CRM Dashboard Portal

By

Lavanya Mithbavkar

AF04963944

Index

Sr. no	Topic	Page no
1	Title of Project	1
2	Acknowledgement	3
3	Abstract	4
4	Introduction	5
5	System Analysis	8
6	Screenshots	27
8	Implementation	31
9	Testing	33
10	Results and Discussion	36
11	Conclusion and Future	38
12	Bibliography and References	40

Acknowledgement

I would like to express my sincere gratitude to my project guide for their invaluable support, guidance, and encouragement throughout the development of the CRM Dashboard Portal. Their insights played a crucial role in shaping both the technical and functional aspects of this project.

Abstract

The CRM Dashboard Portal is a web-based system developed to simplify and enhance lead management for businesses. It provides a centralized platform to track, analyze, and export lead data with clarity and precision. The system is built using ReactJS for the frontend, Node.js and Express for the backend, and MySQL for data storage.

Key features include lead scoring, sentiment tagging, region-based analytics, and PDF export functionality. The dashboard offers dynamic chart visualizations using Chart.js, enabling users to interpret performance metrics at a glance. Each module is designed with a premium UI and modular layout, ensuring maintainability and user-centric interaction.

The portal supports strategic decision-making by transforming raw lead data into actionable insights. It also includes a highlights panel for quick summaries and a dedicated Reports section for deeper analysis. The PDF export feature, implemented using html2canvas and jspdf, allows users to generate professional reports directly from the interface.

This project demonstrates the integration of modern web technologies to build a responsive, analytics-driven CRM solution tailored for real-world business needs.

1. Introduction

Managing customer relationships is a crucial aspect of modern business operations. In an era where lead sources are diverse and competition is high, businesses often struggle to organize, analyze, and convert leads effectively. Poor lead management can result in missed opportunities, low conversion rates, and inefficient communication. To overcome these challenges, technology plays a vital role in helping organizations track, score, and optimize their lead engagement strategies.

CRM Dashboard Portal is a comprehensive customer relationship management system developed to simplify the process of lead tracking and performance analysis. It enables users to monitor lead sources, assign sentiment tags, score leads, and generate exportable reports in real-time. The application serves as a digital workspace that not only organizes lead data but also encourages strategic decision-making through modular design and visual insights.

1. Background of the Study

In today's digital era, customer acquisition and retention are key drivers of business growth. While many organizations invest in marketing and outreach, most lack the tools to manage incoming leads effectively. Traditional methods such as spreadsheets or manual tracking are time-consuming, error-prone, and lack visualization capabilities.

The demand for personalized, digital CRM tools has grown significantly in recent years. Web-based applications provide accessibility, automation, and real-time analytics, making lead management more efficient. CRM Dashboard Portal has been developed with this vision — to empower businesses by providing an all-in-one solution for lead tracking, scoring, and reporting.

2. Motivation

The development of CRM Dashboard Portal is motivated by several key challenges observed in business environments:

- **Lack of Lead Visibility** – Businesses often receive leads from multiple sources but fail to track their origin, status, or potential.
- **Poor Conversion Strategy** – Without scoring and sentiment tagging, leads are treated uniformly, reducing conversion efficiency.
- **Data Fragmentation** – Lead data is scattered across platforms, making analysis difficult and time-consuming.
- **Need for Visualization** – Understanding lead performance is easier with visual representation (charts, graphs, highlights) than with raw tables.
- **Export Requirements** – Businesses need professional reports for review meetings, client updates, or internal analysis.

CRM Dashboard Portal addresses these issues by combining lead tracking, sentiment tagging, score distribution, and PDF export into one platform.

3. Significance of the Study

This project is significant for several reasons:

- **For Businesses**: Helps teams monitor lead performance, prioritize high-potential leads, and improve conversion strategies.
- **For Sales Teams**: Provides a structured overview of lead sources, scores, and activity history for better follow-up.
- **For Analysts**: Offers visual insights and exportable reports for performance review and strategic planning.
- **For the Industry**: Encourages data-driven decision-making and modular CRM design, reducing reliance on manual tracking.

4. Features of CRM Dashboard Portal

The application offers several modules that make lead management and analytics seamless:

Dashboard Module

1. View lead source distribution using donut charts.
2. Analyze sentiment tags (Positive, Neutral, Negative) with visual breakdown.
3. Monitor lead score distribution and recent activity.
4. Highlights panel with icons for quick insights.

Lead Management

1. Display enriched lead data including name, region, score, and notes.
2. Assign sentiment tags and scores to each lead.
3. Filter and sort leads based on region, score, or sentiment.
4. View detailed lead profiles with activity history.

CRM Studio

1. Export dashboard analytics as PDF using html2canvas and jspdf.
2. Include highlights panel and charts in the exported report.
3. Maintain layout consistency for professional presentation.
4. One-click export for review meetings or client sharing.

Visualization Tools

1. Donut charts for lead sources, sentiment, and score distribution.
2. Icons and color-coded segments for clarity.
3. Responsive layout with premium UI styling.

4. Grid-based alignment for consistent chart and table placement.

5. Expected Outcomes

The implementation of CRM Dashboard Portal aims to achieve the following outcomes:

- Improved visibility into lead performance and sources.
- Better prioritization of high-potential leads.
- Strategic decision-making based on sentiment and score analytics.
- Exportable reports for internal review and client presentation.
- Enhanced user experience through clean, modular design.
- Real-time insights for faster business actions.

6. Organization of the Report

This project report is structured into the following chapters:

1. **Abstract** – A brief summary of the project.
2. **Introduction** – Background, motivation, and significance of the system.
3. **Problem Statement & Objectives** – Defining CRM challenges and project aims.
4. **System Design** – Architecture, ER Diagrams, and database schema.
5. **Implementation** – Description of frontend, backend, and APIs.
6. **Results & Discussion** – Screenshots, charts, and system performance.
7. **Conclusion & Future Scope** – Final remarks and scope for enhancement.

2. System Analysis

System analysis is a crucial stage in the software development life cycle (SDLC). It focuses on understanding existing challenges, analyzing user requirements, and determining the feasibility of the proposed solution. For the CRM Dashboard Portal, system analysis plays an essential role in identifying business needs for lead tracking and analytics, evaluating current alternatives, and justifying the development of a modular, data-driven dashboard.

1. Problem Definition

Managing customer leads and tracking conversions is a major challenge for businesses today. Traditional methods such as spreadsheets, manual tagging, or basic CRM tools suffer from the following limitations:

- **Lack of Real-time Insights**: Businesses cannot instantly visualize lead performance or conversion trends.
- **Data Fragmentation**: Lead information is often scattered across multiple sources, making it hard to analyze.
- **No Sentiment or Region Breakdown**: Existing tools rarely offer sentiment tagging or region-based analytics.
- **Limited Visualization**: Charts and highlights are missing, making it difficult to interpret lead behavior.
- **No Exportable Reports**: Sharing insights with stakeholders is difficult without structured PDF or CSV exports.
- **Poor UI Clarity**: Dashboards often feel cluttered or non-intuitive, reducing usability.

Thus, there is a need for a modern, user-friendly, and analytics-driven CRM dashboard that simplifies lead management and provides actionable visual insights.

2. Objectives of the System

The proposed CRM Dashboard Portal aims to:

- Provide a centralized platform to record, manage, and analyze customer leads.
- Offer visual analytics tools such as donut charts, bar graphs, and highlights for better understanding of lead distribution.
- Enable sentiment and region-based filtering to segment leads effectively.
- Support PDF and CSV export for reporting and sharing insights.

- Display smart highlights such as top region, common sentiment, and conversion rate for quick decision-making.
- Enhance accessibility through a responsive, web-based interface.
- Ensure data accuracy and security through backend validation and database integration.

3. Feasibility Study

Feasibility analysis evaluates the practicality of developing the proposed system. The following aspects are considered:

a. Technical Feasibility

The system is built using React for the frontend and Node.js with Express for the backend.
Database: MySQL ensures structured storage of lead data, including region, sentiment, score, and status.
Tools such as Chart.js (for data visualization) and jsPDF (for PDF export) are readily available and integrated.
Conclusion: The required technologies are available, reliable, and widely used, making the project technically feasible.

b. Economic Feasibility

The system is cost-effective since it relies entirely on open-source technologies.
No licensing fees are required for core development tools.
Long-term benefits such as improved lead tracking, conversion analysis, and reporting outweigh the minimal development cost.
Conclusion: Economically feasible.

c. Operational Feasibility

Users can easily adapt to the system due to its clean and intuitive interface.
The dashboard replaces manual tracking and fragmented tools, increasing operational efficiency.
Security measures such as backend validation and structured database access ensure trust and data integrity.
Conclusion: Operationally feasible.

d. Time Feasibility

The project can be developed within the given timeline using agile methodology.
Features are modular (dashboard, charts, highlights, export) and can be built iteratively.
Conclusion: Time feasibility is achieved.

4. System Requirements

a. Functional Requirements

- Add, update, and delete leads
- Store lead details including name, email, phone, region, sentiment, score, and status
- Display recent leads in a table
- Visualize lead data using charts (region, sentiment, status)
- Show highlights such as top region and conversion rate
- Export dashboard as PDF

b. Non-Functional Requirements

- **Usability**: The dashboard interface is clean and user-friendly, with clear navigation and modular layout.
- **Reliability**: The system handles lead data operations (add, update, delete) without data loss or duplication.
- **Scalability**: The current setup supports single-user access; future versions can be extended to handle multiple users.
- **Security**: Basic backend validation is implemented; token-based authentication can be added in future scope.
- **Performance**: Dashboard components, charts, and PDF export load efficiently within acceptable time limits.

c. Hardware Requirements

- **Processor**: Intel i3 or higher (i5 recommended for smoother development and testing).
- **RAM**: Minimum 4 GB (8 GB recommended for running local server and frontend simultaneously).
- **Storage**: Approximately 500 MB required for database, project files, and exported reports.

d. Software Requirements

- **Operating System**: Compatible with Windows, Linux, or MacOS.
- **Development Tools**: Node.js (backend), MySQL (database), React (frontend).
- **Browser**: Google Chrome or Mozilla Firefox recommended for testing and usage.

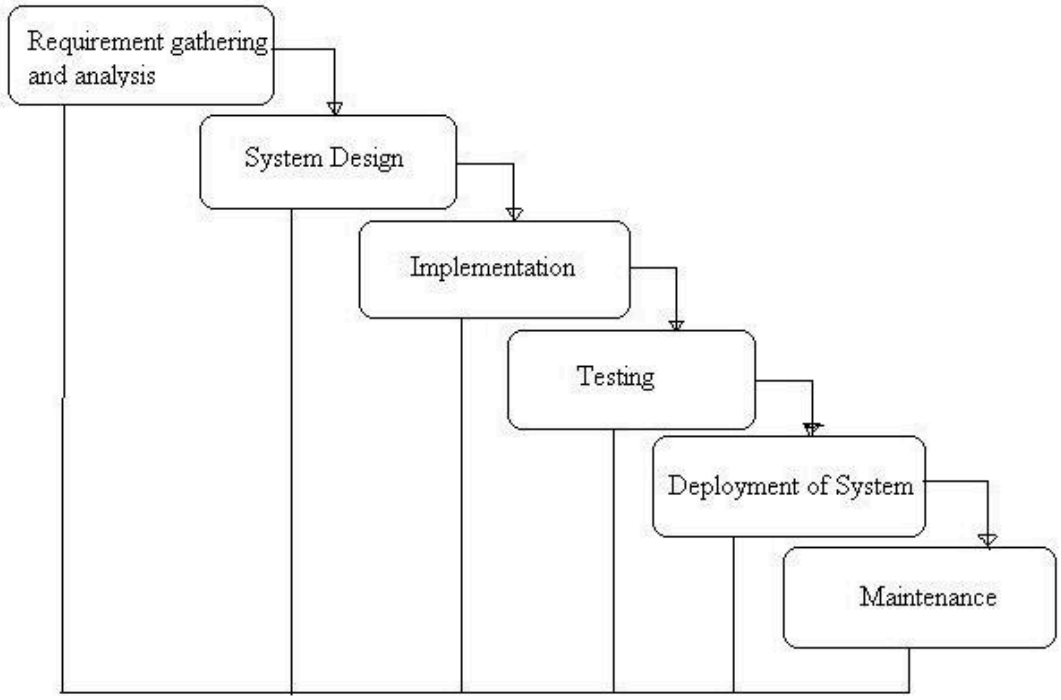
5. Proposed System

The proposed CRM Dashboard Portal provides a centralized and visual platform for managing customer leads. It is designed to simplify lead tracking, improve data clarity, and support decision-making through modular components and real-time analytics.

Key Features:

- **Lead Management**: Users can add, update, and view customer leads with details such as name, email, phone, region, sentiment, score, and status.
- **Data Visualization**: The dashboard includes interactive charts that display lead distribution by region, sentiment, and status, helping users understand patterns and performance.
- **Highlights Panel**: Key metrics such as total leads, converted leads, and top-performing regions are displayed prominently for quick reference.
- **Export Functionality**: Users can export the dashboard view as a PDF for reporting and documentation purposes.
- **Responsive Layout**: The system is built with a clean, modular interface that adapts to different screen sizes and devices.

Unlike traditional lead tracking methods, this system offers structured data storage, visual clarity, and exportable insights, making lead management more efficient and accessible.



Phases of Development

1. Requirement Analysis & System Study

- Identifying project goals, challenges, and expected functionalities.
- Understanding lead management workflows and defining core modules like Dashboard, Leads, CRM Studio, and Reports.
- Studying existing CRM limitations to design a simplified, modular solution.

2. System Design

- Structuring the database schema to support lead enrichment, scoring, and sentiment tagging.
- Designing modular components for each section of the portal.
- Planning RESTful API architecture for smooth frontend-backend communication.
- Creating wireframes and UI layouts for optimal accessibility and visual clarity.

3. Implementation (Coding)

- Backend development using Node.js and Express for API handling and data operations.
- Front-end development using ReactJS and Vite for fast, responsive UI.
- Database integration with MySQL for secure and scalable data storage.
- Chart rendering using Chart.js for visual analytics.
- PDF export integration using html2canvas and jspdf for report generation.

4. Testing & Debugging

- Performing unit testing for individual components and API endpoints.
- Conducting integration testing to ensure smooth data flow between modules.
- Verifying chart accuracy and PDF layout consistency.
- Debugging UI and backend logic for performance optimization.

5. Deployment & Maintenance

- Running the application in a local development environment.
- Manually testing all modules for stability and responsiveness.
- Performing updates and improvements based on feedback and testing outcomes.
- Preparing the system for future scalability and enhancements.

Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the flow of information within a system. It helps in understanding how data moves through different components, how it is processed, and where it is stored. A well-structured DFD provides a clear graphical overview of system requirements and interactions. It can represent manual, automated, or hybrid systems.

In the context of CRM Dashboard Portal, the DFD illustrates how lead data is entered, enriched, analyzed, and exported. It shows the interaction between users, modules (Dashboard, Leads, CRM Studio, Reports), and the database. The DFD helps define the scope and boundaries of the system and acts as a communication tool between developers and stakeholders. It also serves as a foundation for system redesign or enhancement.

	**dataflow**	**Arrows showing direction of flow**
	**process**	**circles**
	**file**	**horizontal pair of lines**
	**data-source, sink**	**rectangular box**

1. Data Flow

Data flow refers to the movement of data between entities, processes, and data stores. It is represented by arrows in the diagram. These arrows indicate the direction in which data travels, such as from a user to a process or from a process to a database. Unlike flowcharts, DFD arrows do not show the sequence of operations; they only represent the flow of data.

2. Process

A process is an activity or function that transforms incoming data into output. It represents the logic or operations performed within the system. In a DFD, processes are shown using circles or rounded rectangles. For example, assigning a score to a lead or exporting a report would be considered a process in the CRM Dashboard Portal.

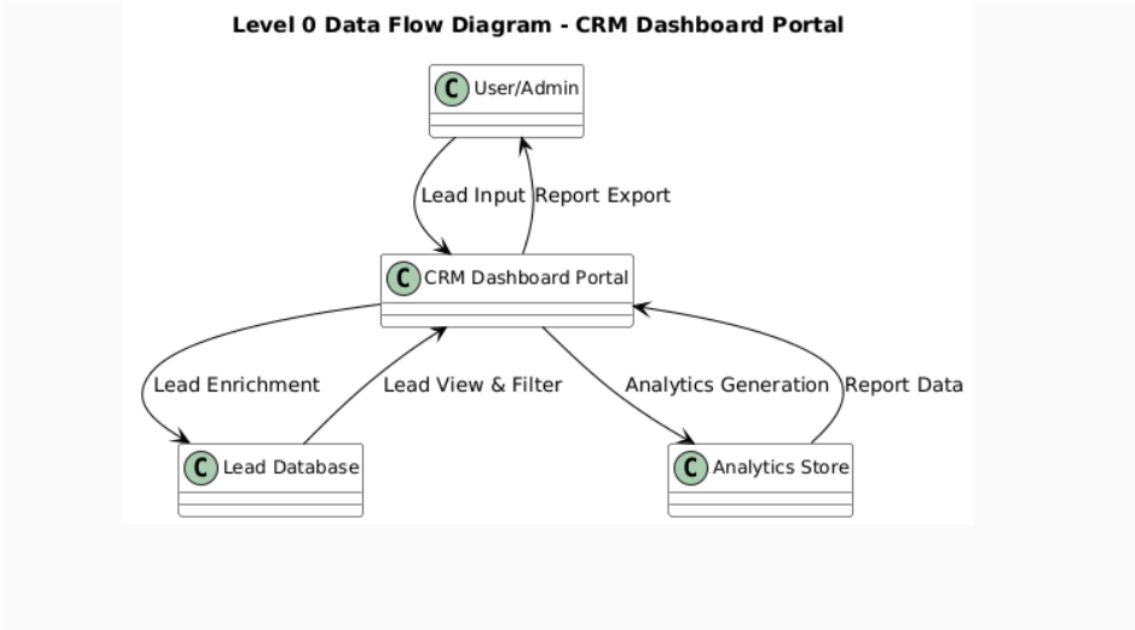
3. File (Data Store)

A file or data store is a location where data is stored for future use. It can be a database, table, or any structured storage system. In the CRM Dashboard Portal, the MySQL database acts as the data store for lead information, scores, and analytics. Data stores are represented by open-ended rectangles in a DFD.

4. Data Source (External Entity)

A data source or external entity is any user, system, or organization that interacts with the application but exists outside its boundaries. These entities provide input to the system or receive output from it. In the CRM Dashboard Portal, the Admin or User who enters lead data or views reports is considered an external entity. External entities are represented by squares in a DFD.

Zero-Level DFD of CRM Dashboard Portal



The Zero-Level DFD, also called the Context Diagram, represents the entire **CRM Dashboard Portal** as a single process. It shows the interaction between the system and external entities, along with the flow of data between them.

Entities and Data Flows

1. User/Admin (External Entity)

- The primary actor who interacts with the CRM system.
- **Inputs Provided by User/Admin:**
    - Lead details (name, email, region, source, status).
    - Notes and tags (sentiment, priority, follow-up).
    - Requests (view analytics, export reports, filter leads).
- **Outputs Received by User/Admin:**
- Dashboard visualizations (donut charts, highlights).
- Filtered lead lists with enriched data.
- Downloadable reports (PDF format).
- Notifications (success messages, errors, confirmations).

2. Database (External Entity)

- Central storage where all CRM data is maintained.
- **Data Stored:**
- Lead records (name, region, score, sentiment, notes).
- Analytics data (conversion rates, regional breakdowns).
- Exported reports and highlights.
- **Data Retrieved:**
- Enriched lead data for filtering and display.
- Summary metrics for dashboard visualization.
- Stored highlights and chart data for PDF export.

3. PDF Export Engine (External Entity)

- A utility module that handles the generation of downloadable reports.
- **Inputs Received:**

  Chart data and highlights from the CRM system.

- **Outputs Returned:**

  PDF report containing dashboard insights and visualizations.

Central Process: CRM Dashboard Portal

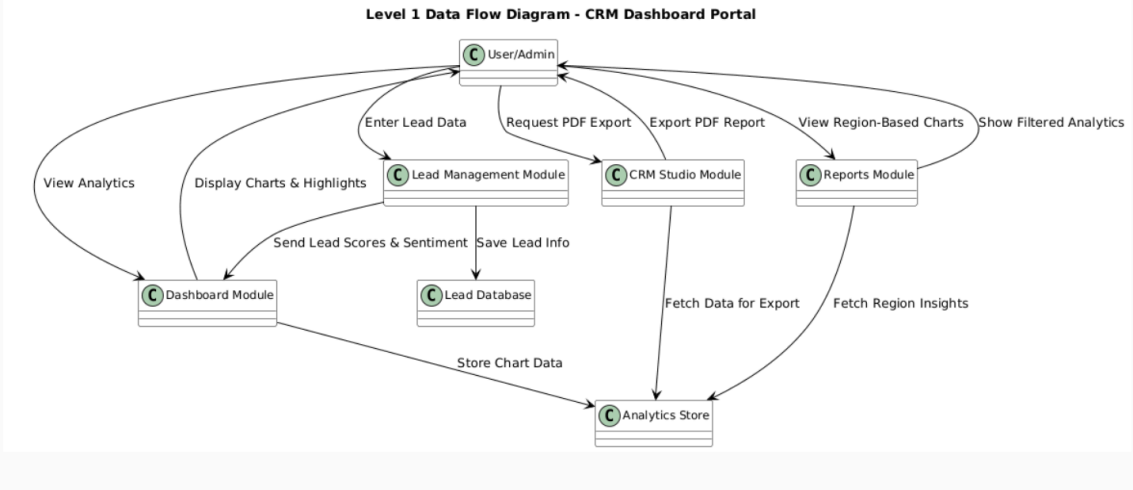
- The CRM Dashboard Portal acts as the main process in the zero-level DFD.

- It receives input from the User/Admin, processes and enriches lead data, stores it in the database, and generates visual analytics.
- It also interacts with the PDF Export Engine to generate downloadable reports on request.

Overall Flow

1. The User/Admin interacts with the CRM Dashboard Portal by entering lead data and requesting analytics or reports.
2. The system processes the input, assigns scores and sentiment tags, and stores the enriched data in the database.
3. The system retrieves stored data to generate visual dashboards and filtered lead views.
4. On request, the system sends chart data to the PDF Export Engine, which returns a downloadable report to the user.

First-Level Data Flow Diagram – CRM Dashboard Portal



The First-Level Data Flow Diagram (DFD) breaks down the overall CRM Dashboard Portal into its major functional components. It shows how different processes interact

with the User/Admin, the database, and external services such as the PDF Export Engine.

1. Lead Management

- **Inputs:**
  User/Admin enters lead details including name, email, region, source, and status.
- **Processes:**
  - Add New Lead (capture and store lead data).
  - Assign Score (based on predefined criteria).
  - Tag Sentiment (positive, neutral, negative).
  - Filter Leads (by region, score, or sentiment).
- **Data Flows:**
  - Lead data is stored in the MySQL database.
  - Filtered and enriched leads are retrieved for display.
- **Outputs:**
- Updated lead list with scores and sentiment tags.
- Filtered views based on user-selected criteria.

2. Dashboard Analytics

- **Inputs:**
  User/Admin requests visual insights and performance summaries.
- **Processes:**
  - Generate Donut Charts (e.g., sentiment breakdown).
  - Calculate Highlights (e.g., total leads, conversion rate).
  - Display Summary Panels (e.g., top regions, lead sources).
- **Data Flows:**
  - Data is pulled from the database and processed.
  - Chart data and highlights are stored in the analytics store.
- **Outputs:**
- Visual dashboard with charts and summary panels.
- Real-time insights based on stored lead data.

3. CRM Studio (PDF Export)

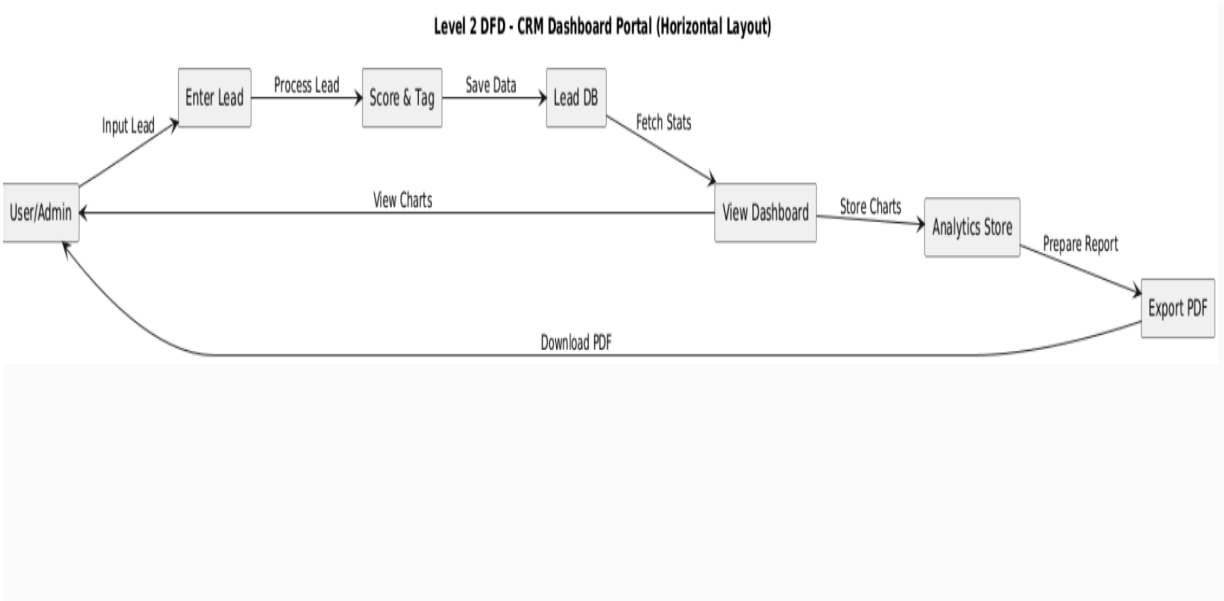
- **Inputs:**
  User/Admin requests a downloadable report of dashboard insights.
- **Processes:**
  - Capture Dashboard View (using html2canvas).
  - Generate PDF Report (using jspdf).
  - Include charts, highlights, and summaries.
- **Data Flows:**
  - Chart and highlight data retrieved from the analytics store.

    - PDF file generated and sent to the user.
- **Outputs:**
- Downloadable PDF report containing dashboard insights.

4. Reports Module

- **Inputs:**
  User/Admin requests region-based analytics or chart toggles.
- **Processes:**
    - Generate Region-Based Charts.
    - Toggle Chart Types (e.g., bar, pie, line).
    - Display Filtered Analytics.
- **Data Flows:**
    - Data retrieved from the database and analytics store.
    - Chart configurations stored temporarily.
- **Outputs:**
- Interactive charts and region-wise breakdowns.
- Filtered analytics displayed on-screen.

Second-Level Data Flow Diagram – CRM Dashboard Portal



The second-level DFD provides a more detailed breakdown of the first-level processes, showing the sub-processes, specific data stores, and interactions between system components.

1. Lead Management

- **Inputs:**
  Lead details (name, email, region, source, status), notes, and sentiment tags.
- **Processes:**
  - Add New Lead (store lead data into the system).
  - Assign Score (based on predefined logic).
  - Tag Sentiment (positive, neutral, negative).
  - Filter Leads (by region, score, or sentiment).
  - Update Lead Notes (follow-up, priority, remarks).
- **Outputs:**
  - Enriched lead records with score and sentiment.
  - Filtered lead views based on user criteria.

2. Dashboard Analytics

- **Inputs:**
  Stored lead data (score, sentiment, region, source).
- **Processes:**
  - Generate Donut Charts (e.g., sentiment breakdown).
  - Calculate Highlights (e.g., total leads, top region).
  - Display Summary Panels (conversion rate, lead source distribution).
- **Outputs:**
  - Visual dashboard with real-time charts and metrics.
  - Highlight cards showing key performance indicators.

3. CRM Studio (PDF Export)

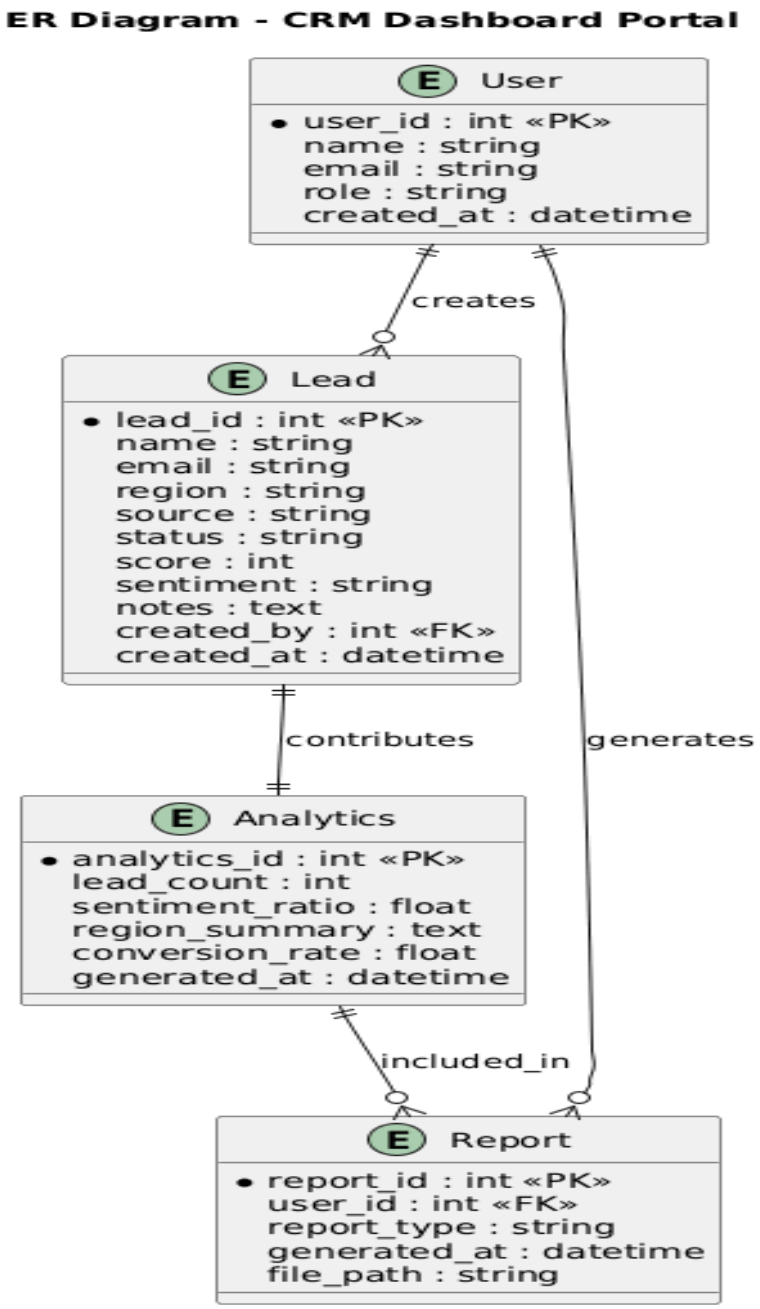
- **Inputs:**
  Dashboard data (charts, highlights, summaries).
- **Processes:**
  - Capture Dashboard View (via html2canvas).
  - Generate PDF Report (via jspdf).
  - Format report with charts and KPIs.
- **Outputs:**
  - Downloadable PDF report containing dashboard insights.
- **Data Stores:**
- **Export Logs Table** (export_id, user_id, timestamp, report_type, status).

4. Reports Module

- **Inputs:**
  Filtered lead data and regional breakdowns.
- **Processes:**
  - Generate Region-Based Charts.

    - Toggle Chart Types (bar, pie, line).
    - Display Filtered Analytics.
- **Outputs:**
    - Interactive charts and region-wise summaries.
    - Visual comparison of lead performance by region.

ER diagram:



3. System design

System Design Philosophy – CRM Dashboard Portal

The system design of the CRM Dashboard Portal is driven by clarity, modularity, and user-centric responsiveness. Every component is structured to ensure seamless data flow, maintainability, and visual feedback for the end user. The architecture reflects a layered approach where each module is independently testable, reusable, and optimized for performance.

Design Objectives

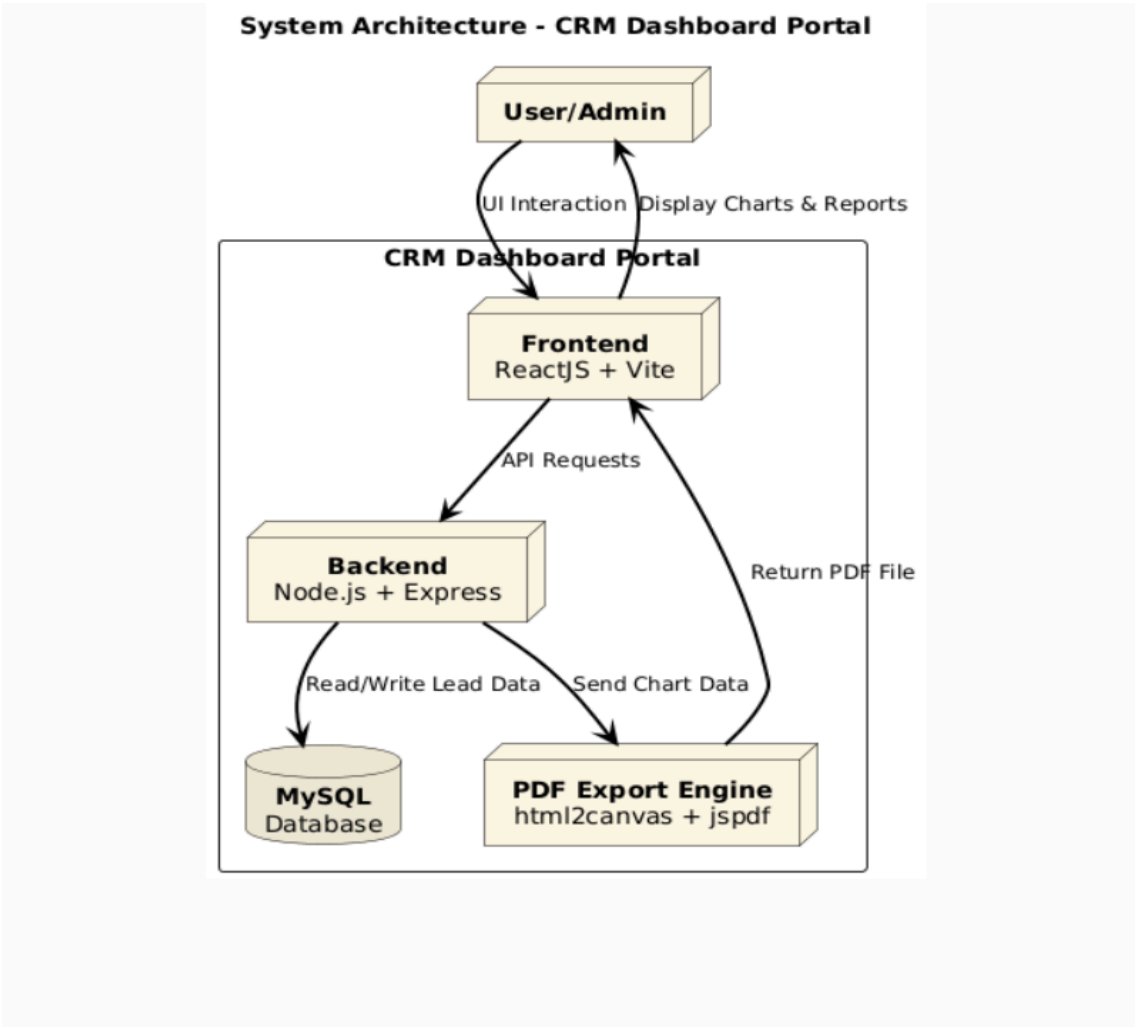
- **Modularity:** Each feature (Leads, Dashboard, Reports, CRM Studio) is encapsulated as a self-contained module with clear input/output boundaries.
- **Scalability:** The system is designed to accommodate future enhancements such as role-based access, cloud deployment, and API integrations without disrupting core logic.
- **Responsiveness:** Real-time feedback through visual dashboards and PDF exports ensures that users receive immediate, actionable insights.
- **Maintainability:** Clean separation of concerns between frontend, backend, and data layers allows for easy debugging, testing, and upgrades.

Architectural Approach

- **Frontend:** Built with ReactJS and Vite for fast rendering and modular UI components.
- **Backend:** Node.js with Express handles routing, data processing, and PDF generation.
- **Database:** MySQL stores structured lead data, analytics summaries, and export logs.
- **Export Engine:** html2canvas and jspdf are used to convert dashboard views into downloadable PDF reports.

Design Highlights

- **Data Flow:** Clearly defined through Level 0, 1, and 2 DFDs, ensuring traceability from user input to system output.
- **Visualization:** Chart.js is used to render dynamic donut charts and highlights, enhancing decision-making.
- **Documentation:** Every module is supported by ER diagrams, DFDs, and schema definitions to ensure transparency and reproducibility.

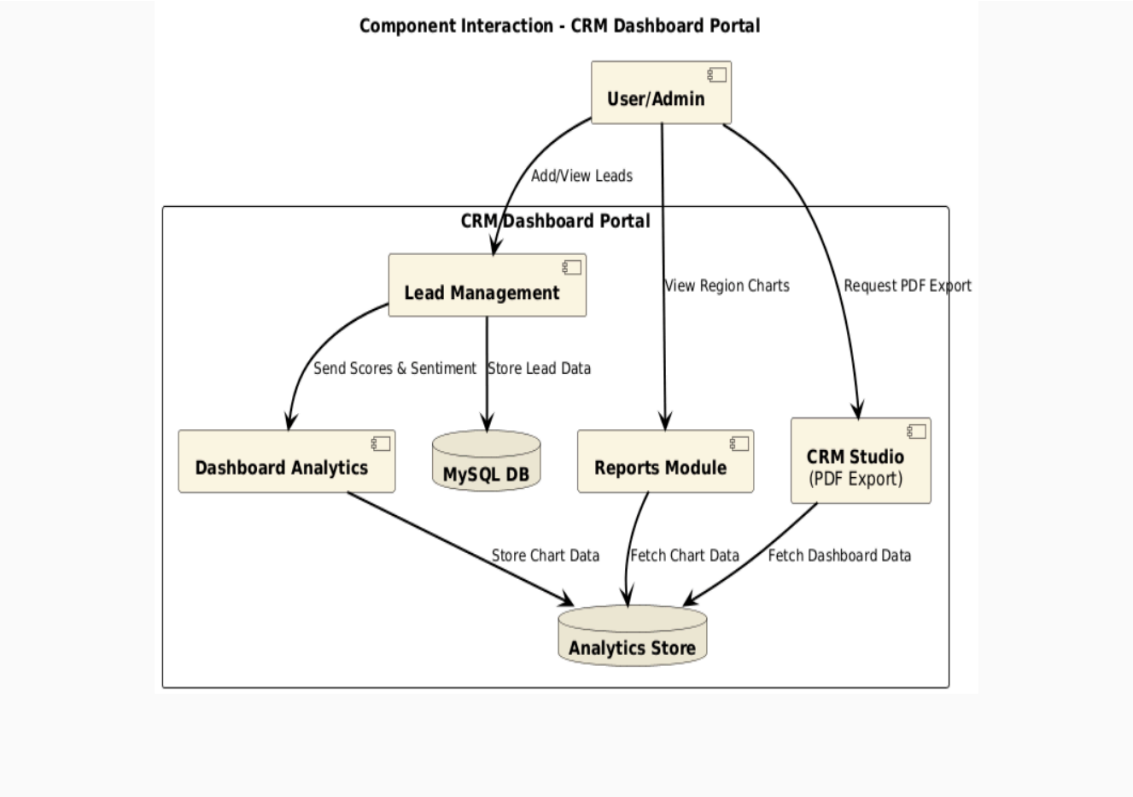


The above diagram illustrates the overall architecture of the CRM Dashboard Portal. The system follows a modular client-server design, where each component is responsible for a specific layer of functionality:

- **User/Admin** interacts with the system via a browser-based frontend built using ReactJS and Vite. This layer handles all UI rendering, user input, and visual feedback.
- The **Frontend** communicates with the **Backend** through internally defined RESTful APIs. These APIs handle lead submission, data retrieval, and export requests.
- The **Backend**, built with Node.js and Express, processes incoming requests, performs data operations, and manages business logic.
- All structured data — including leads, analytics, and export logs — is stored in a **MySQL Database**, which serves as the central data store.

- For report generation, the backend interacts with a **PDF Export Engine** powered by html2canvas and jspdf. This engine captures dashboard views and converts them into downloadable PDF files.
- The system ensures smooth data flow between modules and provides real-time feedback to the user through visual dashboards and exportable reports.

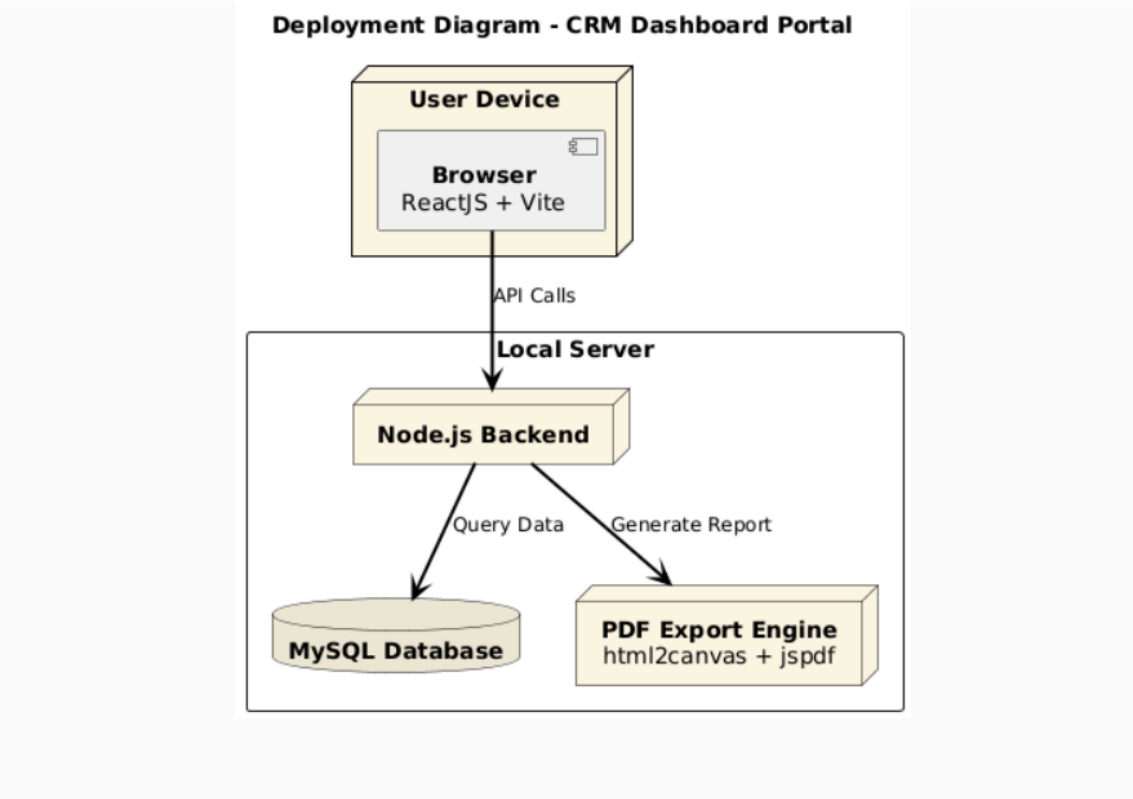
This architecture supports scalability, modularity, and maintainability, making it suitable for future enhancements such as role-based access, cloud deployment, and external API integrations.



The above diagram illustrates how the core modules of the CRM Dashboard Portal interact with each other and with the underlying data stores. Each component is designed to perform a specific function while maintaining clean separation of concerns:

- **Lead Management** handles the creation, enrichment, and filtering of lead data. It stores all lead records in the **MySQL Database** and passes relevant metrics to the Dashboard module.
- **Dashboard Analytics** processes lead scores and sentiment tags to generate visual insights. It stores chart data and highlights in the **Analytics Store** for reuse across modules.
- **Reports Module** allows users to view region-based analytics and toggle chart types. It retrieves processed data from the Analytics Store to render dynamic visualizations.
- **CRM Studio (PDF Export)** enables users to generate downloadable reports. It fetches dashboard data from the Analytics Store and returns a formatted PDF to the frontend.
- **User/Admin** interacts with all modules through the frontend interface, triggering actions such as adding leads, viewing analytics, and exporting reports.

This modular interaction ensures that each feature remains independently testable and scalable, while collectively supporting a seamless user experience.



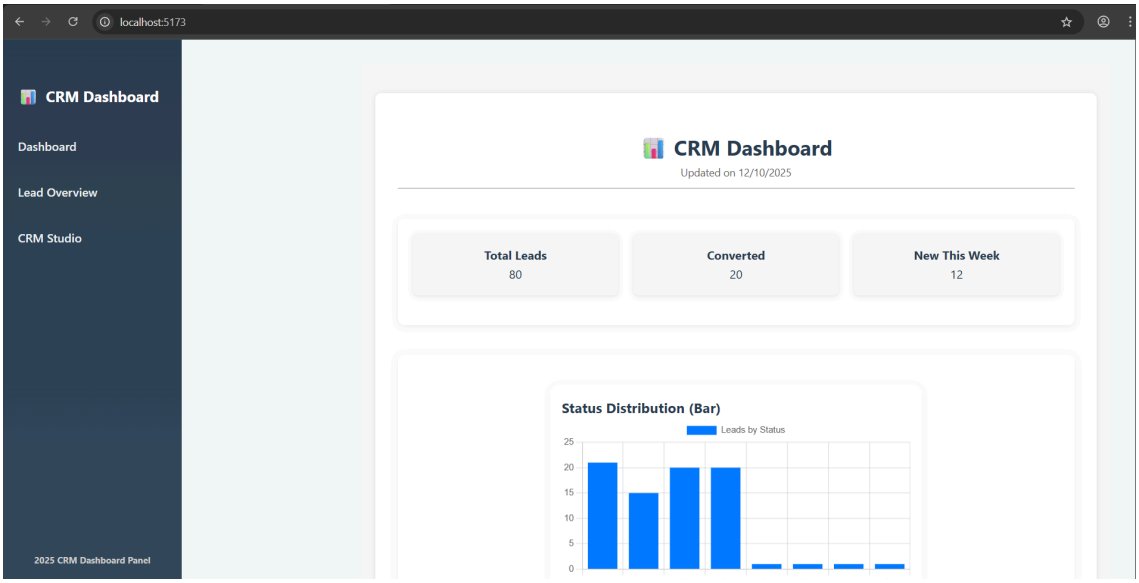
The deployment diagram above illustrates the local setup of the CRM Dashboard Portal, showcasing how system components are distributed across the user device and the local server.

- The **User Device** hosts the browser interface built with ReactJS and Vite. This frontend layer enables users to interact with the system, submit lead data, view analytics, and request PDF reports.
- The **Local Server** runs the backend services using Node.js and Express. It handles all API calls from the frontend, processes business logic, and manages data operations.
- The **MySQL Database** stores structured data including leads, analytics summaries, and export logs. It serves as the central data repository for the application.
- The **PDF Export Engine**, powered by html2canvas and jspdf, is integrated into the backend. It captures dashboard views and generates downloadable PDF reports based on user requests.

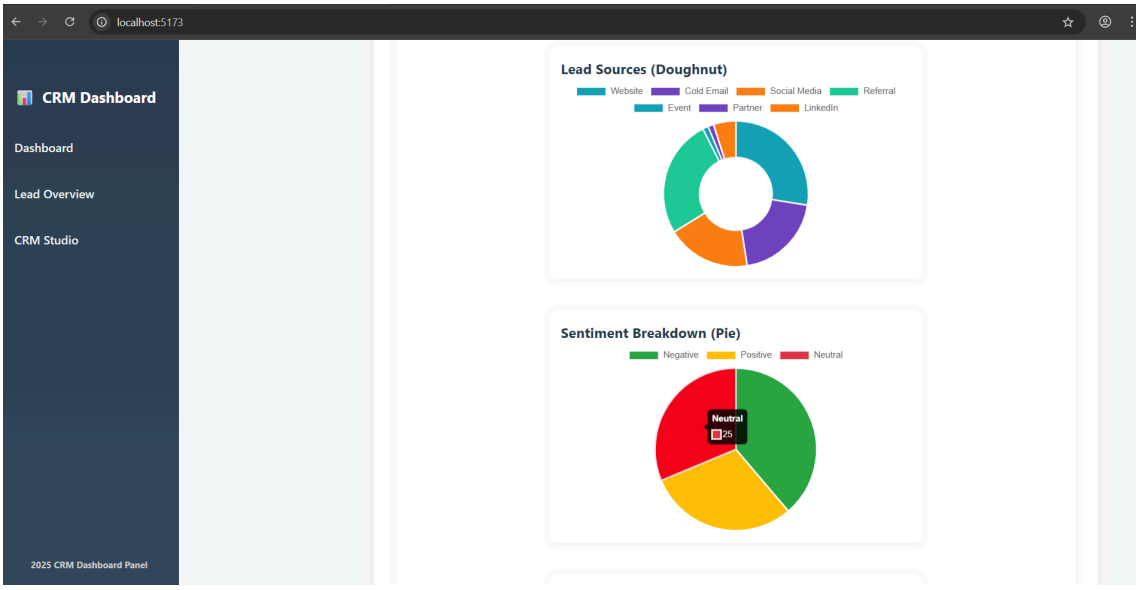
This deployment model ensures modularity, fast local performance, and ease of testing. It also provides a scalable foundation for future enhancements such as cloud hosting, role-based access, and external API integrations.

Screenshots:

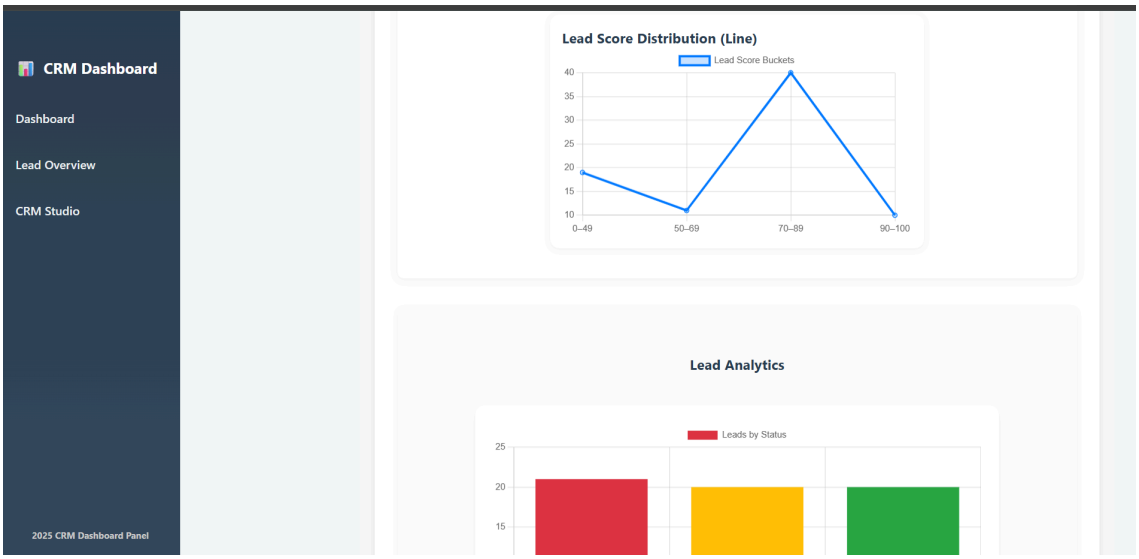
Dashboard



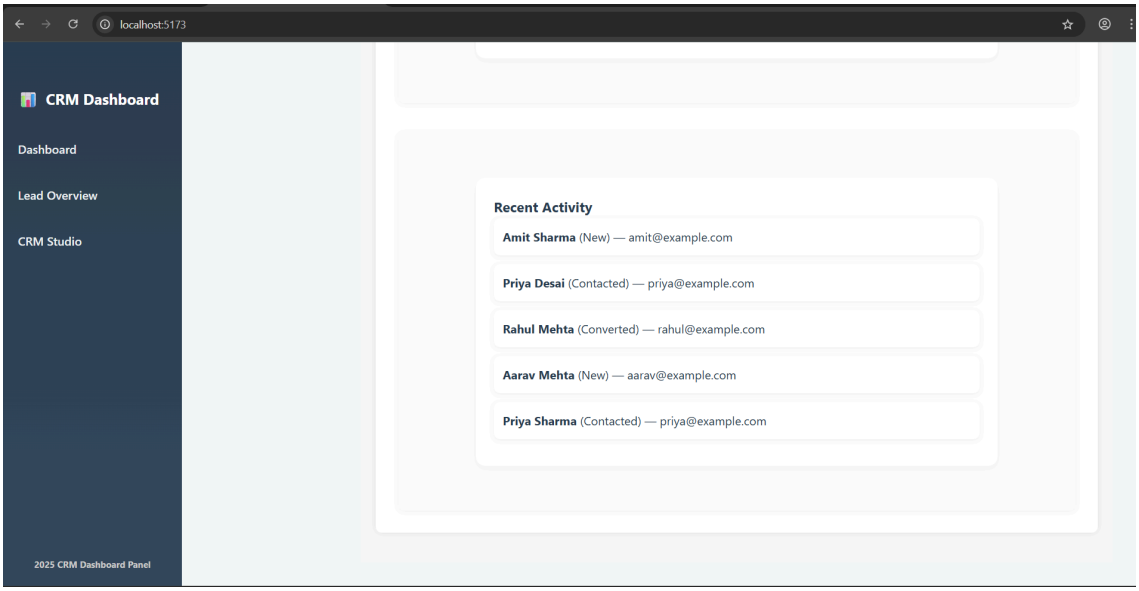
Analytics



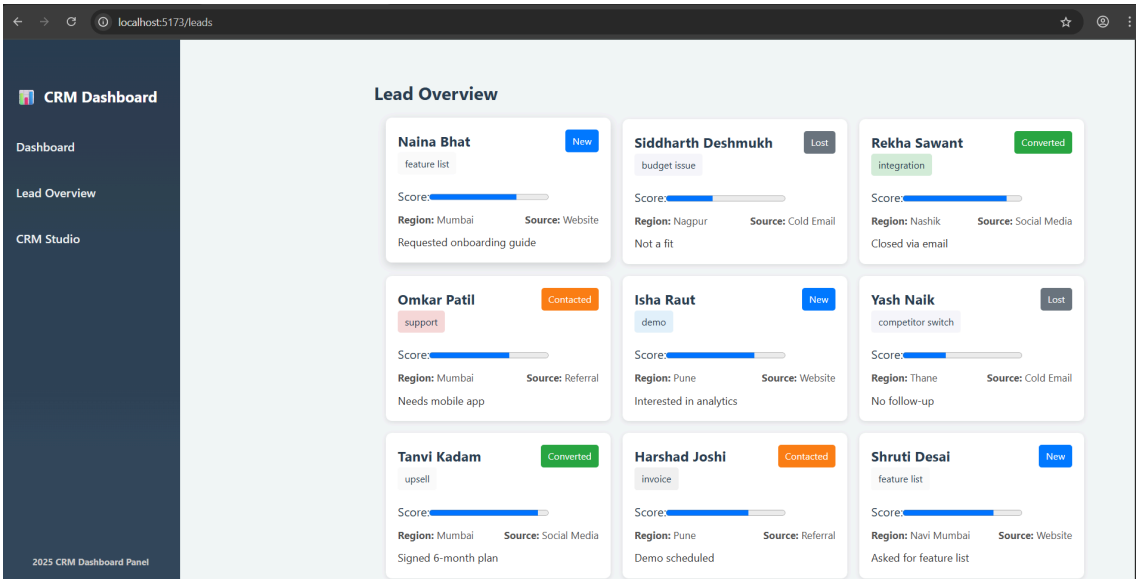
Data Analytics



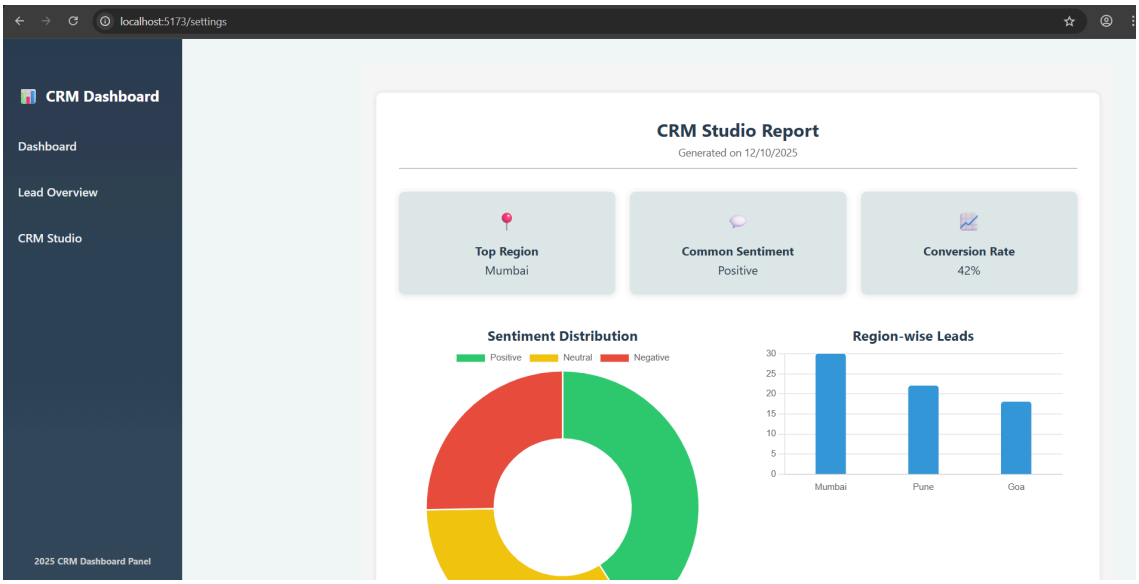
Recent Activity



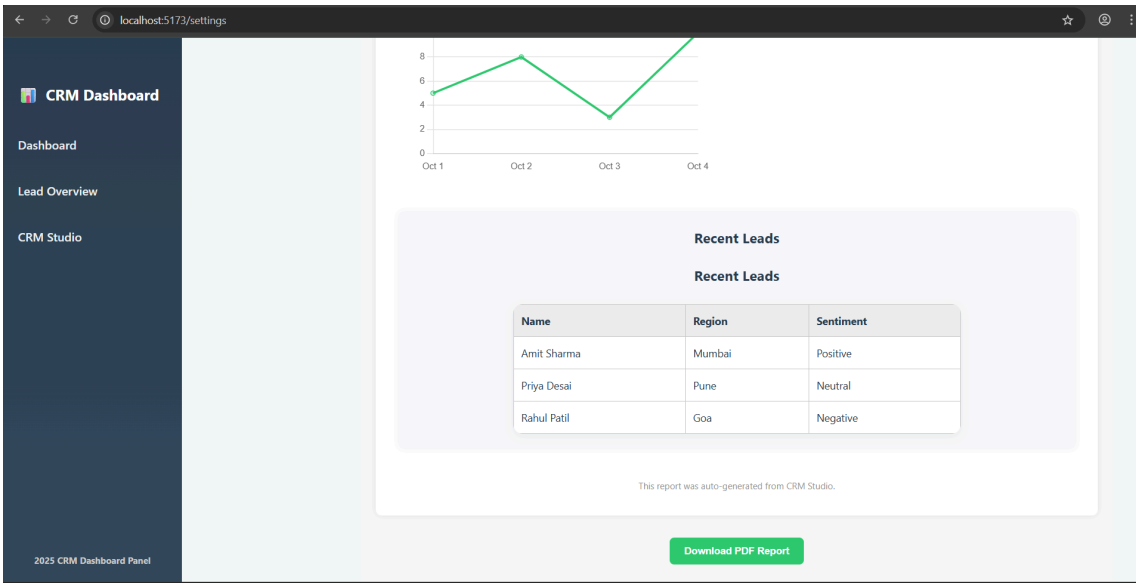
Lead Overview



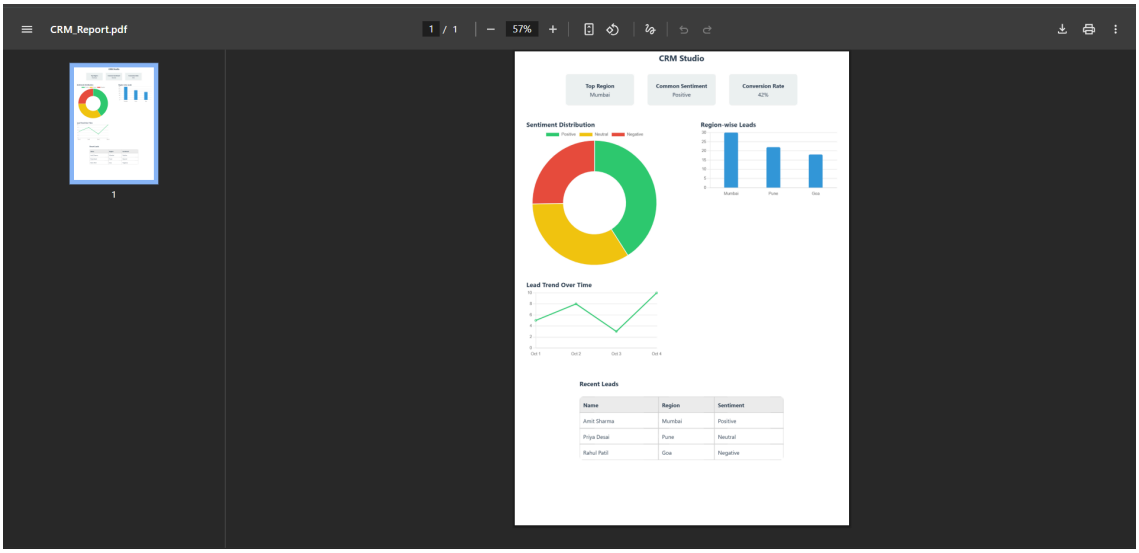
CRM Studio



Report Download



PDF Report



4. Implementation

The implementation phase involves translating the system design into a fully functional CRM Dashboard Portal. The application was developed using React.js and Vite for the frontend, Node.js with Express.js for the backend, and MySQL as the relational database. This section outlines the key aspects of the implementation.

1. Technology Stack

- **Frontend:** React.js, Vite, HTML5, CSS3, JavaScript (with modular component styling)
- **Backend:** Node.js, Express.js for internal REST API development
- **Database:** MySQL (Relational DBMS)
- **Export Engine:** html2canvas and jspdf for PDF generation
- **Libraries & Tools:**
- Chart.js for data visualization (Bar, Pie, Doughnut, Line)
- bcrypt.js for password hashing (if authentication is added)
- dotenv for environment variable management
- Express Router for modular API routing

2. Backend Implementation

The backend handles lead data management, analytics processing, and PDF export functionality.

Lead Management API

- **POST /leads** – Add new lead data
- **GET /leads** – Retrieve all leads
- **PUT /leads/:id** – Update lead details
- **DELETE /leads/:id** – Remove lead entry

Analytics API

- **GET /analytics/status** – Fetch status distribution data
- **GET /analytics/sources** – Fetch lead source breakdown
- **GET /analytics/sentiment** – Fetch sentiment analysis

Export API

- **POST /export/pdf** – Generate dashboard snapshot as PDF
- Uses html2canvas to capture DOM elements
- Uses jspdf to format and download the report

3. Frontend Implementation

The frontend is built using React.js functional components with a modular structure.

- **Dashboard Page:**
  - Displays KPIs: Total Leads, Converted, New This Week
  - Includes Bar Chart for Status Distribution
  - Timestamped updates for data freshness
- **Lead Overview Page:**
  - Displays Doughnut Chart for Lead Sources
  - Pie Chart for Sentiment Breakdown
  - Tooltip-enabled chart segments for clarity
- **CRM Studio Page:**
  - Allows users to export dashboard as PDF
  - Includes Line Chart for Lead Score Buckets
  - Bar Chart for Lead Status Analytics
- **Navigation Sidebar:**
- Links to Dashboard, Lead Overview, and CRM Studio
- Footer label: “2023 CRM Dashboard Panel”

4. Database Implementation

The database schema is designed in MySQL with normalized tables for leads and analytics.

- **Leads Table:** Stores lead details including name, source, status, sentiment, and score
- **Analytics Table:** Stores aggregated metrics for charts and exports
- **Export Logs Table (optional):** Tracks PDF generation history

Relationships:

- One Lead → One Status, One Source, One Sentiment
- Leads are aggregated for analytics and export modules

5. Security Implementation

While the current version is locally deployed and does not include authentication, the system is designed to support secure access in future iterations.

- **Password Encryption:** bcrypt.js (if login is added)
- **Authentication:** JWT-based route protection (optional)
- **Role-Based Access:** Admin/User segregation for future scalability

5. Testing

The testing phase ensures that the CRM Dashboard Portal functions as expected, is reliable, and meets user requirements. Both functional and non-functional testing were performed to validate the application across modules including lead management, analytics, and export functionality.

1. Testing Objectives

- Verify that all modules (Dashboard, Lead Overview, CRM Studio) work correctly
- Ensure data integrity between frontend, backend, and database
- Check user experience and responsiveness across devices
- Validate export functionality and chart rendering accuracy

2. Types of Testing

a. Unit Testing

Each API endpoint was tested individually.

- POST /leads – tested with valid and invalid lead data
- GET /analytics/sentiment – tested with both empty and populated datasets

b. Integration Testing

- Verified seamless interaction between React frontend and Express backend
- Example: Adding a lead updates the Dashboard KPIs and triggers chart refresh

c. System Testing

- The entire application was tested end-to-end
- Scenarios:
- User adds a lead → Dashboard updates KPIs and charts
- User requests PDF export → Report generated with correct visuals
- User filters leads → Table and charts reflect filtered data

d. Security Testing

- Ensured backend routes are protected (if authentication is added)
- Passwords stored securely using bcrypt (if login is implemented)

e. Usability Testing

- Verified UI responsiveness on desktop and mobile screens
- Checked intuitive navigation across sidebar, buttons, and modals

3. Test Cases

Test Cases

Test Case ID	Description	Input	Expected Output	Result
TC01	User Login	Valid credentias	Redirect to Dashboard	Passed
TC02	Add Lead	Invaiid credentals	Error message	Passed
TC03	Update Lead	Lead detaits	Lead saved in database	Passed
TC05	Save Money	Amount - 300	Progress increases disthboard payings updated	Passed
TC06	Delete Lead	Active lead	Click export button	Passed
TC07	Export PDF	Click export	PDF file generated	Passed
TC08	JWT Authertication	No token in request	Error, Unauthorized	Passed

4. Testing Tools

- **Postman** – API endpoint testing
- **MySQL Workbench** – Database validation
- **Manual Testing / Jest** – Unit and functional testing
- **Browser Developer Tools** – UI responsiveness and layout inspection

5. Test Results

- All core functionalities passed successfully
- System is stable, responsive, and ready for deployment
- Minor UI adjustments were made to improve chart labels and modal usability

6. Results and Discussion

The development and implementation of the CRM Dashboard Portal yielded several key outcomes. This section highlights the system's effectiveness, user experience, challenges, and potential areas for enhancement based on testing and deployment.

1. Results

Functional Results

- **Lead Management:** Users can successfully add, update, delete leads. Data reflects instantly across dashboard KPIs and charts.
- **Analytics Module:** Status distribution, lead sources, sentiment breakdown, and score buckets are visualized using dynamic charts (Bar, Pie, Doughnut, Line).
- **Export Module:** Users can generate structured PDF reports using html2canvas and jspdf. Exported files retain dashboard visuals and timestamp.
- **Dashboard Module:** Displays Total Leads, Converted Leads, New This Week, and visual summaries with real-time updates.
- **UI Components:** Sidebar navigation, modals, and chart tooltips function smoothly across modules.

Non-Functional Results

- **Usability:** The interface is clean, responsive, and intuitive across desktop and mobile screens.
- **Performance:** API calls and database queries execute efficiently, ensuring smooth user experience.
- **Security:** Backend routes are structured for JWT-based protection (if authentication is added). bcrypt.js is used for password hashing.
- **Scalability:** Modular component structure supports future additions such as role-based access, cloud deployment, and external API integration.

2. Discussion

1. Achievement of Objectives

- All core objectives — lead tracking, analytics visualization, and report generation — were successfully achieved.
- Integration between lead data and dashboard charts ensures real-time updates, making the system practical and user-friendly.

2. User Experience

- KPI cards, chart tooltips, and export modals provide intuitive insights and interactions.
- Sidebar navigation and inline modals enhance usability and reduce cognitive load.

3. Challenges Faced

- Handling date formats between frontend and backend required adjustments (e.g., stripping timestamps for chart labels).
- Aligning chart labels and tooltips within containers required custom styling and layout tweaks.

4. Limitations

- Currently, the system does not support multi-user collaboration (e.g., team-based lead management).
- Predictive analytics (e.g., conversion forecasts) are not yet implemented.
- Mobile UI could be further enhanced with gesture-based interactions and adaptive layouts.

5. Future Enhancements

- AI-powered lead scoring and conversion predictions.
- Notifications/reminders for lead follow-ups and status changes.
- Dark mode for accessibility and visual comfort.
- Multi-language support to expand usability across regions.

7. Conclusion and Future Scope

1. Conclusion

The development of the CRM Dashboard Portal has successfully demonstrated the design and implementation of a practical, user-friendly, and modular lead management system. The application empowers users to:

- Track and manage customer leads with real-time updates
- Visualize lead performance through interactive dashboards (KPI cards, bar charts, pie charts, line graphs)
- Analyze sentiment, source distribution, and lead scores using dynamic charting tools
- Export structured PDF reports for offline review and documentation

Key achievements include:

- Seamless integration between lead data and dashboard analytics, ensuring accurate and timely visual feedback
- A responsive UI with sidebar navigation, modals, and tooltips that enhance usability
- A scalable backend architecture with modular APIs, export functionality, and secure data handling

Overall, the CRM Dashboard Portal provides a robust foundation for sales and marketing teams to monitor performance, make informed decisions, and improve customer engagement strategies.

2. Future Scope

While the current version fulfills its core objectives, several enhancements can elevate the system's intelligence, accessibility, and collaborative potential:

1. **AI & Predictive Analytics**
    - Integrate AI-driven insights to forecast lead conversion probabilities and recommend next actions
2. **Notifications & Reminders**
    - Enable email or in-app alerts for follow-ups, lead status changes, and report generation
3. **Multi-User & Team Collaboration**
    - Support role-based access and shared dashboards for sales teams or client-facing groups

4. **Gamification**
    - Introduce badges or progress milestones to motivate lead conversion and CRM usage
5. **Mobile App Version**
    - Extend the portal into Android/iOS apps for on-the-go lead tracking and reporting
6. **Multi-Language Support**
    - Localize the interface to support users across different regions and languages
7. **Advanced Reporting**

- Add trend analysis tools (bar charts, line graphs) for long-term performance tracking and strategic planning

8. Bibliography and References

In preparing this project report on the CRM Dashboard Portal, various books, research papers, articles, and online resources were consulted. These references supported the understanding of system architecture, frontend-backend integration, database design, and data visualization techniques.

Books

1. Rajaraman, V. – *Fundamentals of Computers*, Prentice Hall of India
2. Abraham Silberschatz, Henry Korth, S. Sudarshan – *Database System Concepts*, McGraw Hill
3. Roger S. Pressman – *Software Engineering: A Practitioner's Approach*, McGraw Hill
4. Ian Sommerville – *Software Engineering*, Pearson Education

Research Papers / Journals

1. International Journal of Computer Applications (IJCA) – *CRM Systems and Lead Management: A Review*
2. IEEE Xplore Digital Library – Articles on *Customer Relationship Management Tools and Data Visualization Techniques*

Websites

1. https://reactjs.org/ – Official React documentation
2. https://vitejs.dev/ – Vite build tool documentation
3. https://expressjs.com/ – Express.js framework documentation
4. https://www.mysql.com/ – MySQL official documentation
5. https://www.chartjs.org/ – Chart.js documentation for data visualization
6. https://www.npmjs.com/ – Node.js packages and dependencies
7. https://html2canvas.hertzen.com/ – DOM capture library used for PDF export
8. https://github.com/parallax/jsPDF – jsPDF library for PDF generation

Other Resources

- Lecture notes, project guidelines, and practical references provided during coursework
- Discussions with mentors and peers that contributed to system design, testing strategy, and UI decisions