# Software Requirements Specification (SRS) for IoT Data Management Platform

## 1. Introduction

### 1.1 Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for the IoT Data Management Platform, a scalable, user-friendly web application designed to manage, analyze, and visualize IoT sensor data. The platform empowers users to explore features, register accounts, log in securely, analyze sensor correlations, and customize dashboards, ensuring efficient data-driven decision-making for diverse user groups, including new users and data analysts.

### 1.2 Scope

The IoT Data Management Platform provides:

- **Exploration**: A Homepage for new users to discover features, select data/graph types, and visualize sensor data.
- **Authentication**: Secure login, registration, and password reset functionalities.
- **Analysis**: A dedicated page for analyzing sensor correlations with real-time streaming.
- **Customization**: A dashboard for creating personalized workspaces with widgets. The system is responsive, supports dark mode, and includes consistent navigation with footer links. It integrates with IoT data feeds and ensures accessibility and security.

### 1.3 Definitions, Acronyms, and Abbreviations

- **IoT**: Internet of Things
- **UI**: User Interface
- **SRS**: Software Requirements Specification
- **WCAG**: Web Content Accessibility Guidelines
- **API**: Application Programming Interface

### 1.4 References

- User Stories: Combined User Stories for IoT Data Management Platform (provided).
- Standards: WCAG 2.1, IEEE 830-1998 for SRS structure.

## 2. Overall Description

### 2.1 Product Perspective

The platform is a standalone web application with a front-end built using modern frameworks (e.g., React) and a backend for data processing and authentication (e.g., Node.js, REST API). It interfaces with IoT sensor data feeds (JSON format) and an email service for password resets, operating within a cloud-based infrastructure for scalability.

### 2.2 User Classes and Characteristics

- **New Users**: Non-technical users exploring the platform to understand its capabilities and visualize sample IoT data.
- **Registered Users**: Authenticated users accessing personalized dashboards and data.
- **Data Analysts**: Technical users analyzing sensor correlations and customizing dashboards for insights.
- **Administrators** (future): Manage user accounts and system settings (not in current scope).

## 2.3 Operating Environment

- **Client**: Modern web browsers (Chrome, Firefox, Safari) on desktop and mobile devices.
- **Server**: Cloud-hosted (e.g., AWS, Azure) with Node.js backend and MongoDB for user data.
- **Network**: Stable internet for real-time streaming via WebSocket.

## 2.4 Constraints

- Development uses web technologies (HTML5, CSS3, JavaScript) for cross-platform compatibility.
- Dependency on third-party email services (e.g., SendGrid) for password resets.
- Real-time streaming requires low-latency WebSocket connections.

## 2.5 Assumptions

- Users have basic familiarity with web interfaces.
- IoT sensor data is available in JSON format.
- Backend APIs for authentication and data processing are pre-configured.

# 3. System Features

Each feature is traceable to the provided user stories, ensuring alignment with user needs.

## 3.1 Homepage Exploration (User Story 1)

**Description**: Enables new users to explore platform features, select data and graph types, analyze sensor correlations, and view visualizations.

- **Priority**: High
- **Input**: Data type (e.g., temperature), graph type (e.g., histogram), up to three sensors (checkboxes), time range (date-time pickers), Live Stream toggle.
- **Output**: Welcome message, feature highlights (e.g., Advanced Data Analytics, Customizable Dashboards), visualizations (e.g., histogram), correlation results.
- **Acceptance Criteria**:
  - Displays a welcome message, key features, and a prominent "Get Started" button.
  - Provides dropdowns for data type and graph type, rendering visualizations on submission (e.g., histogram with confirmation message).
  - Includes an "Analyze Sensor Correlation" section with sensor selection (up to three), time range inputs, and Live Stream toggle for real-time data.
  - Features a "Switch to Dark Mode" button in the header.
  - Ensures responsive layout for desktop (min-width: 1024px) and mobile (max-width: 768px).
- **Traceability**: User Story 1

## 3.2 User Login (User Story 2)

**Description**: Allows registered users to log in securely to access personalized features.

- **Priority**: High
- **Input**: Email, password.
- **Output**: Redirect to Homepage or Dashboard on successful authentication.
- **Acceptance Criteria**:
  - Provides email and password input fields on the Login page.
  - Authenticates credentials via secure API call and redirects to Homepage/Dashboard.
  - Includes "Forgot Password?" link to password reset page and "No account? Register here" link to Sign-Up page.
  - Responsive design with footer links (e.g., Contact Us, Privacy Policy).
- **Traceability**: User Story 2

## 3.3 User Registration (User Story 3)

**Description**: Enables new users to create accounts to access platform features.

- **Priority**: High
- **Input**: Full name, email, password, confirm password.
- **Output**: Account creation confirmation and redirect to Login page.
- **Acceptance Criteria**:
  - Provides input fields for full name, email, password, and confirm password with client-side validation (e.g., password strength, email format).
  - Creates account via API and redirects to Login page on successful submission.
  - Includes "Already have an account? Login" link.
  - Responsive design with footer links.
- **Traceability**: User Story 3

## 3.4 Password Reset (User Story 4)

**Description**: Allows users to reset forgotten passwords via email.

- **Priority**: Medium
- **Input**: Email address.
- **Output**: Email with reset link and confirmation message.
- **Acceptance Criteria**:
  - Provides email input field on Forgot Password page with validation.
  - Sends reset link via email service and displays confirmation on submission.
  - Includes "Back to Login" link.
  - Responsive design with footer links.
- **Traceability**: User Story 4

## 3.5 Sensor Correlation Analysis (User Story 5)

**Description**: Enables data analysts to analyze correlations between IoT sensors on a dedicated page.

- **Priority**: High
- **Input**: Up to three sensors (checkboxes), start/end time (date-time pickers), expected correlation value (numeric input, e.g., 0.10), Live Stream toggle.
- **Output**: Correlation results, filtered data, real-time line chart.
- **Acceptance Criteria**:
  - Provides checkboxes for selecting up to three sensors, date-time pickers for time range, and numeric input for correlation value.
  - Processes data on "Analyze" button click and filters results on "Filter Data" click.
  - Updates real-time line chart when Live Stream is toggled (WebSocket-based).
  - Includes "Switch to Dark Mode" button in header and footer with copyright.
- **Traceability**: User Story 5

### 3.6 Customizable Dashboard (User Story 6)

**Description**: Allows data analysts to create and manage personalized dashboards with widgets.

- **Priority**: High
- **Input**: Widget selections (e.g., Line Chart, Data Table), layout adjustments (drag-and-drop).
- **Output**: Custom dashboard with arranged widgets in a grid layout.
- **Acceptance Criteria**:
  - Provides "Add Widget," "Remove," "Resize," and "Save Layout" buttons for widget management.
  - Supports drag-and-drop rearrangement in a responsive grid layout (e.g., CSS Grid).
  - Displays widgets (e.g., Line Chart, Data Table) with dynamic data rendering.
  - Responsive design for desktop and mobile, with footer links.
- **Traceability**: User Story 6

# 4. Non-Functional Requirements

## 4.1 Performance

- Page load time: <2 seconds under normal network conditions (4G/Wi-Fi).
- Real-time streaming: Updates within 1 second for Live Stream toggle.
- Supports up to 1,000 concurrent users with minimal latency.

## 4.2 Scalability

- Handles increasing IoT data volumes (e.g., 10,000 sensor records/hour).
- Cloud-based infrastructure (e.g., AWS EC2, Kubernetes) for horizontal scaling.

## 4.3 Security

- Uses HTTPS for data transfer and bcrypt for password encryption.
- Implements CSRF protection and JWT-based authentication.
- Validates all user inputs to prevent SQL injection and XSS attacks.

## 4.4 Usability

- Intuitive UI with consistent navigation (header, footer).
- Responsive design for screen sizes (320px–2560px).
- Dark mode toggle with high-contrast themes for accessibility.

## 4.5 Accessibility

- Complies with WCAG 2.1 Level AA (e.g., keyboard navigation, ARIA labels).
- Supports screen readers for all interactive elements.

## 4.6 Reliability

- System uptime: 99.9% (excluding scheduled maintenance).
- Error handling for invalid inputs and network failures with user-friendly messages.

# 5. System Interfaces

## 5.1 User Interface

- **Technology**: React.js for dynamic UI, Tailwind CSS for styling.
- **Components**: Dropdowns, checkboxes, date-time pickers, buttons, grid layouts.
- **Features**: Dark mode toggle, responsive design, drag-and-drop widgets.

### 5.2 Software Interfaces

- **Backend API**: RESTful API (Node.js/Express) for authentication, data retrieval, and widget management.
- **Database**: MongoDB for user data and PostgreSQL for sensor data.
- **Email Service**: SMTP integration (e.g., SendGrid) for password reset emails.

### 5.3 Hardware Interfaces

- Interfaces with IoT devices via JSON data feeds (e.g., MQTT or HTTP endpoints).
- Supports standard IoT protocols for sensor data ingestion.

### 5.4 Communication Interfaces

- **HTTPS**: Secure data transfer between client and server.
- **WebSocket**: Real-time data streaming for Live Stream feature.
- **SMTP**: Email notifications for password resets.

# 6. Design Constraints

- **Technology**: Limited to web technologies (React, Node.js) for cross-platform compatibility.
- **Browser Support**: Chrome, Firefox, Safari (latest versions).
- **Data Format**: IoT data in JSON format with predefined schemas.
- **Dependencies**: Requires stable third-party email and IoT data services.

# 7. Quality Attributes

- **Maintainability**: Modular code structure with clear documentation (e.g., JSDoc).
- **Portability**: Cloud-agnostic deployment (AWS, Azure, or GCP).
- **Testability**: Supports unit (Jest), integration, and end-to-end (Cypress) testing.

# 8. Future Enhancements

- Advanced analytics with machine learning for predictive insights.
- Role-based access control (e.g., admin, analyst roles).
- Support for additional visualization types (e.g., heatmaps, 3D plots).
- Mobile app integration for on-the-go access.

# 9. Documentation

- **User Manual**: Guides for Homepage, Login, Sign-Up, Forgot Password, Analyze, and Dashboard pages, with screenshots and step-by-step instructions.
- **Developer Guide**: API documentation, widget development guide, and deployment instructions.
- **Testing Plan**: Includes test cases for all acceptance criteria and performance benchmarks.

# 10. Traceability Matrix

| User Story | Feature | Section |
|---|---|---|
| *User Story 1* | Homepage Exploration | 3.1 |
| *User Story 2* | User Login | 3.2 |
| *User Story 3* | User Registration | 3.3 |
| *User Story 4* | Password Reset | 3.4 |
| *User Story 5* | Sensor Correlation Analysis | 3.5 |
| *User Story 6* | Customizable Dashboard | 3.6 |

# 11. Glossary

- **Widget**: Customizable UI component (e.g., chart, table) on the Dashboard.
- **Live Stream**: Real-time data feed for dynamic visualizations.
- **Correlation**: Statistical measure of relationships between sensor data.
- **Responsive Design**: UI adaptation to different screen sizes (desktop, mobile).

# 12. Appendix

- **Wireframes**: Mockups for Homepage, Login, Sign-Up, Forgot Password, Analyze, and Dashboard pages (to be provided).
- **API Endpoints**:
    - POST /auth/login: Authenticate user.
    - POST /auth/register: Create user account.
    - POST /auth/reset-password: Send password reset link.
    - GET /data/sensors: Retrieve sensor data.
    - POST /data/analyze: Process correlation analysis.