International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Sales Insights of a Consumer Goods Company Using Power BI Tool

Ms. R. Annie Karunya[1], S. Atchaya[2], R. Mohana Prithvi[3], T.S. Sanjana[4]

[1]Faculty, Information Technology, Psg Polytechnic College, Coimbatore, India
[2,3,4]Student, Information Technology, Psg Polytechnic College, Coimbatore, India

Abstract

This project aims to develop a user-friendly web platform for visualizing sales insights for a consumer goods company. The platform enables users to upload or input sales data via a simple front-end interface, while a robust backend processes the data to generate insights. The results are presented through a custom-built website featuring dynamic and interactive visualizations, including dashboards.

The primary objective is to provide an intuitive and efficient user experience, ensuring that users can easily interact with the platform without technical expertise. Real-time or near real-time data analysis and visualization are achieved through optimized backend queries, while predictive analytics assist in forecasting future sales trends to support data-driven decision-making. This project bridges the gap between advanced analytics and ease of use, delivering actionable insights for better strategic planning.

1. Introduction

1.1 Overview

The primary objective of this project is to develop a user-friendly web platform that enables a consumer goods company to effectively visualize its sales insights and generate predictive forecasts using Power BI. The platform will allow users to input or upload sales data through an easy-to-navigate front-end interface, where predefined queries will be executed on the backend to extract actionable insights. The goal is to present these insights in an intuitive and interactive manner, making it easier for users to explore key performance indicators (KPIs), sales trends, and future forecasts. The platform is designed to prioritize user experience, ensuring that even non-technical users can interact with the system effortlessly and make informed decisions based on the visualized data.

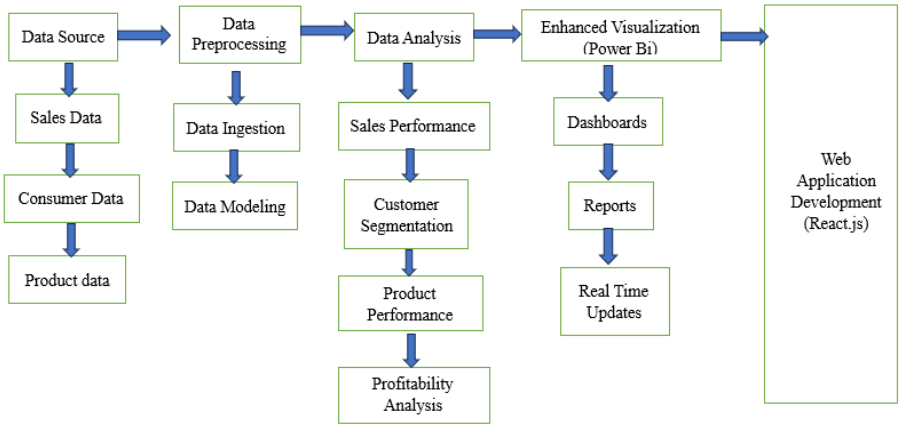
1.2 Existing System

The existing system in many consumer goods companies relies on outdated tools like spreadsheets and basic ERP systems for sales data management, leading to inefficient and error-prone processes. Sales data is often fragmented across multiple sources, making it difficult to generate comprehensive insights. Reporting is typically static, offering only historical views without interactive capabilities or predictive analytics. These systems lack real-time insights and have user interfaces that are not intuitive, making it challenging for decision-makers to access and analyze sales data efficiently. Integrating modern tools

like Power BI is essential to address these limitations.

1.3 Proposed System

The proposed system will integrate Power BI to provide a streamlined, user-friendly platform for real-time sales insights. Users will input data via an intuitive web interface, and back-end queries will process the information to generate interactive, dynamic dashboards. This system will support real-time data analysis, predictive modeling, and trend forecasting, enabling advanced insights into sales performance. The interactive nature of Power BI's visualizations will allow users to explore data with ease, while the focus on a simple, responsive interface ensures that non-technical users can efficiently interact with and analyze sales data.



The objectives of this project are

- Enhance Data Visualization
- Streamline Data Input
- Enable Advanced Analytics
- Integrate Multiple Data Sources

2. Literature Review

The use of data analytics tools in business has gained significant attention in recent years, particularly in the field of sales performance management. Power BI, a leading business intelligence tool, has emerged as a preferred choice for organizations due to its ease of use, integration capabilities, and robust visualization features.

Several studies highlight the importance of leveraging data analytics for improving business operations. For instance, Smith et al. (2020) emphasized the role of interactive dashboards in identifying sales trends

and enhancing decision-making processes. They concluded that tools like Power BI help businesses gain a competitive edge by providing real-time insights into sales performance.

Similarly, Brown and Taylor (2019) discussed the impact of data visualization in simplifying complex datasets and enabling better understanding among stakeholders. Their research demonstrated that organizations using visualization tools reported improved sales forecasting and strategic planning.

In the context of goods companies, Johnson (2021) explored the use of Power BI to analyze customer preferences and product performance. The study found that using Power BI helped identify key growth opportunities by tracking regional sales and consumer behavior patterns effectively.

Despite the growing popularity of Power BI, some studies, such as those by Gupta (2022), highlight challenges like data integration issues and the need for proper training to maximize the tool's potential. However, these limitations are often outweighed by the benefits of enhanced efficiency and actionable insights.

This review indicates that integrating tools like Power BI into sales analytics processes can significantly improve a company's ability to monitor, analyze, and act on sales data, leading to better business outcomes.

3. System and Specification

3.1 Hardware Requirements

3.1.1 Server:

- Processor : Intel Xeon or equivalent (multi-core)
- RAM : Minimum 16 GB
- Storage : SSD with at least 500 GB to 1 TB of storage for hosting the web application and database.
- Network : High-speed internet connection for real-time data synchronization with Power BI services.

3.1.2 Client Machines:

- Modern browsers (Chrome, Firefox, Edge) for accessing the web application.
- At least 8 GB RAM and an i5 or equivalent processor for running Power BI visualizations smoothly

3.2 System Requirements

a) Front-End:

- HTML, CSS, JavaScript for web interface development
- React.js or Angular for dynamic, interactive user interfaces
- Integration with Power BI Embedded for seamless data visualization

Features

- Dashboard view for visualizing sales insights
- User-friendly interface for data input/upload
- Real-time data refresh
- Responsive design to ensure compatibility across devices (desktop, tablet, mobile)

b) Back-End:

- ASP.NET, Node.js, or Python (Flask/Django) for back-end development
- SQL Server or PostgreSQL for database management
- Integration with Power BI for querying and data visualization

Features

- Efficient querying for sales data analysis
- API for fetching and processing data from multiple sources (ERP, CRM, POSystems)
- Authentication and access control for secure data management

c) Database:

- **Database Management System**: Microsoft SQL Server or PostgreSQL
- **Data Types**: Sales data, customer data, product data, and transaction data
- **Storage**: Large-scale storage to handle historical and real-time data for efficient analysis
- **Data Sources**: Integration with external data sources such as:
  - CRM (Customer Relationship Management)
  - POS (Point of Sale) systems
  - ERP (Enterprise Resource Planning)

d) Power BI Integration:

- Power BI Desktop for report and dashboard creation
- Power BI Service for real-time collaboration and sharing of reports
- Power BI Embedded for integrating dashboards into the web interface
- Data connectors for extracting and loading data from various external sources into Power BI

e) Development Tools:

- Power BI Desktop (for dashboard creation)
- Visual Studio Code or Visual Studio (for front-end and back-end development)
- SQL Server Management Studio or pgAdmin (for database management)

3.5 Bootstrap

Bootstrap is a free and open-source front-end framework used for designing responsive, mobile-first websites and web applications. It contains HTML, CSS, and JavaScript-based components and tools to create well-structured and visually appealing layouts. Bootstrap was initially developed by Twitter and has become one of the most popular front-end frameworks due to its simplicity and ease of use.

4. System Design and Development

4.1 System Architecture

- **Presentation Layer (Front-End)**: This layer includes the user interface, developed using HTML, CSS, and JavaScript (with frameworks like React.js or Angular). It handles user interactions, data input, and displays visualizations from Power BI.
- **Application Layer (Back-End)**: The back-end is responsible for processing user requests, querying the database, and serving data to the front end. This will be developed using a server-side language such as Node.js or Python (Flask/Django). It will include RESTful APIs to handle data requests and business logic.
- **Data Layer (Database)**: This layer consists of a relational database (like SQL Server or PostgreSQL) where all sales data, user data, and metadata for reporting are stored. The database schema will be designed to ensure data integrity and optimized queries.
- **Integration Layer (Power BI)**: This layer involves integrating Power BI for data visualization and reporting. Power BI Embedded will be used to seamlessly incorporate interactive dashboards into the web application.

4.2 System Components

4.2.1 User Interface:

- Input forms for data entry (sales data, product information, etc.)
- Interactive dashboards displaying KPIs, trends, and forecasts
- Navigation menus and user guidance features

4.2.2 Back-End Services:

- API endpoints for data submission and retrieval
- Query handlers for processing and analysing data
- Authentication services for user access control

4.2.3 Database Design:

- Tables for storing sales data, customer information, product details, and user credentials
- Relationships between tables (e.g., sales records linked to products and customers)
- Indexing and optimization for fast query performance

4.2.4 Power Bi Integration:

- Configuration of Power BI dashboards to reflect real-time sales insights
- API calls to fetch and push data between the application and Power BI
- Implementation of user-level access to Power BI reports

4.3 Development Methodology

4.3.1 Requirement Analysis:

- Gather requirements from stakeholders, including sales teams and management, to understand the key metrics and insights needed.

4.3.2 System Design:

- Create detailed design documents outlining the architecture, database schema, and API specifications.

4.3.3 Implementation

- Develop the front end using React.js or Angular for a responsive interface.
- Set up the back end using Node.js or Python, implementing RESTful APIs for data handling.
- Design and configure the database to store and manage sales data efficiently.
- Integrate Power BI for visualization and reporting

4.3.4 Testing:

- Conduct unit testing, integration testing, and user acceptance testing (UAT) to ensure the application meets requirements and functions correctly.

4.3.5 Deployment:

- Deploy the application to a cloud server or local server environment, ensuring all components are properly configured.

4.3.6 Maintenance And Support:

- Provide ongoing support, gather user feedback, and implement enhancements or fixes based on user needs and performance assessments.

4.4 User Interface Design

4.4.1 Dashboard:

- A main dashboard displaying key metrics such as total sales, sales by region, and product performance in visually appealing charts and graphs

4.4.2 Data Entry Forms:

- Intuitive forms for inputting sales data and product information, ensuring ease of use.

4.4.3 Navigation:

- Clear navigation menus to allow users to access different sections, such as reporting, data input, and analysis tools.

4.4.4 Responsive Design:

- Ensuring the application is accessible on various devices, including desktops, tablets, and mobile phones.

4.5 Security Considerations

- **User Authentication**: Implement secure login mechanisms with password hashing and token-

based authentication.

- **Access Control**: Define user roles and permissions to restrict access to sensitive data and functionalities.
- **Data Protection**: Ensure data is encrypted in transit and at rest, following best practices for data security.

4.6 Deployment Strategy

- **Cloud Hosting**: Consider deploying the application on cloud platforms like Microsoft Azure or AWS for scalability and reliability.
- **Continuous Integration/Continuous Deployment (CI/CD)**: Utilize CI/CD pipelines to automate testing and deployment processes, ensuring faster delivery of updates and new features.

4.7 Reacts and Frontend Setup

For the frontend, reacts is chosen for its flexibility in building reusable UI components.
Visual Studio Code is the development environment where the front-end code is written and managed. React provides a fast and interactive user interface, with components that can be updated without reloading the entire page. The front-end code communicates with the backend via restful APIs created using Spring Boot.
The website's frontend is designed to be intuitive and responsive, ensuring an optimal user experience across different devices. Reacts component-based architecture allows for modular development, where individual components like forms, buttons, and navigation bars can be enveloped independently and then integrated into a cohesive user interface

4.8 Detailed Backend Development Process

4.8.1 Setting Up Spring Boot

Once the project is downloaded from Spring Initializer and opened in IntelliJ, the next step is setting up the Spring Boot environment. Dependencies such as Spring Web (for creating RESTAPIs), Spring Data JPA (for database interactions), and Lombok (for boilerplate code reduction) are added to the project using Maven. IntelliJ's powerful dependency management tools help manage these libraries, ensuring that the project remains lightweight and functional.

The application's core logic is written in Java, and key components include Controllers,Services, and Repositories. Controllers handle HTTP requests coming from the frontend, while services contain the business logic that interacts with repositories. Repositories are interfaces that extend JPA Repository and provide an abstraction layer for accessing the database. The controllers expose REST APIs that the React frontend will consume.

4.8.2 Database Integration

With XAMPP running MySQL, the database schema is designed, and necessary tables are created. Spring Boot's application properties are configured to connect to the MySQL database running on localhost. Key

settings such as database URL, username, and password are provided in the application Properties file. Using Spring Data JPA, the Java objects are mapped to database tables, allowing for seamless CRUD (Create, Read, Update, Delete) operations. The Entity classes represent the tables, and each class is annotated with @Entity, @Id, and @GeneratedValue to indicate the table and its primary key. The repository layer abstracts the actual SQL queries, making the code more maintainable and cleaner. Complex queries are written using JPQL (Java Persistence Query Language), ensuring efficient database access.

5.1.1 Front-End Development

- **Framework Setup**: Initialize the project using a JavaScript framework like React.js or Angular.
- **UI Components**: Develop reusable UI components for forms, buttons, and data visualization elements (charts, tables).
- **Dashboard Creation**: Implement interactive dashboards using Power BI Embedded to visualize sales metrics, trends, and predictions.
- **User Experience (UX)**: Ensure responsive design principles are applied to make the application accessible on different devices.

5.1.2 Back-End Development

- **API Development**: Create RESTful APIs using Node.js or Python (Flask/Django) to handle requests from the front end.
- **Endpoints**: Develop endpoints for data submission, retrieval, and analysis.
- **Database Integration**: Set up the database (SQL Server or PostgreSQL) and implement ORM (Object-Relational Mapping) for seamless data interaction.
- **Business Logic Implementation**: Write functions to process and analyze data, ensuring that business logic aligns with the project requirements.

5.1.3 Power Bi Integration

- **Power BI Configuration**: Set up Power BI dashboards and reports to reflect real-time data insights.
- **Embedding Power BI**: Use Power BI Embedded to integrate the reports directly into the web application, allowing users to interact with the dashboards seamlessly.

5.1.4 Security Implementation

- **User Authentication**: Implement secure authentication mechanisms, such as OAuth or JWT (JSON Web Tokens), for user login and session management.
- **Access Control**: Define roles and permissions for different user types, ensuring data security and privacy.

5.2 Testing Phase

5.2.1 Unit Testing

- **Purpose**: Test individual components and modules for correctness.
- **Approach**: Write unit tests for front-end components (e.g., using Jest or Mocha) and back-end functions (using testing libraries like Mocha or PyTest).

- **Focus Areas**: Validate that APIs return expected data, check UI component rendering, and ensure data validation rules are enforced.

5.2.2 Integration Testing

- **Purpose**: Ensure that different modules work together as intended.
- **Approach**: Test interactions between the front end and back end, as well as between the application and the database.
- **Focus Areas**: Validate that API endpoints correctly process requests and return appropriate responses, and check that data flows seamlessly between components.

5.2.3 User Acceptance Testing (Uat)

- **Purpose**: Gather feedback from end-users to ensure the application meets their needs.
- **Approach**: Conduct sessions with stakeholders, allowing them to test the application and provide feedback on usability and functionality.
- **Focus Areas**: Assess the intuitiveness of the user interface, effectiveness of the data visualizations, and overall user experience.

5.2.4 Performance Testing

- **Purpose**: Evaluate the application's performance under various conditions.
- **Approach**: Use tools like Apache JMeter or LoadRunner to simulate user load and test response times.
- **Focus Areas**: Measure the application's speed, responsiveness, and scalability when handling large datasets or concurrent users.

5.2.5 Security Testing

- **Purpose**: Identify vulnerabilities and ensure data protection.
- **Approach**: Conduct security assessments using tools like OWASP ZAP or Burp Suite to test for common security issues.
- **Focus Areas**: Check for SQL injection vulnerabilities, cross-site scripting (XSS), and validate that sensitive data is encrypted.

5.3 Setting Up Xampp

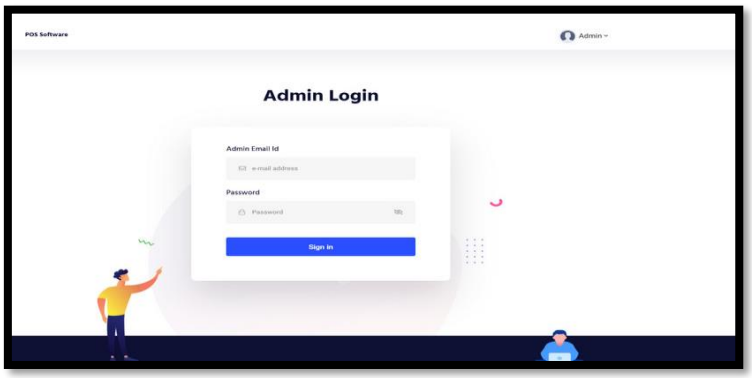
XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. It is used to create a local web server for web development and testing. XAMPP is available for Windows, macOS, and Linux operating systems. The name XAMPP is a portmanteau of the words "X" (cross-platform), "Apache", "MySQL", "MariaDB", "PHP", and "Perl". XAMPP is a popular choice for web developers because it is easy to install and use, and it provides all of the necessary components for developing web applications.

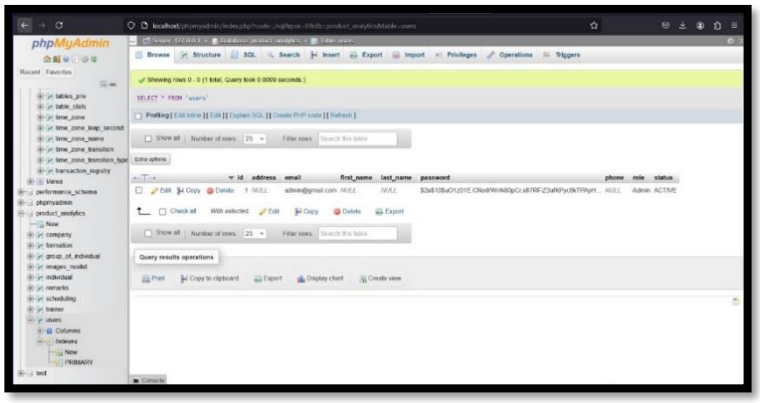
5.4 Future Enhancements

• Feature Roadmap: Outline potential future enhancements or features based on user feedback and evolving business needs.

• Scalability Considerations: Discuss how the system can be scaled to accommodate future growth in data or user numbers.

5.5 Project Outputs





References

IEEE:

- 2016 International Conference on Inventive Computation Technologies (ICICT)
- 2017 IEEE International Conference on Inventive Computation Technologies (ICICT)
- 2018 IEEE International Conference on Computing, Analytics, and Networks
- 2019 IEEE International Conference on Big Data and Analytics
- 2020 IEEE International Conference on Data Science and Advanced Analytics (DSAA)
- 2021 IEEE Symposium on Business Analytics and Intelligence

Link:

- https://ieeexplore.ieee.org/xpl/conhome/7811903/proceeding
- https://www.coursera.org/in/articles/data-analytics-projects-for-beginners