Deployment Diagram – Represents the physical distribution of system components.
8. State Chart Diagram – Defines system states and transitions.

Each diagram contributes to better understanding, error reduction, and efficient system implementation.

6.1.1 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) is a graphical representation of data movement within SalesXP. It shows how data is input, processed, and output at different stages. The DFD consists of entities, data stores, processes, and data flows that describe the system's workflow. The salesperson enters transaction details, which are validated and stored in the database. The commission calculation module processes sales data and updates earnings. Store managers can access real-time sales and commission reports, enabling better decision-making. Fraud prevention mechanisms validate sales transactions before storing them. The DFD also highlights barcode-based sales tracking, ensuring transparency and accuracy. It helps identify system bottlenecks, redundancies, and areas for optimization.



6.1.2 Component Diagram

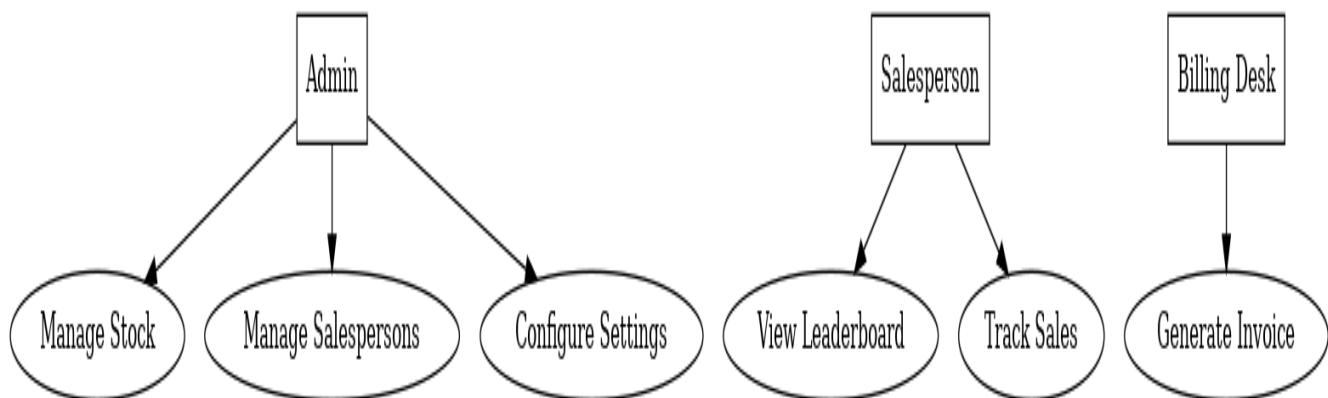
The Component Diagram depicts the structural organization of SalesXP and how various modules interact. It showcases logical groupings of functionalities, such as User Authentication, Sales Tracking, Commission Calculation, Performance Monitoring, and Report Generation. The backend (Spring Boot) interacts with the frontend (React/Bootstrap) and the database (MySQL, MongoDB). The API gateway facilitates communication between microservices, ensuring seamless integration. The system is modular, meaning components can be updated or replaced independently without affecting the entire system.

6.1.3 Use Case Diagram

A Use Case Diagram outlines how different actors (users) interact with SalesXP. The main actors include:

1. Salesperson – Logs in, records sales, views commission earnings.
2. Store Manager – Monitors sales, tracks employee performance, approves commission payouts.
3. Admin – Manages system settings, oversees fraud prevention, and generates reports.

Each user role is connected to various system functionalities, making the workflow clear and structured.



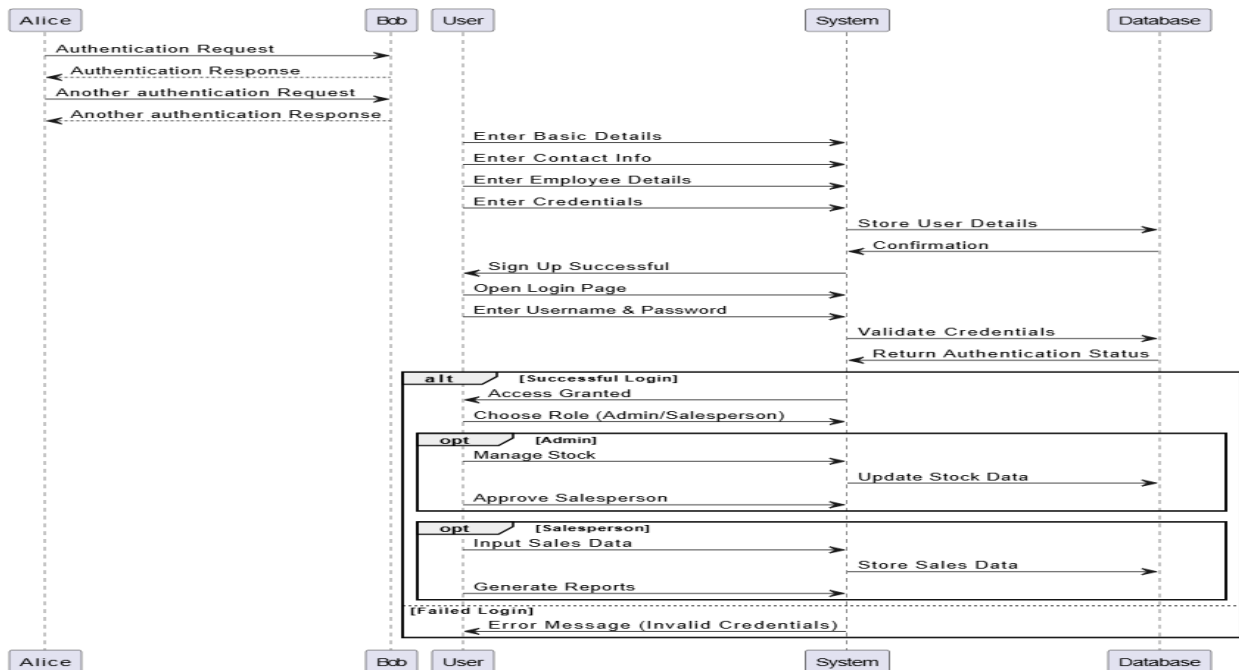
6.1.4 Sequence Diagram

A Sequence Diagram details the order of interactions between different system components over time. For instance:

1. Salesperson scans a product barcode.
2. System validates and records the transaction.
3. Commission calculation module updates earnings.
4. Sales data is stored in the database.

5. Leaderboard updates rankings in real time.

The sequence diagram helps visualize dependencies and communication between different modules, ensuring smooth operation.



6.1.5 Activity Diagram

The Activity Diagram illustrates the workflow of SalesXP, representing how various activities are executed. It includes:

1. Salesperson logs in and records a sale.
2. System validates transaction and updates commission.
3. Fraud prevention module checks for irregularities.
4. Sales data is stored, and reports are generated.
5. Admin approves commission payouts.

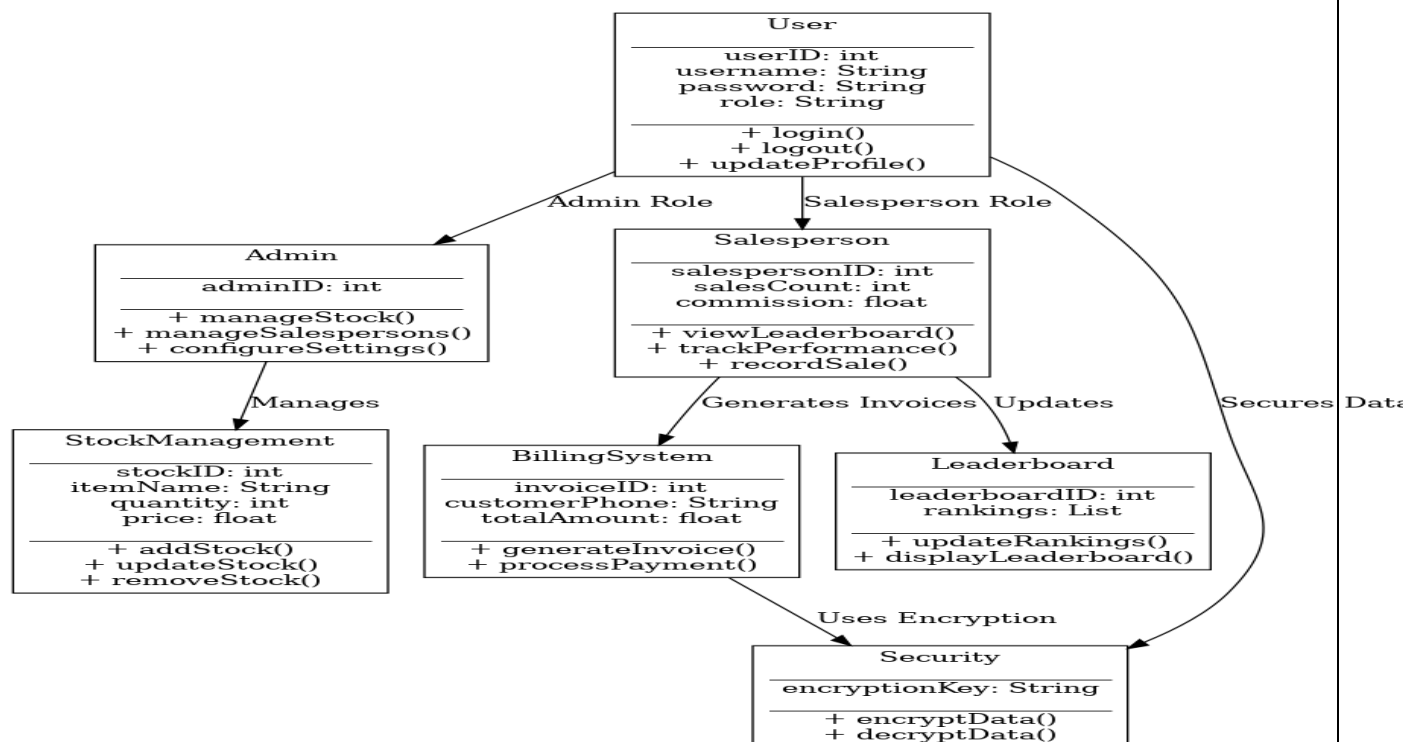
By defining workflows, the activity diagram ensures that the system operates efficiently and correctly.

6.1.6 Class Diagram

The Class Diagram represents system entities and their relationships. The key classes in SalesXP include:

1. User Class – Defines attributes like User ID, Name, Role, and Login Credentials.
2. Sales Class – Tracks transactions with attributes such as Sales ID, Product ID, Amount, and Timestamp.
3. Commission Class – Stores commission details, including percentage and payout status.
4. Leaderboard Class – Maintains rankings based on sales performance.

The class diagram ensures a structured database and clear object-oriented relationships.



6.1.7 Deployment Diagram

The Deployment Diagram showcases the physical architecture of SalesXP, including servers, databases, and cloud environments. Key deployment components include:

1. Application Server: Hosts the Spring Boot backend.
2. Database Server: Stores sales transactions³⁵, commission records, and reports.

3. Cloud Infrastructure: Ensures scalability and remote accessibility.
4. POS System Integration: Connects barcode scanners for real-time transaction logging.

This diagram helps in understanding hardware and network configurations.

6.1.8 State Chart Diagram

The State Chart Diagram outlines the different states that a salesperson or manager experiences within SalesXP. Examples include:

1. Login State: User enters credentials and gains access.
2. Sales Processing State: Sales transaction is recorded and validated.
3. Commission Approval State: Sales data is reviewed and commission payouts are processed.
4. Leaderboard Update State: Rankings are adjusted based on sales performance.

CHAPTER 7: TECHNOLOGY DESCRIPTION & IMPLEMENTATION

7.1 Introduction

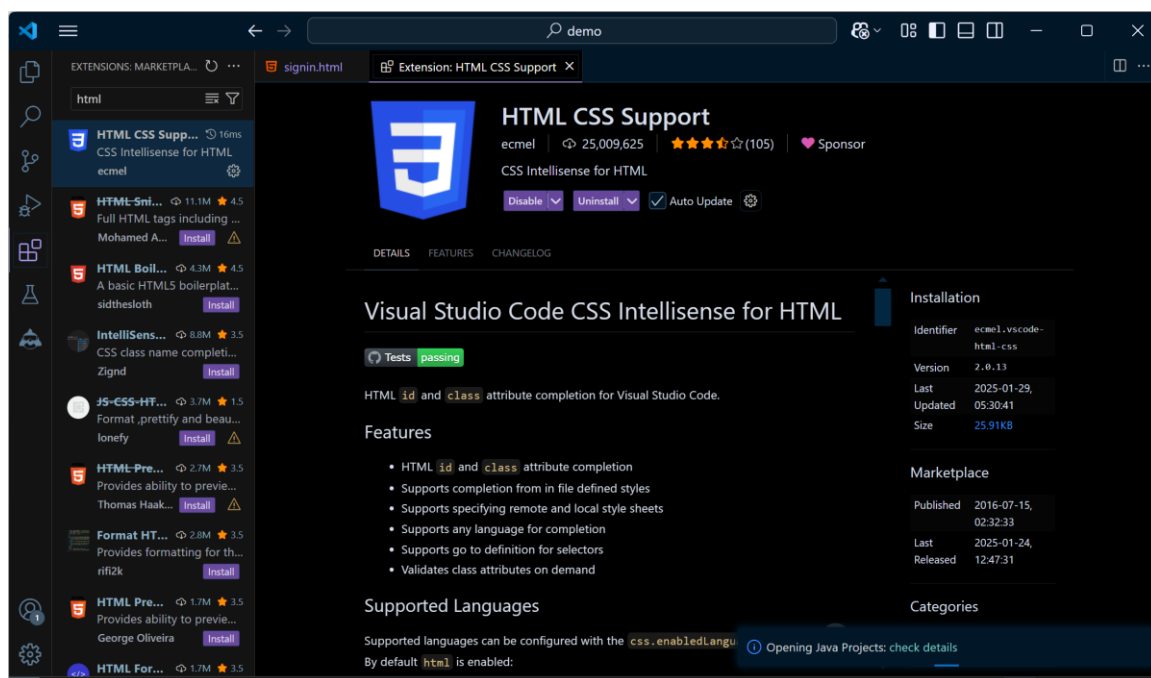
Technology selection and implementation play a crucial role in the development of any software system. SalesXP is a web-based platform designed to track sales commissions and enhance salesperson performance through structured incentives. This chapter discusses the technologies used in the frontend, backend, database, APIs, and deployment.

7.2 Frontend Technologies

The frontend of SalesXP is designed for a seamless user experience, ensuring that salespersons and administrators can interact efficiently with the system.

7.2.1 HTML (HyperText Markup Language)

- The structure of the web application is built using HTML5.
- It ensures a semantic layout, making the site accessible and SEO-friendly.



7.2.2 CSS (Cascading Style Sheets)

- Used for styling and improving the visual appeal.
- Implemented Bootstrap for responsive design, ensuring proper display on desktops,

tablets, and mobile devices.

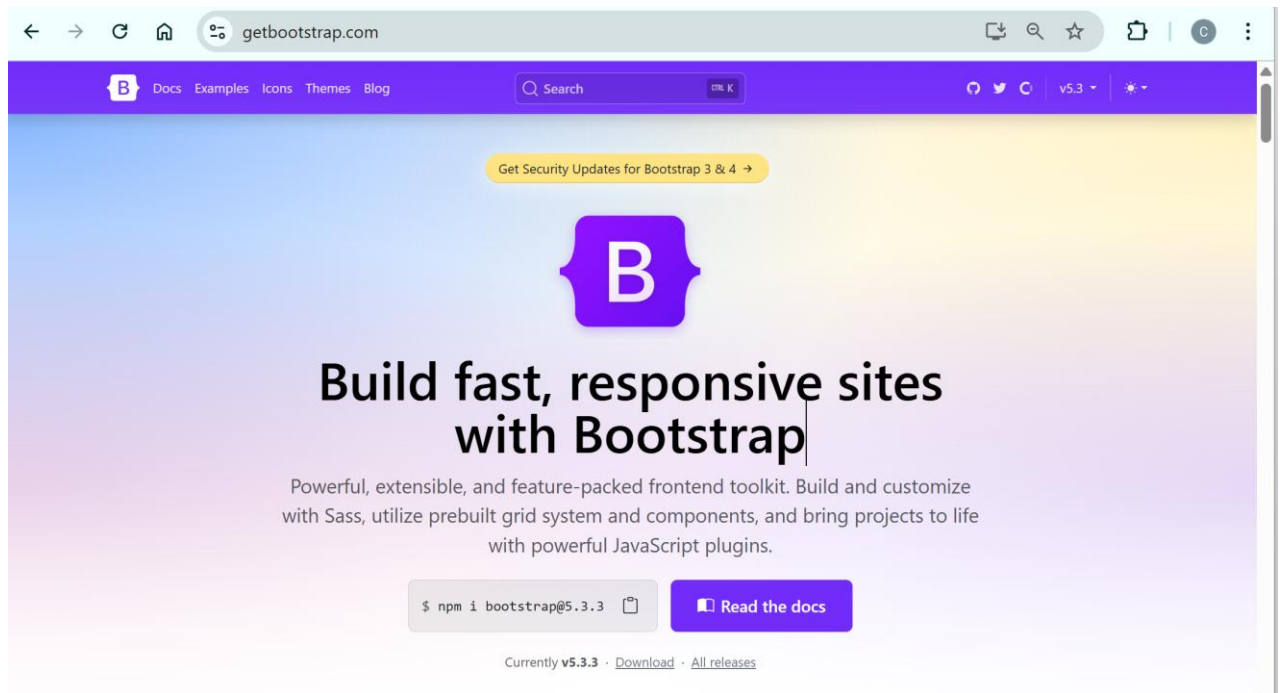
7.2.3 JavaScript

- Dynamic behavior is added to the web pages using JavaScript.
- Enables real-time updates like medicine search results and availability status without reloading the page.



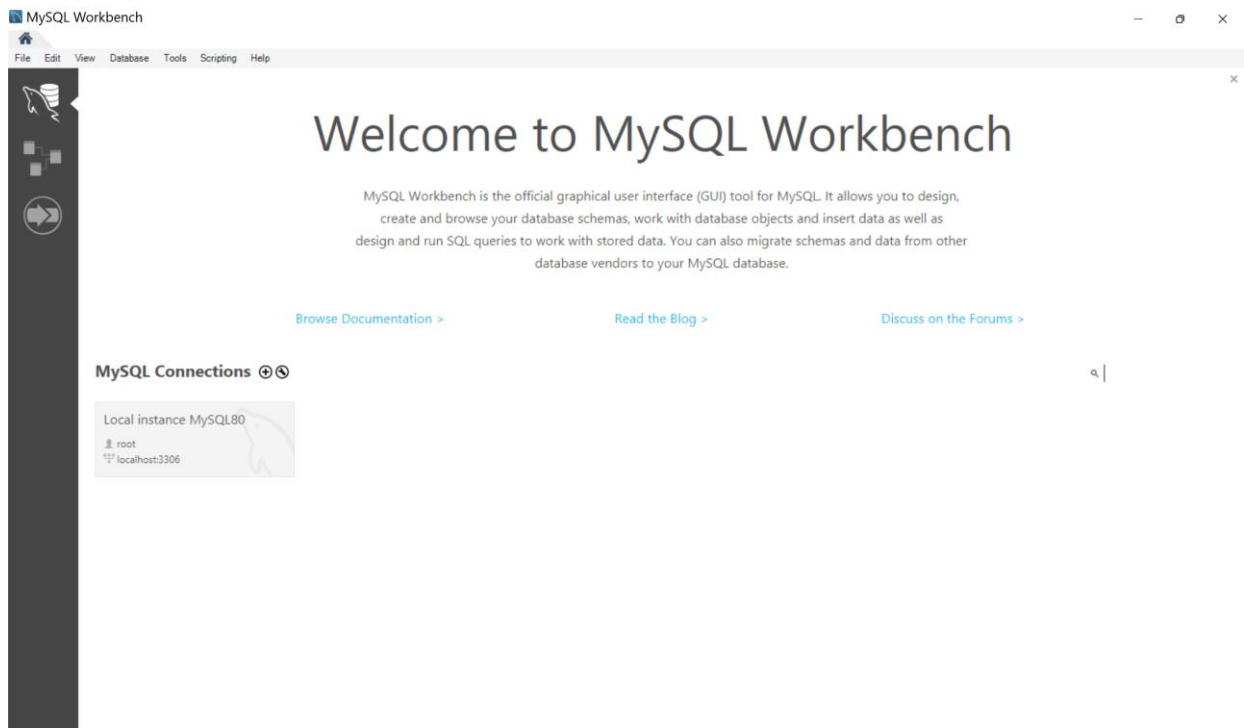
7.2.4 Bootstrap

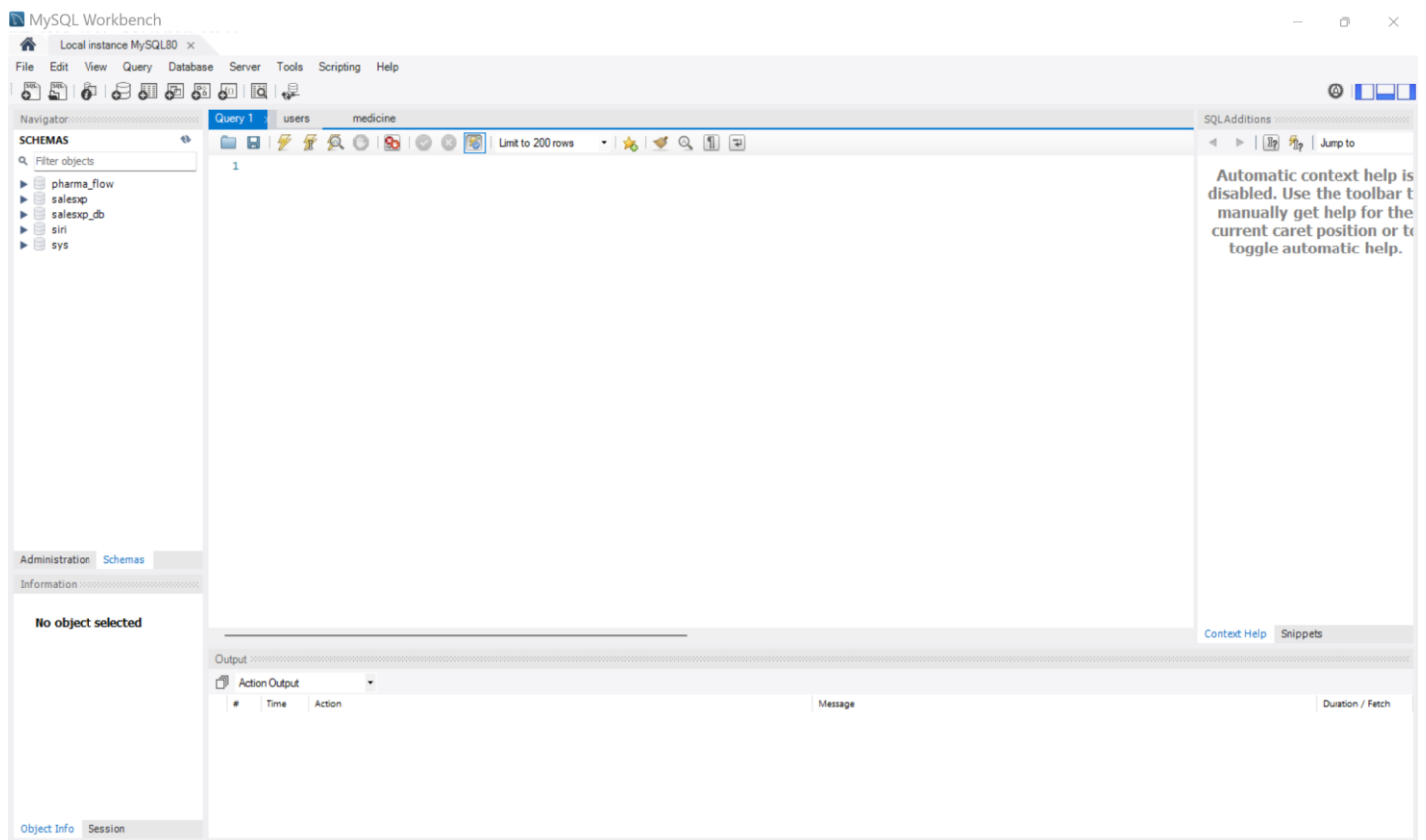
- Bootstrap simplifies UI development with predefined components.
- Provides a mobile-first design approach, ensuring ease of access for users.



7.3 Backend Technologies

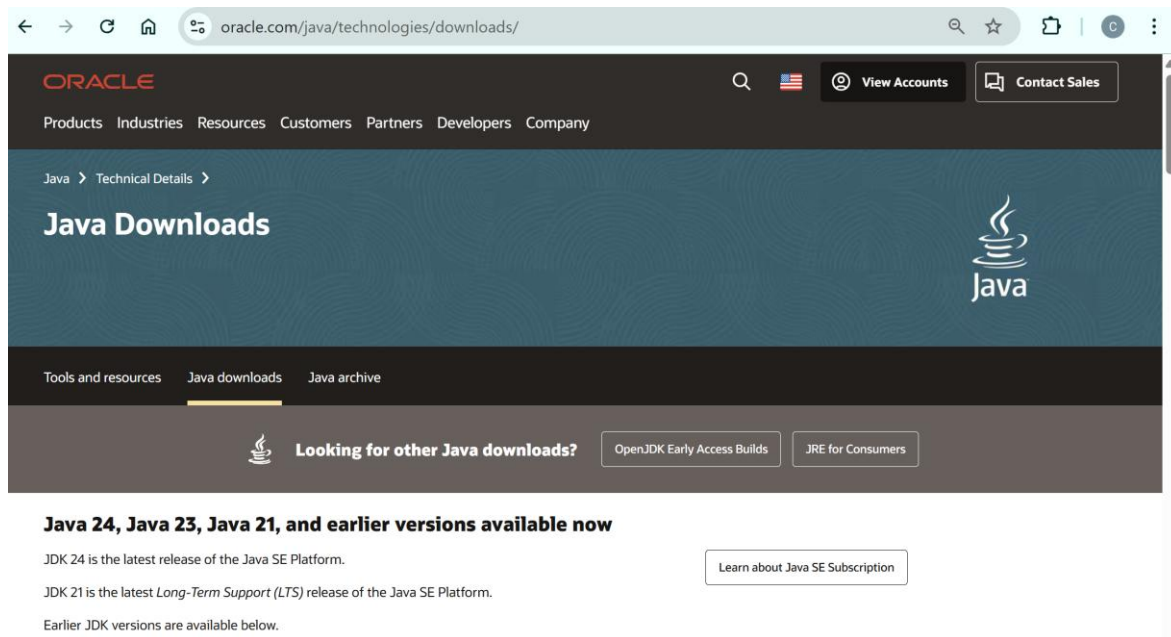
The backend is responsible for processing user requests, interacting with the database, and providing results efficiently.





7.3.1 Core Java

- The main programming language used for backend development.
- Handles business logic, authentication, and data processing.



```
C:\Users\Chaithanya>java --version
java 23.0.1 2024-10-15
Java(TM) SE Runtime Environment (build 23.0.1+11-39)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.1+11-39, mixed mode, sharing)

C:\Users\Chaithanya>
```

7.3.2 Spring Boot

- A lightweight and scalable framework for developing RESTful APIs.
- Provides features like dependency injection, security, and exception handling.
- Ensures seamless communication between the frontend and database.

Downloading Apache Maven 3.9.9

Apache Maven 3.9.9 is the latest release: it is the recommended version for all users.

System Requirements

Java Development Kit (JDK)	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains .
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) and Windows batch files.

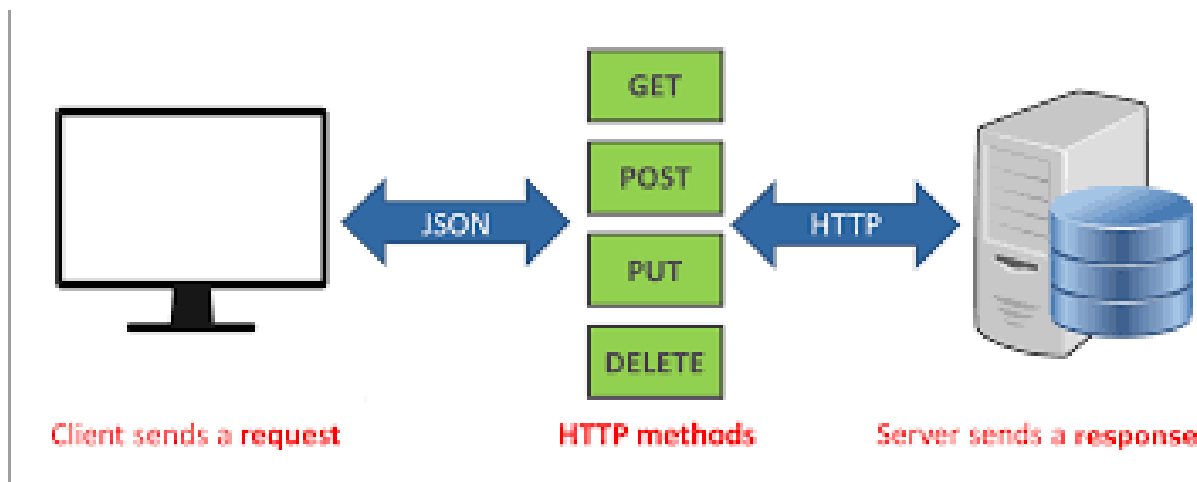
Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

7.3.3 REST APIs

- Enables communication between frontend, backend, and third-party services.
- Follows JSON-based data exchange for efficiency.



7.4 Database Management

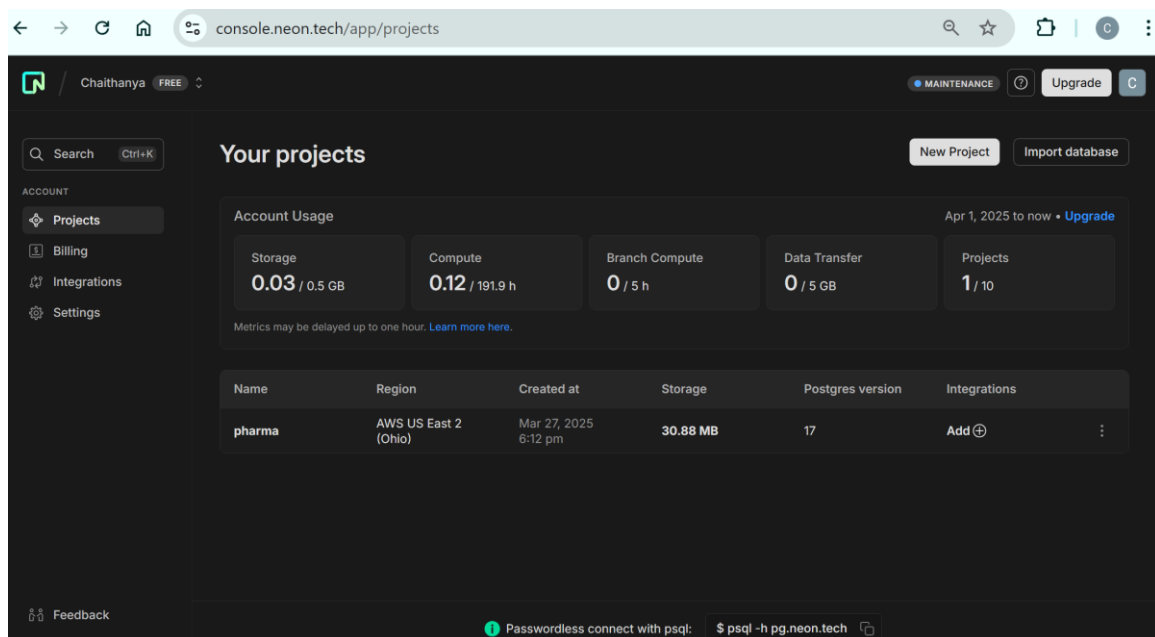
The database is essential for storing and managing application data efficiently.

7.4.1 MySQL

- A relational database management system (RDBMS) used for structured data storage.
- Stores user details, medicine inventory, pharmacy locations, and transaction records.

7.4.2 Neon DB

- A cloud-based database service used for scalability and reliability.
- Ensures fast data retrieval and backup management.



7.4.3 TablePlus

- A database management tool used for query execution, debugging, and performance tuning.
- Helps developers efficiently manage database schema and records.

id	address	branch	date_of_birth	date_of_joining	email	employee_id	full_name	gender	phone_number	role	password
13	New york	East Branch	2001-02-07	2025-06-05	mark@gmail.com	1	mark	Male	7896543244	Admin	Mark@2025
14	america	Main Branch	1989-12-10	2024-05-11	cap@gmail.com	2	steve	Male	0987654321	Admin	Avengers@2025
15	AP	Main Branch	1997-04-08	2023-06-07	ram@gmail.com	42	ram	Male	9867094233	Admin	Ram@2025

7.5 Deployment & Hosting

Deployment is crucial for making the application accessible to users.

salesxptwo-1 Docker Free Upgrade your instance →

chayupakula / salesxptwo ~1 main
<https://salesxptwo-1.onrender.com>

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now](#)

April 2, 2025 at 6:08 PM Live

All logs Search Live tail GMT+5:30

```

Apr 2 06:19:20 PM ● --> Detected a new open port HTTP-6000
Apr 2 06:20:21 PM ● --> Detected a new open port HTTP-6000
Apr 2 06:20:26 PM ● --> Detected service running on port 6000
Apr 2 06:20:27 PM ● --> Docs on specifying a port: https://render.com/docs/web-services#port-binding
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.467Z INFO 1 --- [demo] [ionShutdownHook] o.s.b.w.o.tomcat.GracefulShutdown : Commencing graceful shutdown. Waiting for active r
equests to complete
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.476Z INFO 1 --- [demo] [tomcat-shutdown] o.s.b.w.o.tomcat.GracefulShutdown : Graceful shutdown complete
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.476Z INFO 1 --- [demo] [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence u
nit 'default'
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.483Z INFO 1 --- [demo] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.676Z INFO 1 --- [demo] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.467Z INFO 1 --- [demo] [ionShutdownHook] o.s.b.w.o.tomcat.GracefulShutdown : Commencing graceful shutdown. Waiting for active r
equests to complete
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.476Z INFO 1 --- [demo] [tomcat-shutdown] o.s.b.w.o.tomcat.GracefulShutdown : Graceful shutdown complete
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.476Z INFO 1 --- [demo] [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence u
nit 'default'
Apr 2 06:34:35 PM ● 2025-04-02T13:04:35.483Z INFO 1 --- [demo] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
  
```

You can also use the [Render CLI](#) to explore logs in your command line.
 Looking for more logs? Try [Log Streams](#).

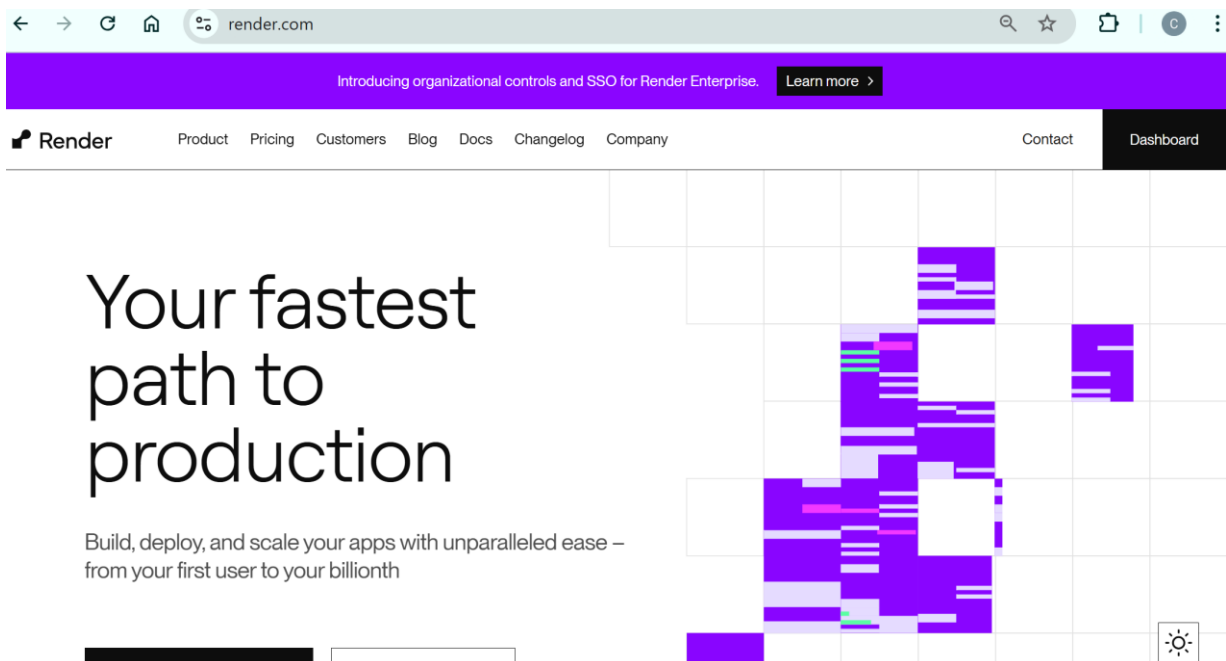
7.6.1 Docker

- Used for containerization, ensuring the application runs consistently across different environments.
- Simplifies scalability and deployment process.



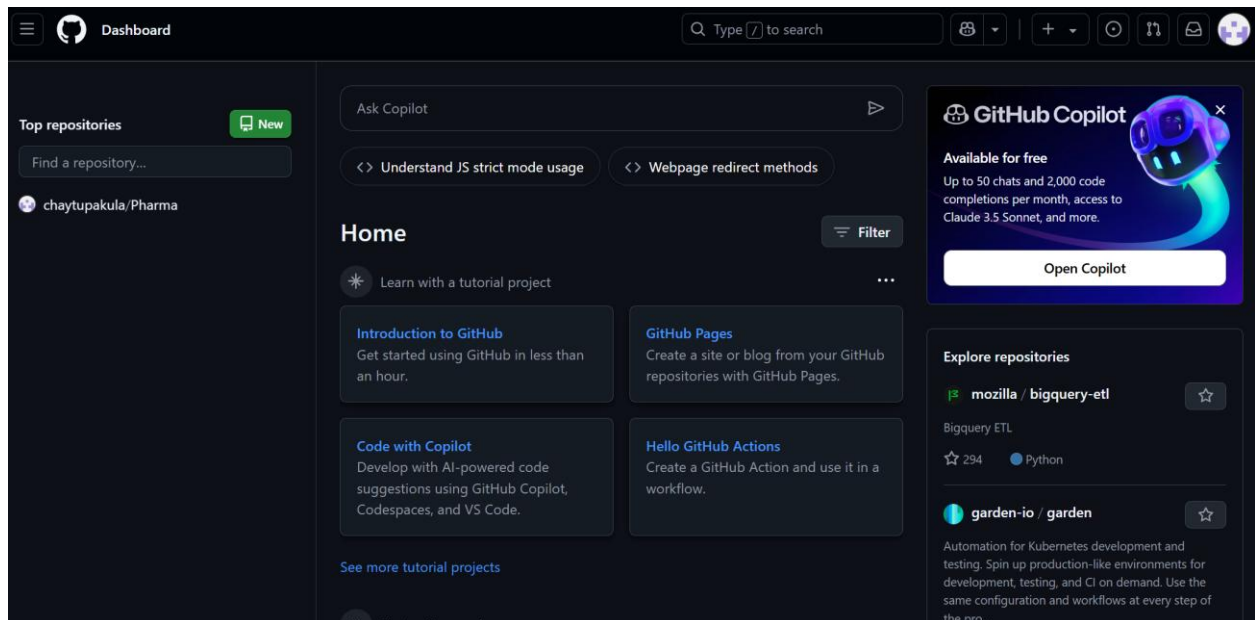
7.6.2 OnRender

- A cloud hosting platform used for deploying the application securely.
- Provides auto-scaling, database hosting, and load balancing.



7.6.3 GitHub

- Version control is managed through GitHub.
- Helps in collaboration, bug tracking, and feature updates.



7.7 Implementation Process

The implementation follows a structured approach.

7.7.1 Development Lifecycle

1. Requirement Analysis – Understanding user needs.
2. System Design – Creating database schema, UI mockups.
3. Implementation – Developing individual modules.
4. Testing & Debugging– Ensuring bug-free deployment.
5. Deployment – Making the system available to users.
6. Maintenance & Updates – Continuous improvements and fixes.



7.7.2 Security Measures

- Regular security patches to prevent cyber threats.

CHAPTER 8

SOURCE CODE

DemoApplication.java:

```
package com.example.demo;

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

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

Signup.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Registration</title>
    <style>
        :root {
            --primary-color: #4a86e8;
            --primary-hover: #3a76d8;
            --success-bg: #d4edda;
            --success-color: #155724;
            --error-bg: #f8d7da;
            --error-color: #721c24;
            --border-color: #e0e0e0;
            --shadow: 0 2px 5px rgba(0,0,0,0.1);
        }

        * {
            box-sizing: border-box;
            margin: 0;
            padding: 0;
        }
```

```
body {  
  font-family: 'Segoe UI', Roboto, Arial, sans-serif;  
  line-height: 1.6;  
  color: #333;  
  background-color: #f8f9fa;  
  margin: 0;  
  padding: 20px;  
}
```

```
.container {  
  max-width: 800px;  
  margin: 0 auto;  
  background-color: white;  
  border-radius: 8px;  
  box-shadow: var(--shadow);  
  padding: 30px;  
}
```

```
h2 {  
  color: #2c3e50;  
  margin-bottom: 20px;  
  text-align: center;  
  font-size: 28px;  
  border-bottom: 2px solid var(--primary-color);  
  padding-bottom: 10px;  
}
```

```
.progress-bar {  
  margin-bottom: 30px;  
  display: flex;  
  justify-content: space-between;  
  position: relative;  
}
```

```
.progress-bar::before {  
  content: "";  
  position: absolute;  
  top: 15px;  
  left: 0;  
  width: 100%;  
  height: 3px;
```

```
    background-color: var(--border-color);
    z-index: 1;
}
```

```
.progress-step {
    position: relative;
    z-index: 2;
    text-align: center;
    width: 33%;
}
```

```
.step-circle {
    width: 30px;
    height: 30px;
    border-radius: 50%;
    background-color: white;
    border: 3px solid var(--border-color);
    margin: 0 auto 5px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: bold;
    transition: all 0.3s ease;
}
```

```
.progress-step.active .step-circle {
    border-color: var(--primary-color);
    background-color: var(--primary-color);
    color: white;
}
```

```
.progress-step.complete .step-circle {
    border-color: #4CAF50;
    background-color: #4CAF50;
    color: white;
}
```

```
.step-label {
    font-size: 14px;
    color: #666;
    font-weight: 500;
}
```

```

.progress-step.active .step-label {
  color: var(--primary-color);
  font-weight: 600;
}

.progress-step.complete .step-label {
  color: #4CAF50;
  font-weight: 600;
}

.form-section {
  display: none;
  animation: fadeIn 0.5s;
}

.form-section.active {
  display: block;
}

@keyframes fadeIn {
  from { opacity: 0; transform: translateY(10px); }
  to { opacity: 1; transform: translateY(0); }
}

fieldset {
  border: 1px solid var(--border-color);
  border-radius: 6px;
  padding: 20px;
  margin-bottom: 25px;
  box-shadow: 0 1px 3px rgba(0,0,0,0.05);
}

legend {
  font-weight: 600;
  padding: 0 10px;
  color: var(--primary-color);
  font-size: 18px;
}

.form-group {
  margin-bottom: 15px;
}

```

```

}

label {
  display: block;
  margin-bottom: 5px;
  font-weight: 500;
  color: #555;
}

input, select, textarea {
  width: 100%;
  padding: 10px 12px;
  border: 1px solid var(--border-color);
  border-radius: 4px;
  font-size: 15px;
  transition: border-color 0.3s;
}

input:focus, select:focus, textarea:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(74, 134, 232, 0.2);
}

.date-group {
  display: flex;
  gap: 10px;
}

.date-group select {
  flex: 1;
}

.hint {
  font-size: 12px;
  color: #777;
  margin-top: 4px;
}

.error {
  color: var(--error-color);
  font-size: 13px;
}

```

```

    margin-top: 4px;
    display: none;
}

.error.visible {
    display: block;
}

.notification {
    padding: 15px;
    border-radius: 6px;
    margin: 25px 0;
    display: flex;
    align-items: center;
    animation: slideDown 0.4s ease-out;
}

@keyframes slideDown {
    from { opacity: 0; transform: translateY(-20px); }
    to { opacity: 1; transform: translateY(0); }
}

.notification.success {
    background-color: var(--success-bg);
    color: var(--success-color);
    border-left: 4px solid var(--success-color);
}

.notification.error {
    background-color: var(--error-bg);
    color: var(--error-color);
    border-left: 4px solid var(--error-color);
}

.notification-icon {
    margin-right: 10px;
    font-size: 20px;
}

.btn-container {
    display: flex;
    justify-content: space-between;

```

```

    margin-top: 30px;
}

.btn {
    padding: 12px 20px;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    font-weight: 500;
    cursor: pointer;
    transition: all 0.2s;
}

.btn-primary {
    background-color: var(--primary-color);
    color: white;
}

.btn-primary:hover {
    background-color: var(--primary-hover);
}

.btn-secondary {
    background-color: #e2e6ea;
    color: #495057;
}

.btn-secondary:hover {
    background-color: #dae0e5;
}

.btn-primary:disabled {
    background-color: #b0c7e9;
    cursor: not-allowed;
}

/* Loading spinner */
.spinner {
    display: inline-block;
    width: 20px;
    height: 20px;
    border: 3px solid rgba(255,255,255,.3);

```



```
border-radius: 50%;
border-top-color: white;
animation: spin 1s ease-in-out infinite;
margin-right: 8px;
vertical-align: middle;
display: none;
}
```

```
@keyframes spin {
  to { transform: rotate(360deg); }
}
```

```
.login-link {
  text-align: center;
  margin-top: 20px;
  font-size: 15px;
}
```

```
.login-link a {
  color: var(--primary-color);
  text-decoration: none;
  font-weight: 500;
}
```

```
.login-link a: hover {
  text-decoration: underline;
}
```

```
/* Responsive adjustments */
@media (max-width: 600px) {
  .container {
    padding: 15px;
  }
```

```
  fieldset {
    padding: 15px;
  }
```

```
.date-group {
  flex-direction: column;
  gap: 15px;
}
```

```

        .btn {
            padding: 10px 16px;
        }
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Employee Registration Portal</h2>

        <div class="progress-bar">
            <div class="progress-step active" id="step1">
                <div class="step-circle">1</div>
                <div class="step-label">Personal Information</div>
            </div>
            <div class="progress-step" id="step2">
                <div class="step-circle">2</div>
                <div class="step-label">Account Setup</div>
            </div>
            <div class="progress-step" id="step3">
                <div class="step-circle">3</div>
                <div class="step-label">Employment Details</div>
            </div>
        </div>

        <div id="notification" style="display: none;"></div>

        <form id="signupForm">
            <!-- Section 1: Personal Information -->
            <div class="form-section active" id="section1">
                <fieldset>
                    <legend>Personal Information</legend>

                    <div class="form-group">
                        <label for="fullName">Full Name</label>
                        <input type="text" id="fullName" placeholder="Enter your full name">
                        <div id="fullNameError" class="error">Please enter your full name</div>
                    </div>

                    <div class="form-group">
                        <label for="dob">Date of Birth</label>

```

```

<div class="date-group">
  <select id="dobDay">
    <option value="">Day</option>
    <script>
      for(let i=1; i<=31; i++) {
        document.write(`<option value="${i}">${i}</option>`);
      }
    </script>
  </select>
  <select id="dobMonth">
    <option value="">Month</option>
    <option value="1">January</option>
    <option value="2">February</option>
    <option value="3">March</option>
    <option value="4">April</option>
    <option value="5">May</option>
    <option value="6">June</option>
    <option value="7">July</option>
    <option value="8">August</option>
    <option value="9">September</option>
    <option value="10">October</option>
    <option value="11">November</option>
    <option value="12">December</option>
  </select>
  <select id="dobYear">
    <option value="">Year</option>
    <script>
      for(let i=2006; i>=1950; i--) {
        document.write(`<option value="${i}">${i}</option>`);
      }
    </script>
  </select>
</div>
<div id="dobError" class="error">Please select your complete date of birth</div>
</div>

<div class="form-group">
  <label for="gender">Gender</label>
  <select id="gender">
    <option value="">Select gender</option>
    <option value="Male">Male</option>
    <option value="Female">Female</option>

```

```

        <option value="Other">Other</option>
        <option value="Prefer not to say">Prefer not to say</option>
    </select>
    <div id="genderError" class="error">Please select your gender</div>
</div>

<div class="form-group">
    <label for="phoneNumber">Phone Number</label>
    <input type="tel" id="phoneNumber" placeholder="e.g. +1 (234) 567-8901">
    <div id="phoneNumberError" class="error">Please enter a valid phone
number</div>
</div>

<div class="form-group">
    <label for="email">Email Address</label>
    <input type="email" id="email" placeholder="your.email@example.com">
    <div id="emailError" class="error">Please enter a valid email address</div>
</div>

<div class="form-group">
    <label for="address">Address</label>
    <textarea id="address" rows="3" placeholder="Enter your full
address"></textarea>
    <div id="addressError" class="error">Please enter your address</div>
</div>
</fieldset>

<div class="btn-container">
    <div></div> <!-- Empty div for flex spacing -->
    <button type="button" class="btn btn-primary" id="nextToSection2">Continue to
Account Setup</button>
</div>
</div>

<!-- Section 2: Account Security -->
<div class="form-section" id="section2">
    <fieldset>
        <legend>Account Security</legend>

        <div class="form-group">
            <label for="password">Password</label>
            <input type="password" id="password" placeholder="Create a password">

```

```

        <div class="hint">Password must be at least 8 characters with letters and
numbers</div>
        <div id="passwordError" class="error">Password must be at least 8
characters</div>
    </div>

    <div class="form-group">
        <label for="confirmPassword">Confirm Password</label>
        <input type="password" id="confirmPassword" placeholder="Confirm your
password">
        <div id="confirmPasswordError" class="error">Passwords do not match</div>
    </div>

    <div id="passwordStrength" class="password-strength" style="margin-top: 15px;">
        <div style="margin-bottom: 5px; font-size: 14px; color: #555;">Password
Strength:</div>
        <div style="display: flex; gap: 5px; margin-bottom: 10px;">
            <div id="strength1" style="height: 5px; flex: 1; background-color: #e0e0e0;
border-radius: 2px;"></div>
            <div id="strength2" style="height: 5px; flex: 1; background-color: #e0e0e0;
border-radius: 2px;"></div>
            <div id="strength3" style="height: 5px; flex: 1; background-color: #e0e0e0;
border-radius: 2px;"></div>
            <div id="strength4" style="height: 5px; flex: 1; background-color: #e0e0e0;
border-radius: 2px;"></div>
        </div>
        <div id="strengthText" style="font-size: 13px; color: #666;">Not set</div>
    </div>
</fieldset>

<div class="btn-container">
    <button type="button" class="btn btn-secondary"
id="backToSection1">Back</button>
    <button type="button" class="btn btn-primary" id="nextToSection3">Continue to
Employment Details</button>
</div>
</div>

<!-- Section 3: Employment Information -->
<div class="form-section" id="section3">
    <fieldset>
        <legend>Employment Information</legend>

```

```

<div class="form-group">
  <label for="employeeId">Employee ID</label>
  <input type="number" id="employeeId" placeholder="Enter your employee ID">
  <div id="employeeIdError" class="error">Please enter a valid employee ID</div>
</div>

```

```

<div class="form-group">
  <label for="doj">Date of Joining</label>
  <div class="date-group">
    <select id="dojDay">
      <option value="">Day</option>
      <script>
        for(let i=1; i<=31; i++) {
          document.write(`<option value="${i}">${i}</option>`);
        }
      </script>
    </select>
    <select id="dojMonth">
      <option value="">Month</option>
      <option value="1">January</option>
      <option value="2">February</option>
      <option value="3">March</option>
      <option value="4">April</option>
      <option value="5">May</option>
      <option value="6">June</option>
      <option value="7">July</option>
      <option value="8">August</option>
      <option value="9">September</option>
      <option value="10">October</option>
      <option value="11">November</option>
      <option value="12">December</option>
    </select>
    <select id="dojYear">
      <option value="">Year</option>
      <script>
        for(let i=2025; i>=2000; i--) {
          document.write(`<option value="${i}">${i}</option>`);
        }
      </script>
    </select>
  </div>
</div>

```

```

    <div id="dojError" class="error">Please select your complete date of
joining</div>
  </div>

```

```

    <div class="form-group">
      <label for="branch">Branch</label>
      <select id="branch">
        <option value="">Select branch</option>
        <option value="Main Branch">Main Branch</option>
        <option value="North Branch">North Branch</option>
        <option value="South Branch">South Branch</option>
        <option value="East Branch">East Branch</option>
        <option value="West Branch">West Branch</option>
      </select>
      <div id="branchError" class="error">Please select your branch</div>
    </div>
  </fieldset>

```

```

    <div class="btn-container">
      <button type="button" class="btn btn-secondary"
id="backToSection2">Back</button>
      <button type="button" class="btn btn-primary" id="submitButton">
        <span id="spinner" class="spinner"></span>
        <span id="submitText">Complete Registration</span>
      </button>
    </div>
  </div>
</form>

```

```

<div class="login-link">
  Already registered? <a href="login.html">Login here</a>
</div>
</div>

```

```

<script>
  // DOM elements
  const sections = ['section1', 'section2', 'section3'];
  const steps = ['step1', 'step2', 'step3'];

  // Navigation buttons
  document.getElementById('nextToSection2').addEventListener('click', () => {
    if (validateSection1()) {

```

```

        navigateToSection(2);
    }
});

document.getElementById('backToSection1').addEventListener('click', () => {
    navigateToSection(1);
});

document.getElementById('nextToSection3').addEventListener('click', () => {
    if (validateSection2()) {
        navigateToSection(3);
    }
});

document.getElementById('backToSection2').addEventListener('click', () => {
    navigateToSection(2);
});

document.getElementById('submitButton').addEventListener('click', handleSubmit);

// Form field validation
const fullName = document.getElementById('fullName');
const dobDay = document.getElementById('dobDay');
const dobMonth = document.getElementById('dobMonth');
const dobYear = document.getElementById('dobYear');
const gender = document.getElementById('gender');
const phoneNumber = document.getElementById('phoneNumber');
const email = document.getElementById('email');
const address = document.getElementById('address');
const password = document.getElementById('password');
const confirmPassword = document.getElementById('confirmPassword');
const employeeId = document.getElementById('employeeId');
const dojDay = document.getElementById('dojDay');
const dojMonth = document.getElementById('dojMonth');
const dojYear = document.getElementById('dojYear');
const branch = document.getElementById('branch');

// Password strength meter
password.addEventListener('input', updatePasswordStrength);

function updatePasswordStrength() {
    const val = password.value;

```



```

const strength1 = document.getElementById('strength1');
const strength2 = document.getElementById('strength2');
const strength3 = document.getElementById('strength3');
const strength4 = document.getElementById('strength4');
const strengthText = document.getElementById('strengthText');

// Reset
[strength1, strength2, strength3, strength4].forEach(el => {
  el.style.backgroundColor = '#e0e0e0';
});

if (val.length === 0) {
  strengthText.textContent = 'Not set';
  strengthText.style.color = '#666';
  return;
}

// Check strength
let score = 0;
if (val.length >= 8) score++;
if (val.length >= 10) score++;
if (/[A-Z]/.test(val)) score++;
if (/[0-9]/.test(val)) score++;
if (/^[A-Za-z0-9]/.test(val)) score++;

// Normalize score to 0-4
score = Math.min(4, score);

const colors = ['#FF4136', '#FF851B', '#FFDC00', '#2ECC40'];
const texts = ['Weak', 'Fair', 'Good', 'Strong'];

for (let i = 0; i < score; i++) {
  document.getElementById(`strength${i+1}`).style.backgroundColor =
colors[Math.min(i, 3)];
}

strengthText.textContent = texts[Math.min(score-1, 3)];
strengthText.style.color = colors[Math.min(score-1, 3)];
}

// Password match validation
confirmPassword.addEventListener('input', validatePasswordMatch);

```

```

password.addEventListener('input', () => {
  if (confirmPassword.value) validatePasswordMatch();
});

function validatePasswordMatch() {
  const errorElem = document.getElementById('confirmPasswordError');

  if (password.value !== confirmPassword.value) {
    errorElem.classList.add('visible');
    return false;
  } else {
    errorElem.classList.remove('visible');
    return true;
  }
}

// Section validation functions
function validateSection1() {
  let isValid = true;

  // Validate Full Name
  if (!fullName.value.trim()) {
    document.getElementById('fullNameError').classList.add('visible');
    isValid = false;
  } else {
    document.getElementById('fullNameError').classList.remove('visible');
  }

  // Validate Date of Birth
  if (!dobDay.value || !dobMonth.value || !dobYear.value) {
    document.getElementById('dobError').classList.add('visible');
    isValid = false;
  } else {
    document.getElementById('dobError').classList.remove('visible');
  }

  // Validate Gender
  if (!gender.value) {
    document.getElementById('genderError').classList.add('visible');
    isValid = false;
  } else {
    document.getElementById('genderError').classList.remove('visible');
  }
}

```

```

    }

    // Validate Phone Number
    if (!phoneNumber.value.trim()) {
        document.getElementById('phoneNumberError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('phoneNumberError').classList.remove('visible');
    }

    // Validate Email
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!email.value.trim() || !emailRegex.test(email.value)) {
        document.getElementById('emailError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('emailError').classList.remove('visible');
    }

    // Validate Address
    if (!address.value.trim()) {
        document.getElementById('addressError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('addressError').classList.remove('visible');
    }

    return isValid;
}

function validateSection2() {
    let isValid = true;

    // Validate Password
    if (!password.value || password.value.length < 8) {
        document.getElementById('passwordError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('passwordError').classList.remove('visible');
    }

    // Validate Password Confirmation

```

```

    if (!validatePasswordMatch()) {
        isValid = false;
    }

    return isValid;
}

function validateSection3() {
    let isValid = true;

    // Validate Employee ID
    if (!employeeId.value.trim() || isNaN(parseInt(employeeId.value))) {
        document.getElementById('employeeIdError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('employeeIdError').classList.remove('visible');
    }

    // Validate Date of Joining
    if (!dojDay.value || !dojMonth.value || !dojYear.value) {
        document.getElementById('dojError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('dojError').classList.remove('visible');
    }

    // Validate Branch
    if (!branch.value) {
        document.getElementById('branchError').classList.add('visible');
        isValid = false;
    } else {
        document.getElementById('branchError').classList.remove('visible');
    }

    return isValid;
}

// Navigation function
function navigateToSection(sectionNumber) {
    // Hide all sections and update steps
    sections.forEach((section, index) => {
        document.getElementById(section).classList.remove('active');
    });
}

```

```

    document.getElementById(steps[index]).classList.remove('active', 'complete');
  });

  // Show target section
  document.getElementById(`section${sectionNumber}`).classList.add('active');
  document.getElementById(`step${sectionNumber}`).classList.add('active');

  // Mark previous steps as complete
  for (let i = 1; i < sectionNumber; i++) {
    document.getElementById(`step${i}`).classList.add('complete');
  }
}

// Submit form
async function handleSubmit() {
  if (!validateSection3()) {
    return;
  }

  // Show loading state
  const submitButton = document.getElementById('submitButton');
  const spinner = document.getElementById('spinner');
  const submitText = document.getElementById('submitText');

  submitButton.disabled = true;
  spinner.style.display = 'inline-block';
  submitText.textContent = 'Processing...';

  // Prepare data object
  const data = {
    fullName: fullName.value.trim(),
    dobDay: parseInt(dobDay.value),
    dobMonth: parseInt(dobMonth.value),
    dobYear: parseInt(dobYear.value),
    gender: gender.value,
    address: address.value.trim(),
    phoneNumber: phoneNumber.value.trim(),
    email: email.value.trim(),
    password: password.value,
    employeeId: parseInt(employeeId.value),
    dojDay: parseInt(dojDay.value),
    dojMonth: parseInt(dojMonth.value),
  };

```

```

    dojYear: parseInt(dojYear.value),
    branch: branch.value
  };

  try {
    const response = await fetch('https://sales-fi3s.onrender.com/api/signup', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(data)
    });

    const result = await response.json();

    if (response.ok) {
      showNotification(`Success! ${result.message} (Employee ID: ${result.employeeId ||
data.employeeId})`, 'success');
      document.getElementById('signupForm').reset();
      navigateToSection(1);
    } else {
      throw new Error(result.message || `HTTP error! status: ${response.status}`);
    }
  } catch (error) {
    showNotification(`Error: ${error.message}`, 'error');
  } finally {
    // Reset button state
    submitButton.disabled = false;
    spinner.style.display = 'none';
    submitText.textContent = 'Complete Registration';
  }
}

// Show notification
function showNotification(message, type) {
  const notificationEl = document.getElementById('notification');

  // Create notification content
  const icon = type === 'success' ? '✓' : '✗';
  notificationEl.innerHTML = `
    <div class="notification ${type}">
      <div class="notification-icon">${icon}</div>

```

```

        <div>${message}</div>
    </div>
`;

notificationEl.style.display = 'block';

// Scroll to notification
notificationEl.scrollIntoView({ behavior: 'smooth', block: 'start' });

// Auto-hide after 5 seconds
if (type === 'success') {
    setTimeout(() => {
        notificationEl.style.display = 'none';
    }, 5000);
}
}
</script>
</body>
</html>

```

Sign.js:

```

let currentSection = 1;
const sections = document.querySelectorAll('.form-section');
const roleField = document.getElementById('role');
roleField.value = 'Admin'; // Set default role

function showSection(sectionNumber) {
    sections.forEach(section => section.classList.remove('active'));
    document.getElementById(`section${sectionNumber}`).classList.add('active');
}

function validateAndNext(nextSection) {
    const currentForm = document.getElementById(`section${currentSection}`);
    const inputs = currentForm.querySelectorAll('input, select');
    let isValid = true;

    inputs.forEach(input => {
        if (!input.checkValidity()) {
            input.reportValidity();
            isValid = false;
        }
    })
}

```

```

});

if (isValid) {
  currentSection = nextSection;
  showSection(currentSection);

  if (currentSection === 4) {
    document.getElementById('termsContainer').style.display = 'block';
  }
}
}

function validatePassword() {
  const password = document.getElementById('password');
  const confirmPassword = document.getElementById('confirmPassword');
  const signupButton = document.getElementById('signupButton');

  if (password.value !== confirmPassword.value) {
    confirmPassword.setCustomValidity("Passwords don't match");
    signupButton.disabled = true;
  } else {
    confirmPassword.setCustomValidity("");
    signupButton.disabled = !document.getElementById('terms').checked;
  }
}

function toggleSignupButton() {
  const terms = document.getElementById('terms');
  const signupButton = document.getElementById('signupButton');
  signupButton.disabled = !terms.checked;
}

async function validateForm() {
  const formData = {
    fullName: document.getElementById('fullName').value,
    dob: document.getElementById('dob').value,
    gender: document.getElementById('gender').value,
    address: document.getElementById('address').value,
    phoneNumber: document.getElementById('phone').value,
    email: document.getElementById('email').value,
    employeeId: document.getElementById('empId').value,
    doj: document.getElementById('doj').value,

```



```

    branch: document.getElementById('branch').value,
    password: document.getElementById('password').value
  };

  // Convert dates to day/month/year format
  const [dobYear, dobMonth, dobDay] = formData.dob.split('-');
  const [dojYear, dojMonth, dojDay] = formData.doj.split('-');

  const requestBody = {
    fullName: formData.fullName,
    dobDay: parseInt(dobDay),
    dobMonth: parseInt(dobMonth),
    dobYear: parseInt(dobYear),
    gender: formData.gender,
    address: formData.address,
    phoneNumber: formData.phoneNumber,
    email: formData.email,
    employeeId: parseInt(formData.employeeId),
    dojDay: parseInt(dojDay),
    dojMonth: parseInt(dojMonth),
    dojYear: parseInt(dojYear),
    branch: formData.branch
  };

  try {
    const response = await fetch('http://localhost:6060/api/signup', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(requestBody)
    });

    const result = await response.json();

    if (response.ok) {
      alert(`Signup successful! Employee ID: ${result.employeeId}`);
      window.location.href = '/login'; // Redirect to login page
    } else {
      alert(`Error: ${result.message}`);
    }
  } catch (error) {

```

```
    alert(`Error: ${error.message}`);  
  }  
}
```

Docker:

Stage 1: Build

FROM maven:3.9-eclipse-temurin-21 AS build

WORKDIR /app

COPY pom.xml .

RUN mvn dependency:go-offline

COPY src ./src

RUN mvn clean package -DskipTests

Stage 2: Runtime

FROM eclipse-temurin:21.0.2_13-jre-jammy

WORKDIR /app

COPY --from=build /app/target/*.jar app.jar

EXPOSE 9090

ENTRYPOINT ["java", "-jar", "app.jar"]

Application.properties:

spring.application.name=demo

server.port=6060

Database Configuration

spring.datasource.url=jdbc:postgresql://ep-patient-dust-a5qpfwx4-pooler.us-east-2.aws.neon.tech/neondb?sslmode=require

spring.datasource.username=neondb_owner

spring.datasource.password=npg_9Vy2cYWHLfrI

spring.datasource.driver-class-name=org.postgresql.Driver

JPA/Hibernate

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect

spring.jpa.hibernate.ddl-auto=update

CHAPTER 9: SYSTEM TESTING

9.1 Introduction to System Testing

System testing is a critical phase in the Software Development Life Cycle (SDLC) where the entire SalesXP application is tested as a whole to verify its functionality, performance, security, and usability. The primary goal is to ensure that the system meets the defined requirements and works as expected under different scenarios. This phase involves identifying defects, ensuring data integrity, and validating user interactions before deployment. Testing is crucial for detecting bugs, improving performance, and ensuring system stability. It helps in evaluating both functional and non-functional aspects such as usability, security, and performance. Testing ensures that the commission tracking system calculates payouts correctly, prevents fraudulent claims, and updates rankings in real-time. Various testing techniques such as unit testing, integration testing, system testing, and user acceptance testing (UAT) are performed to ensure reliability. System testing also validates that SalesXP integrates seamlessly with external systems like POS machines, databases, and cloud services. The results from this phase help developers identify and fix issues before the system goes live.

9.2 Types of System Testing

System testing comprises various testing types to ensure the robustness of SalesXP. The major types include:

1. Functional Testing – Verifies that each feature of SalesXP functions as intended.
2. Performance Testing – Ensures the system can handle multiple sales transactions simultaneously.
3. Security Testing – Identifies vulnerabilities to protect sensitive sales and commission data.
4. Usability Testing – Evaluates the system's user-friendliness for salespersons and managers.
5. Compatibility Testing – Tests the system across different browsers, devices, and platforms.
6. Integration Testing – Ensures seamless communication between different system modules.
7. Regression Testing – Checks whether new updates introduce defects in existing functionality.

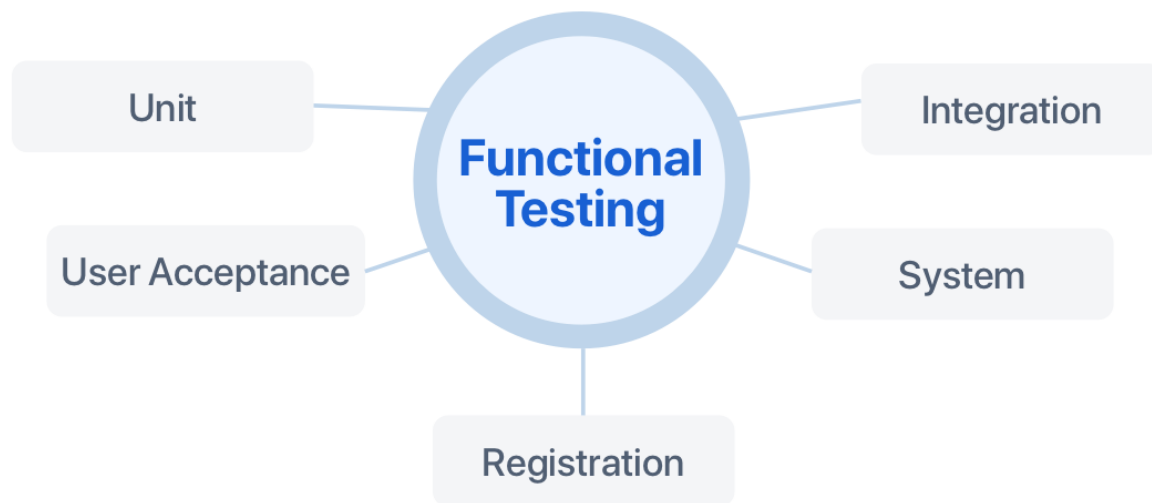
8. Stress Testing – Simulates high loads to analyze system behavior under pressure.

9.3 Functional Testing

Functional testing focuses on verifying that SalesXP's features and functionalities work according to requirements. It tests key components such as:

- User Authentication: Ensuring users can log in securely with unique credentials.
- Sales Entry and Tracking: Validating that every sale is recorded and linked to the correct salesperson.
- Commission Calculation: Confirming that commission payouts are correctly computed based on sales performance.
- Leaderboard Rankings: Checking if salespersons' rankings update in real-time.
- Admin Functions: Ensuring store managers can approve commissions, track performance, and generate reports.

Each function is tested with valid and invalid inputs to verify accuracy, reliability, and data integrity. Automated and manual test cases are executed to ensure error-free operations.

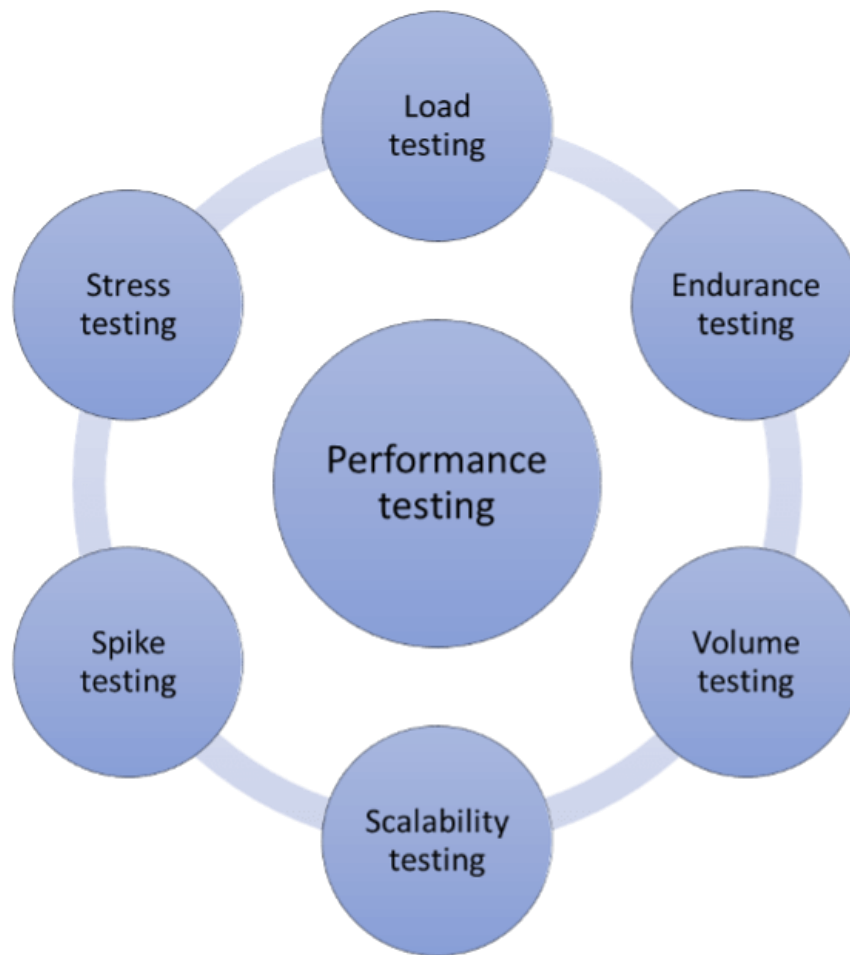


9.4 Performance Testing

Performance testing evaluates SalesXP under different workloads to ensure smooth and efficient operation. This includes:

- **Load Testing:** Simulating multiple users logging in and making transactions at the same time.
- **Response Time Analysis:** Measuring how quickly the system processes sales transactions and updates commissions.
- **Scalability Testing:** Ensuring the system can handle growing amounts of sales data without performance degradation.
- **Database Performance:** Testing query execution times to verify efficient data retrieval.

Performance tests are conducted using automated tools to identify bottlenecks, latency issues, and areas for optimization.



9.5 Security Testing

Security is crucial in a commission tracking system as it deals with financial transactions and personal data. Security testing includes:

- Authentication and Authorization: Ensuring only authorized users can access SalesXP.
- Data Encryption: Verifying that sensitive sales and commission data is stored securely.
- Fraud Prevention Mechanisms: Ensuring that unauthorized sales entries are blocked.
- Penetration Testing: Simulating cyberattacks to identify vulnerabilities.

- **Secure API Testing:** Validating that SalesXP's APIs are protected against unauthorized access.

These tests ensure that SalesXP is protected against security breaches, fraud, and data leaks.



9.6 Usability Testing

Usability testing evaluates how user-friendly SalesXP is for salespersons, store managers, and admins. It assesses:

- **Ease of Navigation:** Ensuring users can easily log in, track sales, and view commissions.
- **Dashboard Design:** Checking whether the user interface is clear and informative.
- **Form Validation:** Ensuring sales entry forms have error messages and guidance for users.
- **Accessibility Features:** Testing compatibility with screen readers and assistive technologies.
- **User Feedback Collection:** Gathering inputs from real users to identify usability

improvements.

Usability testing ensures that SalesXP provides a seamless and intuitive user experience.

Why Usability Test?



Uncover Problems
in the design



Discover Opportunities
to improve the design



Learn About Users
behavior and preferences

NNGROUP.COM **NN/g**

9.7 Compatibility Testing

SalesXP must work across different devices, browsers, and operating systems. Compatibility testing involves:

- **Browser Testing:** Ensuring the system functions correctly on Chrome, Firefox, Edge, and Safari.
- **Device Testing:** Checking performance on desktops, tablets, and mobile devices.

- **Operating System Testing:** Validating compatibility with Windows, macOS, Linux, iOS, and Android.
- **POS System Integration:** Ensuring smooth communication with barcode scanners and payment systems.

This testing ensures that SalesXP delivers a consistent experience across different platforms.



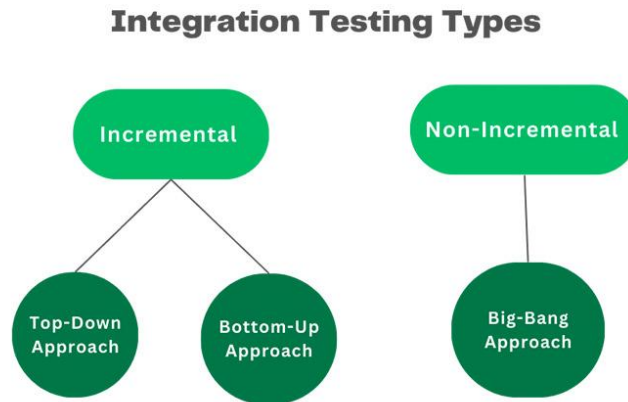
9.8 Integration Testing

Integration testing focuses on verifying seamless communication between different modules of SalesXP. It tests:

- **Frontend & Backend Integration:** Ensuring that UI inputs correctly interact with the database.
- **Database Connectivity:** Verifying that sales data is accurately stored and retrieved.
- **External System Communication:** Ensuring compatibility with third-party APIs and POS machines.

- **Module Interaction:** Checking whether commission calculations update leaderboard rankings in real-time.

Integration testing ensures smooth and efficient operation between all system components.



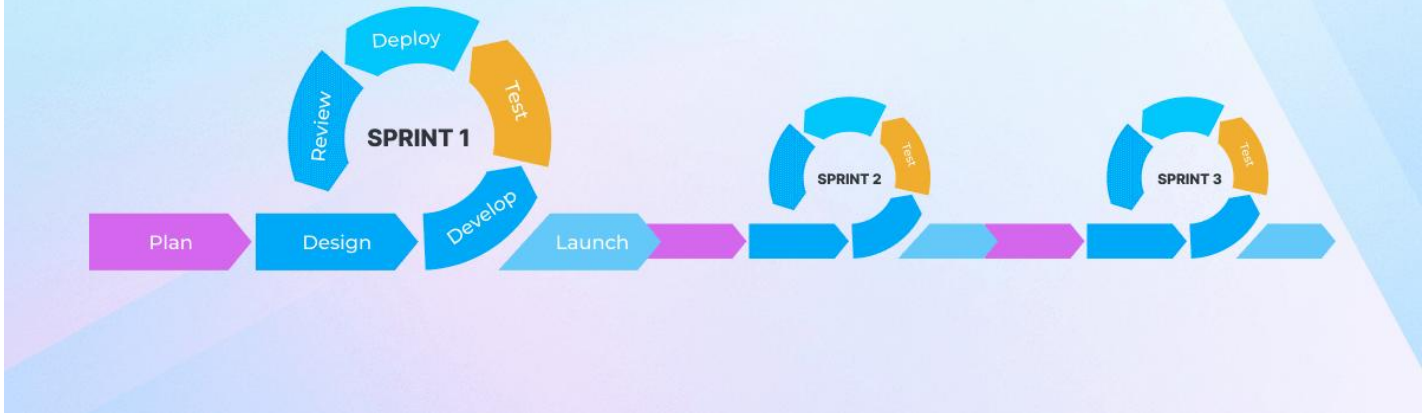
9.9 Regression Testing

Whenever new features or updates are introduced, regression testing ensures that they don't break existing functionalities. This includes:

- Retesting core functionalities after software updates.
- Checking backward compatibility with previous versions.
- Validating new features alongside existing ones.
- Running automated test cases to catch unexpected issues.

Regression testing is essential to maintain system stability and ensure continuous improvement.

What Is Regression Testing



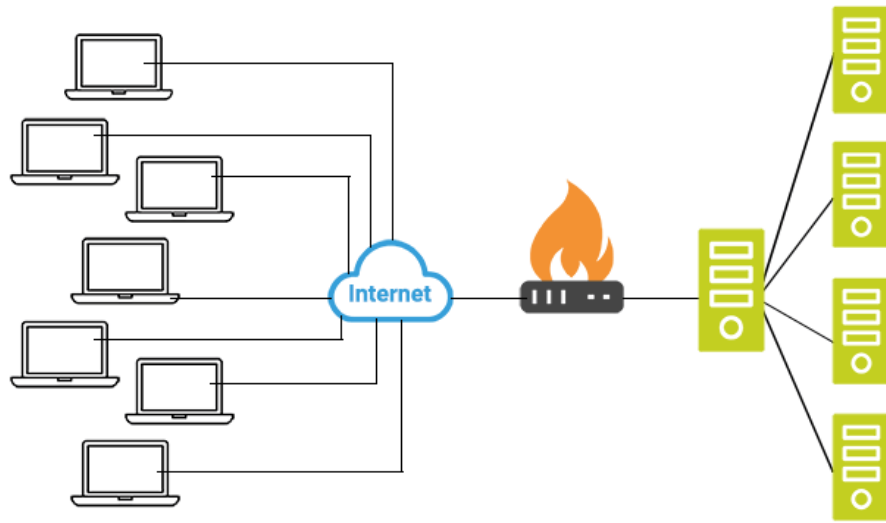
9.10 Stress Testing

Stress testing examines how SalesXP performs under extreme conditions, such as:

- Handling large-scale sales transactions simultaneously.
- Simulating high traffic loads on the system.
- Testing system response when database storage is nearly full.
- Pushing API requests beyond normal limits.

This helps in identifying system breaking points and optimizing performance accordingly.

Stress Testing



Test cases:

Test Case ID	Test Scenario	Test Steps	Expected Result
TC_01	User Login	1. Enter valid credentials 2. Click login	User successfully logs in and is redirected to the dashboard
TC_02	Invalid Login Attempt	1. Enter incorrect credentials 2. Click login	System displays an error message and denies access
TC_03	Sales Entry	1. Scan barcode of sold product 2. Confirm sale	Sale is recorded and associated commission is updated
TC_04	Commission Calculation	1. Complete multiple sales 2. Check commission dashboard	Commission updates correctly based on sales amount

CHAPTER 10: INPUT AND OUTPUT DESIGN

10.1 Input Design

Input design is crucial to ensuring the accuracy, completeness, and efficiency of data entered into the SalesXP system. A well-structured input system minimizes errors and enhances user experience. The key inputs in the SalesXP system include:

1. Salesperson Login: Users enter their unique ID and password to access their dashboard.
2. Sales Entry: Barcodes of sold items are scanned to register sales.
3. Commission Rules Setup: Store managers define commission percentages for different sales levels.
4. Leaderboard Tracking: Sales performance updates based on completed sales.
5. Stock Promotion Input: Store managers set bonuses for promoting slow-moving items.
6. Payout Requests: Salespersons can request commission withdrawals when eligible.
7. Admin Dashboard Inputs: Managers enter filters to view reports by salesperson, date range, or location.

Principles of Good Input Design:

- Ensure data validation to prevent incorrect entries.
- Provide clear input fields with auto-suggestions where applicable.
- Implement error messages for invalid inputs to enhance user guidance.
- Use dropdowns and barcode scanning to reduce manual entry errors.

10.2 Output Design

Output design ensures that the information processed by the system is presented in a meaningful format to users. The key outputs in the SalesXP system include:

1. Sales Dashboard: Displays real-time sales, commission earned, and upcoming targets.
2. Commission Summary Report: Shows total earnings, bonuses, and payout history.
3. Leaderboard Output: Displays the ranking of salespersons based on performance.
4. Admin Reports: Provides sales insights, stock movement, and employee performance.
5. Fraud Detection Alerts: Flags suspicious activities, such as duplicate or fake sales entries.
6. Monthly Performance Charts: Graphs showing individual and team performance trends.

Principles of Good Output Design:

- Use visually clear tables, graphs, and charts for easy interpretation.
- Ensure real-time updates for accurate decision-making.
- Provide export options (PDF, Excel) for report generation.
- Use notifications and alerts to inform users of critical updates.

CHAPTER 11 RESULTS

Directory of E:\ganesh\sales-master\demo

```
03-04-2025  05:13 PM    <DIR>          .
03-04-2025  05:09 PM    <DIR>          ..
03-04-2025  05:09 PM                38 .gitattributes
03-04-2025  05:09 PM            395 .gitignore
03-04-2025  05:09 PM    <DIR>          .mvn
03-04-2025  05:09 PM            335 Dockerfile
03-04-2025  05:09 PM       10,665 mvnw
03-04-2025  05:09 PM        7,061 mvnw.cmd
03-04-2025  05:09 PM        2,637 pom.xml
03-04-2025  05:09 PM    <DIR>          src
03-04-2025  05:13 PM    <DIR>          target
                6 File(s)            21,131 bytes
                5 Dir(s)  77,374,640,128 bytes free
```

E:\ganesh\sales-master\demo>

salesxp /

Drag additional files here to add them to your repository

Or [choose your files](#)

/sales-master/demo/src/main/resources/static/css/admin.css

Uploading 22 of 87 files

/sales-master/demo/Dockerfile

×

/sales-master/demo/mvnw

×

/sales-master/demo/mvnw.cmd

×

/sales-master/demo/pom.xml

×

/sales-master/demo/src/main/java/com/example/demo/auth/controller/LoginController.java

×

/sales-master/demo/src/main/java/com/example/demo/auth/controller/SignupController.java

×

/sales-master/demo/src/main/java/com/example/demo/auth/dto/LoginRequest.java

×

/sales-master/demo/src/main/java/com/example/demo/auth/dto/LoginResponse.java

×

/sales-master/demo/src/main/java/com/example/demo/auth/dto/SignupRequest.java

×

/sales-master/demo/src/main/java/com/example/demo/auth/exception/GlobalExceptionHandler.java

×

Q Search ^ K

+ New ? G

ite your team

services up +

DEV

Static Site

Web Service

Private Service

Background Worker

Cron Job

Postgres

Key Value

Project

Blueprint

salescommission /



Drag additional files here to add them to your repository

Or [choose your files](#)

/demo/src/main/java/com/example/demo/controller/HealthController.java

Uploading 15 of 87 files

/demo/Dockerfile	×
/demo/mvnw	×
/demo/mvnw.cmd	×
/demo/pom.xml	×
/demo/src/main/java/com/example/demo/auth/controller/LoginController.java	×
/demo/src/main/java/com/example/demo/auth/controller/SignupController.java	×
/demo/src/main/java/com/example/demo/auth/dto/LoginRequest.java	×
/demo/src/main/java/com/example/demo/auth/dto/LoginResponse.java	×
/demo/src/main/java/com/example/demo/auth/dto/SignupRequest.java	×
/demo/src/main/java/com/example/demo/auth/exception/GlobalExceptionHandler.java	×
/demo/src/main/java/com/example/demo/auth/model/Employee.java	×
/demo/src/main/java/com/example/demo/auth/repository/EmployeeRepository.java	×
/demo/src/main/java/com/example/demo/auth/service/EmployeeService.java	×
/demo/src/main/java/com/example/demo/config/CorsConfig.java	×



Commit changes

Add files via upload

Add an optional extended description...



Commit changes

Cancel

You are deploying a Web Service

Source Code

Git Provider

Public Git Repository

Existing Image

Q Search

chaytupakula / salescommission seconds ago

View repo

chaytupakula / Pharma 3d ago

Name

A unique name for your web service.

example-service-name

Required

Project Optional

Add this web service to a [project](#) once it's created.

Create a new project to add this to?

You don't have any projects in this workspace. [Projects](#) allow you to group resources into environments so you can better manage related resources.

+ Create a project

ⓘ Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

[Upgrade now](#)

April 3, 2025 at 5:50 PM Building

[2e72bdb](#) Add files via upload

Cancel deploy

All logs

Live tail

GMT+5:30

↑

↻

```

Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: micrometer-observation-1.14.5.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: apiguardian-api-1.1.2.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: jackson-annotations-2.18.3.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: slf4j-api-1.7.25.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: byte-buddy-1.15.11.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: spring-tx-6.2.4.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: android-json-0.9.20131108.vaadin1.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: hibernate-validator-8.0.2.Final.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: junit-jupiter-5.11.4.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] Resolved dependency: spring-boot-starter-json-3.4.4.jar
Apr 3 05:51:41 PM #13 17.85 [INFO] -----
Apr 3 05:51:41 PM #13 17.85 [INFO] BUILD SUCCESS
Apr 3 05:51:41 PM #13 17.85 [INFO] -----
Apr 3 05:51:41 PM #13 17.85 [INFO] Total time: 16.048 s
Apr 3 05:51:41 PM #13 17.85 [INFO] Finished at: 2025-04-03T12:21:41Z
Apr 3 05:51:41 PM #13 17.85 [INFO] -----
    
```

You can also use the [Render CLI](#) to explore logs in your command line.

ⓘ Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

[Upgrade now](#)

April 3, 2025 at 5:50 PM

Building

[2e72bdb](#) Add files via upload

Cancel deploy

All logs ▾ 🔍 Search

🔗 Live tail ▾ GMT+5:30 ⬆ ⬇ ↶

```
Apr 3 05:56:36 PM 2025-04-03T12:26:36.725Z INFO 1 --- [demo] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 95 ms
Apr 3 05:57:13 PM  --> New primary port detected: 8080. Restarting deploy to update network configuration...
Apr 3 05:57:14 PM  --> Docs on specifying a port: https://render.com/docs/web-services#port-binding
Apr 3 05:57:42 PM 
Apr 3 05:57:42 PM  . ____ _ _ _ _
Apr 3 05:57:42 PM  / \ / \ _ _ _ _ ( ) _ _ _ _ \ \ \ \
Apr 3 05:57:42 PM  ( ( ) \ _ _ / _ _ / _ _ / _ _ / \ \ \ \
Apr 3 05:57:42 PM  \ \ / \ _ _ | _ | _ | _ | _ | _ | _ ) ) ) )
Apr 3 05:57:42 PM  '  _/ _/ _ _ | _ | _ | _ _/ _ _/ _ _/ _ _/
Apr 3 05:57:42 PM  =====|_|=====|__/_/ _/_/_/
Apr 3 05:57:42 PM 
Apr 3 05:57:42 PM  :: Spring Boot ::                (v3.4.4)
Apr 3 05:57:42 PM 
Apr 3 05:57:44 PM 2025-04-03T12:27:44.021Z INFO 1 --- [demo] [main] com.example.demo.DemoApplication : Starting DemoApplication v0.0.1-SNAPSHOT using Jav
a 21.0.2 with PID 1 (/app/app.jar started by root in /app)
Apr 3 05:57:44 PM 2025-04-03T12:27:44.025Z INFO 1 --- [demo] [main] com.example.demo.DemoApplication : No active profile set, falling back to 1 default p
rofile: "default"
```

You can also use the [Render CLI](#) to explore logs in your command line.

Looking for more logs? Try [Log Streams](#).

April 3, 2025 at 5:50 PM Building

Cancel deploy

[2e72bdb](#) Add files via upload

All logs

Search

Live tail

GMT+5:30

↑

↶

```
Apr 3 05:58:54 PM 2025-04-03T12:28:54.731Z INFO 1 --- [demo] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
Apr 3 05:59:03 PM 2025-04-03T12:29:03.124Z WARN 1 --- [demo] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
Apr 3 05:59:03 PM 2025-04-03T12:29:03.623Z INFO 1 --- [demo] [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
Apr 3 05:59:09 PM 2025-04-03T12:29:09.922Z INFO 1 --- [demo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 6060 (http) with context path '/'
Apr 3 05:59:10 PM 2025-04-03T12:29:10.121Z INFO 1 --- [demo] [main] com.example.demo.DemoApplication : Started DemoApplication in 95.3 seconds (process running for 104.548)
Apr 3 05:59:11 PM 2025-04-03T12:29:11.325Z INFO 1 --- [demo] [nio-6060-exec-1] o.a.c.o.c.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
Apr 3 05:59:11 PM 2025-04-03T12:29:11.325Z INFO 1 --- [demo] [nio-6060-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
Apr 3 05:59:11 PM 2025-04-03T12:29:11.327Z INFO 1 --- [demo] [nio-6060-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
Apr 3 05:59:14 PM ==> Your service is live 🚀
Apr 3 05:59:19 PM ==> Detected a new open port HTTP:6060
```



SalesXP

Streamlined sales management platform for your business



Administrator

Manage users, settings, and monitor your entire sales ecosystem

➔ Sign In



Billing Desk

Process payments, manage invoices, and handle financial transactions

➔ Sign In



Sales Person

Track leads, manage customer relationships, and close deals

➔ Sign In



Welcome Back

Employee ID

 Enter your employee ID

Password

 Enter your password

☐ Remember me

Sign In

[Forgot Password?](#)

OR



[Don't have an account? Sign up](#)

Employee Registration Portal

1

Personal Information

2

Account Setup

3

Employment Details

Employment Information

Employee ID

123

Date of Joining

1



February



2024



Branch

Main Branch



Back



Processing...

Already registered? [Login here](#)

< >

employees

✕

id	address	branch	date_of_birth	date_of_joining	email	employee_id	full_name	gender	phone_number	role	password	
13	New york	East Branch	2001-02-07	2025-06-05	mark@gmail.com	1	mark	Male	7896543244	Admin	Mark@2025	
14	amcerica	Main Branch	1989-12-10	2024-05-11	cap@gmail.com	2	steve	Male	0987654321	Admin	Avengers@...	
15	AP	Main Branch	1997-04-08	2023-06-07	ram@gmail.com	42	ram	Male	9867094233	Admin	Ram@2025	
16	America	Main Branch	2002-06-05	2025-07-06	steve@gmail.com	77	steve	Male	1234567895	Admin	Steve@2025	
17	uk	Main Branch	2003-07-07	2025-05-06	natash@gmail.c...	14	natasha	Female	1234567891	Admin	Natasha@2...	
18	uk	Main Branch	2000-07-06	2020-06-06	harry@gmail.com	8	Harry	Male	7095704233	Admin	Harry@2025	
19	Us	Main Branch	2000-09-05	2019-06-08	bruce@gmail.com	99	Bruce	Male	1234567899	Admin	Bruce@2025	
20	bza	Main Branch	2006-01-01	2024-02-01	ramzan@gmail.c...	123	john	Male	6304656459	Admin	Vasavi@123	

Admin Panel

- Dashboard
- Users
- Reports
- Settings



Dashboard

Welcome to the Admin Dashboard! Here you can manage the entire system.

Inventory Dashboard

Manage your inventory items

SN	Product	Material	Color	Qty	Wishlist	Date	Price	Disc%	Sell	Barcode
1	Cotton Silk	Cotton Silk	Blue	31	No	26-01-24	10074.35	20.63	11080.01	BAR419007
2	Tussar Silk	Tussar Silk	Silver	45	Yes	22-12-24	9006.03	20.48	6918.72	BAR555506
3	Kandipuram	Kandipuram	Purple	32	No	03-07-24	17385.01	24.48	13467.88	BAR208814

- AddDeleteExport

salescommission.onrender.com says

Inventory updated successfully!

OK

Add New Item

Product:

123

Material:

silk

Color:

green

Qty:

2

Blouse:

Yes

Date:

03 - 04 - 2025

Price:

1000

Disc%:

10

Sell:

900

Baroode:

BAR123456

Cancel

Insert

Dashboard

Welcome to the Admin Dashboard! Here you can manage the entire system.

Inventory Dashboard

Manage your inventory items

SNO	Product	Material	Color	Qty	Blouse	Date	Price	Disc%	
1	Cotton Silk	Cotton Silk	Blue	21	No	28-01-24	15074.30	26.63	
2	Tussar Silk	Tussar Silk	Silver	45	Yes	22-12-24	9806.83	29.45	
3								24.48	
4								10.00	

Add Delete

Delete Item

Select Barcode:

-- Select Barcode --

-- Select Barcode --

BAR415907

BAR956606

BAR288814

BAR123456

4/3/25, 6:07 PM

Admin Panel

SNO	Product	Material	Color	Qty	Broute	Date	Price	Disc%	Sell	Barcode
1	Cotton Silk	Cotton Silk	Blue	21	No	28-01-24	15074.30	26.63	11060.01	BAR415907
2	Tussar Silk	Tussar Silk	Silver	45	Yes	23-12-24	9806.83	29.45	6918.72	BAR0556906
3	Kanchipuram	Kanchipuram	Purple	32	No	03-07-24	17660.01	24.48	13487.88	BAR28814

1/1

Print

1 sheet of paper

Destination

Microsoft Print to PDF

Pages

All

Layout

Portrait

Color

Color

More settings

Print

Cancel

Admin Panel

Dashboard

Users

Reports

Settings

Users

Manage system users

Approved Members

Name	Email	Role
Alice Johnson	alice@example.com	L1
Bob Williams	bob@example.com	L1
Williams	Williams@example.com	L2
Henry	Henry@example.com	L2
Peter	Peter@example.com	L2
Sophia	Sophia@example.com	L2
Laila	Laila@example.com	L3
Lerry	lerry@example.com	L3
Micheal	Micheal@example.com	L3
Trimmsom	Trimmsom@example.com	L4

New Requests

Name	Email	Role	Actions
Charlie Brown	charlie@example.com	L1	<div>ApproveReject</div>
Diana Green	diana@example.com	L3	<div>ApproveReject</div>

Admin Panel

- Dashboard
- Users
- Reports
- Settings



Reports

View system reports

Stock Information - Sarees

Total Products	Sold	Remaining	Revenue (₹)
325	215	110	₹267,999

Team Members Levels

Level	Number of Members
Junior Members [L1]	2
Mid-Level Members [L2]	4
Senior Members [L3]	3
Team Leads [L4]	1


```
C:\users\admin> select * from saree_inventory;
```

SNO	Product	Material	Color	Qty	Blouse	Date	Price	Disc%	Sell	Barcode
1	Cotton Silk	Cotton Silk	Blue	21	No	28-01-24	15074.30	26.63	11060.01	BAR415907
2	Tussar Silk	Tussar Silk	Silver	45	Yes	22-12-24	9806.83	29.45	6918.72	BAR956606
3	Kanchipuram	Kanchipuram	Purple	32	No	03-07-24	17860.01	24.48	13487.88	BAR288814
4	Handloom	Handloom	Blue	46	No	14-07-24	1427.20	29.26	1009.60	BAR424400
5	Chanderi	Chanderi	Pink	1	Yes	21-11-24	14574.68	19.21	11774.88	BAR504901
6	Organza	Organza	Orange	27	No	31-01-25	7970.20	13.55	6890.24	BAR722707
7	Tant Cotton	Tant Cotton	Black	39	Yes	31-01-25	1461.54	21.48	1147.60	BAR414932
8	Organza	Organza	Blue	40	No	22-12-24	9440.21	26.14	6972.54	BAR927288
9	Tussar Silk	Tussar Silk	Maroon	27	Yes	09-05-24	13095.07	19.63	10524.51	BAR141158
10	Mysore Silk	Mysore Silk	Orange	47	No	06-09-24	8554.58	26.25	6309.80	BAR534188

19 rows in set (0.00 sec)

```
C:\users\admin> select * from sales_person;
```

ID	Emp ID	Sales	Commission	Level
1	EMP001	1250000.00	62500.00	L3
2	EMP002	875000.50	43750.03	L2
3	EMP003	2100000.75	105000.04	L4
4	EMP004	650000.00	32500.00	L2
5	EMP005	980000.25	49000.01	L3
6	EMP006	420000.99	21000.05	L1
7	EMP007	1500000.50	75000.03	L3
8	EMP008	3200000.00	160000.00	L4
9	EMP009	725000.75	36250.04	L2
10	EMP010	1100000.25	55000.01	L3

10 rows in set (0.00 sec)

```
C:\users\admin> select * from sales
```

ID	Emp ID	Customer	Number	Date/Time	Amount	Commission
1	EMP001	Rahul Sharma	9876543210	2023-11-01 10:30	15000.00	750.00
2	EMP002	Priya Patel	8765432109	2023-11-01 11:45	22500.50	1125.03
3	EMP003	Amit Singh	7654321098	2023-11-02 09:15	18000.75	900.00
4	EMP001	Neha Gupta	6543210987	2023-11-02 14:20	32000.00	1600.00
5	EMP004	Vikram Joshi	5432109876	2023-11-03 12:00	27500.25	1375.01
6	EMP002	Ananya Reddy	4321098765	2023-11-03 16:30	18999.99	950.00
7	EMP003	Arjun Malhotra	3210987654	2023-11-04 10:00	42000.50	2100.03
8	EMP005	Divya Iyer	2199876543	2023-11-04 15:45	31500.00	1575.00
9	EMP001	Sanjay Verma	1998765432	2023-11-05 11:30	24500.75	1225.04
10	EMP004	Pooja Mehta	9987654321	2023-11-05 17:15	36000.25	1800.01

10 rows in set (0.00 sec)

Chapter 12: Conclusion & Future Work

12.1 Conclusion

SalesXP has successfully addressed the challenges of traditional sales commission tracking by providing an automated, transparent, and performance-driven solution. The system eliminates the drawbacks of manual tracking methods, such as errors, delays, and fraud, by leveraging barcode-based sales tracking and automated commission calculations. By implementing a structured, level-based reward system, SalesXP incentivizes salespersons to enhance their performance, ensuring continuous motivation. The system fosters a healthy competitive environment through real-time leaderboards, allowing employees to monitor their progress and strive for better rankings. Additionally, SalesXP optimizes stock management by offering higher commissions for selling older or slow-moving inventory, benefiting both employees and businesses.

The introduction of analytics and performance tracking features enables store managers to gain valuable insights into employee sales patterns, helping them design better training and incentive programs. The automated calculations and fraud prevention mechanisms significantly reduce administrative workload, allowing businesses to focus on strategic growth. SalesXP's adaptability makes it suitable for various retail businesses, from small stores to large enterprises. By integrating with existing retail management solutions, the platform ensures seamless implementation without disrupting business operations.

In summary, SalesXP enhances employee engagement, improves sales efficiency, and streamlines commission distribution, leading to increased profitability for retail businesses. The system aligns with the ongoing digital transformation in retail, paving the way for future advancements in sales and performance tracking. With its user-friendly interface, real-time analytics, and structured commission model, SalesXP stands out as a powerful tool for modern retail management.

12.2 Future Work

While SalesXP has introduced a structured and efficient approach to sales commission tracking, there are several enhancements and expansions that can be implemented in future iterations:

1. AI-Powered Sales Forecasting

- Implementing artificial intelligence (AI) models to analyze sales patterns and predict future trends.
- Providing salespersons and store managers with predictive insights to optimize inventory and sales strategies.

2. Gamification for Enhanced Engagement

- Introducing additional gamification elements such as achievement badges, sales challenges, and leaderboards with monthly rewards.
- Creating personalized goal-setting features to encourage employees to set and achieve specific sales milestones.

3. Mobile Application for SalesXP

- Developing a mobile app to allow salespersons to track their commissions, sales performance, and rankings on the go.
- Enabling push notifications for commission updates, sales performance, and leaderboard changes.

4. Multi-Currency & International Support

- Expanding the platform to support multiple currencies, tax regulations, and localized commission structures for global scalability.
- Providing language options to cater to a diverse user base.

5. Advanced Fraud Detection Mechanisms

- Implementing AI-driven anomaly detection to prevent fraudulent commission claims.
- Enhancing security measures such as biometric authentication and location-based verification for sales logging.

6. API Integrations with Third-Party Systems

- Expanding API capabilities to integrate with popular ERP, POS, and CRM systems for seamless business operations.
- Enabling data synchronization with accounting and payroll software for automated salary and commission payments.

7. Cloud-Based and SaaS Deployment

- Offering a cloud-based, Software-as-a-Service (SaaS) version of SalesXP to allow businesses to access the system from anywhere.
- Ensuring scalability and remote access for businesses with multiple store locations.

8. Voice and Chatbot Assistance

- Integrating AI-powered voice and chatbot assistants to provide real-time responses to salespersons and managers.
- Automating customer inquiries related to commissions, sales targets, and performance reports.

Chapter 13: References

This section provides a comprehensive list of books, research papers, websites, and technologies that were consulted or used in the development of SalesXP.

13.1 Books & Research Papers

1. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language," Addison-Wesley, 2004.
2. Ian Sommerville, "Software Engineering," Pearson Education, 2015.
3. Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide," Addison-Wesley, 2005.
4. Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, "Database System Concepts," McGraw-Hill, 2019.
5. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, "Mastering Cloud Computing," McGraw-Hill, 2013.
6. Roger S. Pressman, "Software Engineering: A Practitioner's Approach," McGraw-Hill, 2014.
7. Michael J. Pont, "Embedded C," Pearson Education, 2002.
8. Ken Schwaber & Jeff Sutherland, "The Scrum Guide," Scrum.org, 2020.
9. Eric Ries, "The Lean Startup," Crown Publishing, 2011.

13.2 Technologies & Frameworks Used

SalesXP was developed using a combination of frontend, backend, and database technologies. The following frameworks and tools were used:

Frontend Development:

- HTML, CSS, JavaScript, Bootstrap – For designing the user interface and ensuring responsiveness.

Backend Development:

- Core Java, Spring Boot – For handling business logic and API management.
- RESTful API – For enabling communication between the frontend and backend.

Database Management:

- MySQL, Neon DB, TablePlus – For efficient data storage and retrieval.

Development & Deployment Tools:

- Docker – For containerized deployment.
- Maven – For project management and dependency management.
- GitHub – For version control and code collaboration.
- OnRender – For cloud-based application deployment.
- VS Code – For coding and development.

Testing & Security:

- Postman – For API testing and debugging.
- JWT Authentication – For secure authentication and authorization mechanisms.

13.3 Online Resources & Documentation

To ensure best practices and industry-standard implementation, the following official documentation and online resources were referred to:

1. Spring Boot Official Documentation - <https://spring.io/projects/spring-boot>
2. MySQL Developer Guide - <https://dev.mysql.com/doc/>
3. Bootstrap Documentation - <https://getbootstrap.com/docs/>
4. Docker Documentation - <https://docs.docker.com/>
5. Maven Repository - <https://mvnrepository.com/>
6. REST API Design Best Practices - <https://restfulapi.net/>
7. GitHub Guide for Developers - <https://guides.github.com/>
8. Software Engineering Research Papers (IEEE & ACM) - <https://ieeexplore.ieee.org/>

13.4 Industry Use Cases & Case Studies

To validate the effectiveness of SalesXP, various industry case studies and practical use cases

were examined:

1. Amazon Sales Performance Tracking System – How AI and data analytics are used to track and incentivize sales performance.
2. Salesforce Commission Management System – Best practices in automating sales commission calculations.
3. Retail Industry Insights on Employee Performance Management – Reports from McKinsey, Harvard Business Review, and Forbes on sales performance motivation strategies.

