**1) Introduction**

* **Objective:**
  + The objective of this project is to develop a real-time network metrics visualization dashboard.
  + It aims to provide users with a clear and concise graphical representation of key network performance indicators, including packet loss, latency, and packet gain.
  + The dashboard also offers summary statistics and data export capabilities.
* **Scope:**
  + This system focuses on visualizing and analyzing network metrics generated randomly or from a data source.
  + It includes features for:
    - Real-time chart updates.
    - Metric selection (packet loss, latency, packet gain).
    - Chart type selection (line, bar, scatter).
    - Summary statistics display (average, standard deviation, min, max).
    - Data export (CSV, JSON).
    - Chart download (PNG).
    - Theme toggling.
  + The scope is limited to the visualization of the mentioned metrics and does not include advanced network analysis or monitoring features.

**2) System Study**

* **Existing Systems:**
  + Existing network monitoring tools often provide complex dashboards with numerous metrics, which can be overwhelming for users.
  + Many tools require specialized knowledge for configuration and interpretation.
  + Some tools are commercial, which may be costly.
* **Limitations of Existing Systems:**
  + **Complexity:** Existing systems may have a steep learning curve.
  + **Cost:** Commercial solutions can be expensive.
  + **Lack of Customization:** Many tools offer limited customization options.
  + **Overwhelming Data:** Too much data can make it hard to focus on key metrics.
* **Proposed System:**
  + The proposed system is a lightweight, user-friendly dashboard that focuses on essential network metrics.
  + It provides a clean and intuitive interface for visualizing data.
  + It is designed to be easily accessible through a web browser.
  + It utilizes a simple Sqlite database for data storage.
  + It is designed to be easily extensible.

**3) Requirement Analysis**

* **System Functionalities:**
  + Display real-time charts of network metrics.
  + Allow users to select metrics and chart types.
  + Calculate and display summary statistics.
  + Export data in CSV and JSON formats.
  + Download charts as PNG images.
  + Toggle between light and dark themes.
* **Functional Requirements:**
  + The system must fetch and display network metrics data.
  + The system must update charts in real-time.
  + The system must provide data export functionality.
  + The system must provide chart download functionality.
  + The system must provide a theme toggle.
* **Non-Functional Requirements:**
  + **Performance:** The system should update charts and statistics with minimal latency.
  + **Usability:** The interface should be intuitive and easy to use.
  + **Maintainability:** The code should be well-structured and documented.
  + **Reliability:** The system should operate reliably without errors.
* **Hardware Requirements:**
  + A computer capable of running a web server and a web browser.
* **Software Requirements:**
  + Python 3.x
  + Flask web framework
  + SQLite database
  + A modern web browser (Chrome, Firefox, Edge, etc.).
* **Packages/Libraries Used:**
  + Flask
  + SQLite3
  + Chart.js
  + Statistics (Python standard library)
* **Programming Languages Used:**
  + Python
  + HTML
  + CSS
  + JavaScript

**4) System Design**

* **Input Design:**
  + Metric selection (dropdown).
  + Chart type selection (dropdown).
  + Theme toggle button.
  + Export CSV button.
  + Export JSON button.
  + Download Chart button.
* **Code Design:**
  + **Python (Flask):**
    - Handles data fetching, processing, and API endpoints.
    - Manages database interactions.
    - Renders HTML templates.
  + **JavaScript (Chart.js):**
    - Fetches data from the server.
    - Updates charts in real-time.
    - Handles user interactions.
    - Handles data export.
* **Output Design:**
  + Real-time charts (line, bar, scatter).
  + Summary statistics (average, standard deviation, min, max).
  + Downloaded CSV and JSON files.
  + Downloaded PNG chart images.
  + Webpage displayed in browser.
* **Database Design:**
  + **Table: metrics**
    - timestamp (REAL, PRIMARY KEY)
    - packet\_loss (REAL)
    - latency (REAL)
    - packet\_gain (REAL)
* **Data Flow Diagram (DFD):**
  + **Level 0 (Context Diagram):**
    - User interacts with the Network Metrics Dashboard.
    - Dashboard interacts with the Database.
  + **Level 1 DFD:**
    - User selects metric and chart type.
    - Dashboard fetches data from the database.
    - Dashboard processes data and updates chart.
    - Dashboard displays summary statistics.
    - User requests data export or chart download.
    - Dashboard provides data export or chart download.

**5) System Implementation and Testing**

* **Implementation:**
  + Develop the Python Flask application.
  + Create the SQLite database and tables.
  + Implement the HTML, CSS, and JavaScript frontend.
  + Integrate Chart.js for chart visualization.
  + Implement data export and chart download functionality.
* **Testing:**
  + **Unit Testing:** Test individual functions and modules.
  + **Integration Testing:** Test the interaction between different components.
  + **System Testing:** Test the entire system against the requirements.
  + **User Acceptance Testing (UAT):** Allow users to test the system and provide feedback.
  + **Testing Scenarios:**
    - Verify real-time chart updates.
    - Test metric and chart type selection.
    - Validate summary statistics calculations.
    - Ensure data export and chart download functionality.
    - Test theme toggling.
    - Test error handling.
  + Testing will be done by manual inspection of the web page, and verification of the generated files.
  + Testing will also be done by verifying the database entries.