Dashboard Feature

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the development of the **Dashboard Feature**. This feature will enable users to create, customize, manage, and share dashboards with multiple chart types and data sources. The system will support **real-time data updates, collaboration, role-based access control, and export functionality**.

1.2 Scope

The dashboard feature will allow users to:

- **Create, edit, delete, and duplicate dashboards**.
- **Add, configure, and manage multiple charts** within dashboards.
- **Integrate with external data sources** such as databases, APIs, and CSV files.
- **Provide real-time updates and interactive elements** for data visualization.
- **Enable collaboration**, allowing multiple users to work on the same dashboard with different permission levels.
- **Export dashboards** in various formats such as PDF, PNG, and CSV.
- **Ensure security and scalability** for handling large datasets and concurrent users.

1.3 Audience

This document is intended for:

- **Development Team** (Frontend & Backend Engineers)
- **Project Managers**
- **Product Owners & Stakeholders**
- **QA & Testing Teams**
- **UI/UX Designers**

2. Functional Requirements

2.1 Dashboard Management

2.1.1 Create Dashboard

- Users can create new dashboards by providing:
    - **Dashboard Name (Required)**
    - **Description (Optional)**
    - **Category/Tagging System** for organization (Optional)
    - **Default Theme Selection**

2.1.2 Edit Dashboard

- Users can update:
    - **Dashboard name, description, and tags**
    - **Theme, layout, and background settings**
    - **Grid settings (number of columns, margins, etc.)**

2.1.3 Delete Dashboard

- Users should be able to delete their dashboards.
- Provide a **confirmation prompt** before deletion.
- **Soft delete option** with an undo feature (dashboards remain recoverable for a set period).

2.1.4 Duplicate Dashboard

- Users can **clone an existing dashboard**, including all charts and settings.

2.2 Chart Management

2.2.1 Add Chart

- Users can add charts from the following types:
    - **Bar Chart**
    - **Line Chart**
    - **Pie Chart**
    - **Scatter Plot**
    - **Heatmap**
    - **Funnel Chart**
    - **Gauge Chart**
    - **Custom Charts (if applicable)**

2.2.2 Configure Chart

- Users can:
    - Select **data sources** (APIs, Databases, CSV, JSON, Excel)
    - Define **X-axis and Y-axis values**
    - Choose **color schemes and styles**
    - Enable **legends, labels, and gridlines**
    - Apply **filters and sorting**

2.2.3 Edit Chart

- Users can modify:
    - **Data source**
    - **Chart type and appearance**
    - **Chart labels and axis settings**

2.2.4 Delete Chart

- Users can remove charts from dashboards.
- Provide **confirmation before deletion**.

2.2.5 Resize & Rearrange Charts

- Support **drag-and-drop functionality** for resizing and repositioning charts within the dashboard grid.

2.2.6 Chart Limit

- Each dashboard can contain **up to 10 charts**.
- If the user tries to add more, an **error message should be displayed**.

2.3 Data Integration

2.3.1 Connect Data Sources

- Users can connect:
    - **Relational Databases (PostgreSQL, MySQL, SQL Server, etc.)**
    - **NoSQL Databases (MongoDB, Firebase, etc.)**
    - **CSV, JSON, Excel files**
    - **APIs (REST, GraphQL, WebSockets)**
    - **Cloud Storage (AWS S3, Google Drive, OneDrive, etc.)**

2.3.2 Real-time Data Updates

- Dashboards should support **real-time data streaming**.
- Users can configure:
    - **Auto-refresh intervals** (e.g., 1s, 5s, 10s, manual refresh)

2.3.3 Data Transformation

- Users can perform:
    - **Filtering** (e.g., date range, numeric filters)
    - **Sorting** (ascending/descending)
    - **Aggregation** (sum, average, min/max)

2.4 User Interaction & Customization

2.4.1 Dashboard Themes

- Users can apply **predefined themes** or customize:
    - **Colors**
    - **Fonts**
    - **Grid Layouts**

2.4.2 Interactive Elements

- Add features like:
    - **Hover tooltips**
    - **Drill-down on data points**
    - **Linked filtering between charts**

2.4.3 Save & Share Dashboards

- Users can:
    - **Save dashboards for personal use**
    - **Share dashboards with other users**
    - **Generate a public link for external sharing**

2.4.4 Export Dashboards

- Users can export dashboards as:
    - **PDF**
    - **PNG**

- **CSV (for raw data extraction)**

2.5 Access Control & Collaboration

2.5.1 User Roles

- **Admin** – Full control over dashboards and permissions.
- **Editor** – Can modify dashboards but cannot delete them.
- **Viewer** – Can only view dashboards.

2.5.2 Permission Management

- Admins can set:
  - **Who can create, edit, or delete dashboards**
  - **Role-based access to data sources**

2.5.3 Collaboration

- **Multi-user editing support**
- **Version history & rollback functionality**

3. Non-Functional Requirements

3.1 Performance

- **Dashboards should load within 2 seconds** under normal conditions.
- **Scalability** to support thousands of users and dashboards.

3.2 Security

- **Encryption** (data at rest & in transit)
- **OAuth / SSO Authentication**
- **Role-based access control**
- **Audit Logs** (tracking changes and access)

3.3 Usability

- **Responsive design** for desktops, tablets, and mobiles.
- **Accessibility compliance** (WCAG 2.1).

3.4 Reliability

- **99.9% uptime** for dashboards.
- **Automated backups & recovery mechanisms**.

3.5 Maintainability

- **Modular architecture** for scalability.
- **Well-documented APIs and user guides**.

4. Assumptions & Dependencies

4.1 Assumptions

- Users have **existing authentication mechanisms**.
- System integrates with **modern web technologies**.

4.2 Dependencies

- **Third-party charting libraries** (e.g., Chart.js, D3.js)
- **Database & API integrations**
- **Cloud infrastructure for data storage**

5. Glossary

- **Dashboard**: A customizable panel for visualizing data.
- **Chart**: A graphical representation of data.
- **Data Source**: A repository providing structured data.

6. Conclusion

This PRD defines the requirements for the **Dashboard Feature**, ensuring a robust, scalable, and user-friendly dashboard experience with rich **data visualization, real-time updates, and collaboration capabilities**.