**CHAPTER 3** (PART-A)

**Problem Definition**

**1. Defining the Problem:** The primary goal of this project is to create a comprehensive **Retail Management Website** that will help retailers manage their store operations, track sales, manage inventory, and generate insights through a **Power BI Dashboard**. The dashboard will provide detailed visual analytics on sales, inventory status, customer behavior, and business performance.

**2. Details of the Problem:** Retailers often face difficulties in tracking inventory, managing customer orders, and understanding sales trends. Without proper tools, this can lead to mismanagement of stock, loss of sales opportunities, and difficulty in making data-driven business decisions.

**3. Dividing the Problem into Subproblems:**

* **Inventory Management**: Retailers struggle to keep track of stock levels, manage incoming products, and restock.
* **Sales Management**: Tracking daily sales, orders, and revenue generation becomes complex, especially with manual systems.
* **Customer Management**: Understanding customer behavior, purchase history, and loyalty patterns is challenging without proper data analytics.
* **Data Insights & Reporting**: Without effective data visualization, it’s difficult to get real-time insights into sales trends, profit margins, and operational efficiency.

**4. Defining Each Subproblem Clearly:**

* **Inventory Management**:
  + The system needs to record product details, update stock levels, set restock alerts, and manage multiple categories of products.
* **Sales Management**:
  + The system should capture all sales transactions, generate invoices, process customer orders, and handle both online and in-store purchases.
* **Customer Management**:
  + The system should allow tracking of customer data, such as purchase history, preferences, and create customer profiles for better service and personalized marketing.
* **Data Insights & Reporting**:
  + Power BI should integrate with the system to provide visual reports like daily sales trends, inventory status, and customer analytics to help retailers make informed decisions.

**Requirement Specification**

**1. Functional Requirements:**

* **Inventory Management**:
  + Add, update, and delete product details.
  + Track stock levels in real-time.
  + Notify when stock falls below a certain level.
  + Categorize products (electronics, groceries, etc.).
* **Sales Management**:
  + Handle transactions and record sales.
  + Generate and download invoices in PDF.
  + Monitor daily, weekly, and monthly sales.
* **Customer Management**:
  + Store customer data securely.
  + Track customer purchase behavior.
  + Create loyalty programs based on purchase patterns.
* **Data Insights (Power BI Integration)**:
  + Real-time dashboards for sales, inventory, and customer data.
  + Visualize key performance indicators (KPIs) such as sales volume, top-selling products, and stock trends.
  + Compare performance over time and generate custom reports.

**2. Non-Functional Requirements:**

* **Security**: The system should ensure customer data protection with encryption.
* **Performance**: The system should handle multiple concurrent transactions without slowdowns.
* **Scalability**: The system should handle an increasing number of products, customers, and sales data as the business grows.
* **Availability**: The system must be available 24/7 with minimal downtime.

**3. Actions that Can Be Done:**

* Manage inventory efficiently by viewing available stock, adding new products, and receiving restock alerts.
* View real-time sales data and trends through the Power BI dashboard.
* Create customer profiles and track customer activities.
* Download detailed reports for business analysis and decision-making.

#### ****Planning & Scheduling****

A project plan will be created using a **Gantt Chart** and **PERT (Program Evaluation Review Technique)** to ensure tasks are completed on time and within the project scope.

##### **Gantt Chart** (Project Timeline Overview)

The chart will include the following key stages:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration** | **Start Date** | **End Date** |
| Project Initiation | 1 week | 01-Oct-2024 | 07-Oct-2024 |
| Requirements Gathering & Analysis | 2 weeks | 08-Oct-2024 | 21-Oct-2024 |
| UI/UX Design | 3 weeks | 22-Oct-2024 | 11-Nov-2024 |
| Database Design | 2 weeks | 12-Nov-2024 | 25-Nov-2024 |
| Backend Development | 4 weeks | 26-Nov-2024 | 23-Dec-2024 |
| Power BI Integration | 2 weeks | 24-Dec-2024 | 06-Jan-2025 |
| Testing & Quality Assurance | 2 weeks | 07-Jan-2025 | 20-Jan-2025 |
| Deployment | 1 week | 21-Jan-2025 | 27-Jan-2025 |

##### **PERT Chart** (Key Milestones)

* **T1**: Project Initiation (1 week)
* **T2**: Requirements Gathering (2 weeks)
* **T3**: UI/UX Design (3 weeks)
* **T4**: Backend Development (4 weeks)
* **T5**: Power BI Integration (2 weeks)
* **T6**: Testing (2 weeks)
* **T7**: Deployment (1 week)

**Software Requirements :**

* **Programming Languages**: Python (for Flask or Django), JavaScript (for frontend)
* **Frameworks**: Django or Flask (for backend development), React or Bootstrap (for frontend)
* **Database**: MySQL or PostgreSQL
* **Visualization Tool**: Power BI
* **Version Control**: Git, GitHub or GitLab
* **Deployment**: Cloud platform (AWS, Heroku) or local server