Experiment – 7

Aim:

To develop a Software Requirements Specification (SRS) document for the Event Management System.

Requirements:

Hardware Requirements:

- Computer/Laptop
- Keyboard & Mouse
- CPU (minimum i3 processor)

Software Requirements:

- Microsoft Word / Google Docs
- Browser for research/reference
- Any UML or diagramming tool (optional)

Theory:

A **Software Requirements Specification (SRS)** is a comprehensive documentation of a software system's intended purpose, features, constraints, and behaviour. It serves as a foundation for communication between stakeholders, developers, and testers. An SRS provides a complete description of a system to be developed and lays out functional and non-functional requirements.

SRS for Event Management System

1. Introduction

1.1 Purpose:

The purpose of this document is to define the requirements for the web-based **Event Management System (EMS)**. The EMS is intended to assist users in tracking, categorizing local events.

1.2 Scope:

The EMS will provide features such as user authentication, Event logging, report generation, and. It will support both individual and administrative users and enable role-based functionalities.

1.3 Definitions, Acronyms, and Abbreviations:

- EMS: Event Management System
- UI: User Interface
- UX: User Experience
- JWT: JSON Web Token
- API: Application Programming Interface

2. Overall Description

2.1 Product Perspective:

The EMS is a standalone web application using the MERN stack (MongoDB, Express.js, React.js, Node.js) with a RESTful API structure. It interacts with external services for payment processing and report exports.

2.2 Product Functions:

- Secure login/logout system with role-based access (Admin/User)
- Add, view, edit, delete Events
- Categorize Events and attach receipts
- Generate daily, monthly, and custom reports
- Admin dashboard to monitor user activities

2.3 User Classes and Characteristics:

- **User:** Can log Events, view events, and add events of their own.
- **Admin:** Can manage all users, view events , and handle system-wide settings.

2.4 Constraints:

- Cross-platform responsive design
- Secure API with authentication
- MongoDB schema enforcement with Mongoose

2.5 Assumptions and Dependencies:

- Users have a stable internet connection
- The client device has a modern web browser
- External APIs (e.g., payment, currency) are operational

3. Specific Requirements

3.1 Functional Requirements:

- FR1: The system must allow users to register and login securely using JWT authentication.
- FR2: Users must be able to log, edit, and delete Events with descriptions and category tags.
- FR3: The system must allow receipt/image upload as proof of Event.
- FR4: The system must generate visual reports based on time period, category, and amount.
- FR5: Admin must be able to view, filter, and download organizational Event reports.

3.2 Non-Functional Requirements:

- **Performance:** Must return reports within 3 seconds for up to 1000 entries.
- **Security:** All sensitive data should be encrypted. API calls must be token-validated.
- **Scalability:** Should handle growing number of users and transaction records.
- **Reliability:** System uptime of 99.9%. Regular automated database backups.
- **Usability:** User-friendly UI with support for desktop and mobile interfaces.
- **Maintainability:** Modular backend with documentation and Git-based version control.

4. External Interface Requirements

- **User Interfaces:** Web-based interface with dashboards, forms, and charts.
- **Hardware Interfaces:** Web browser running on any standard computing device.
- **Software Interfaces:**
  - MongoDB for data storage
  - Node.js/Express backend
  - REST APIs for frontend-backend communication
  - Third-party APIs for payment or currency conversion (optional)

Conclusion:

The Software Requirements Specification for the Event Management System has been successfully developed. This document outlines the functional and non-functional requirements clearly, forming a strong foundation for design, development, and testing phases. It ensures all stakeholders have a common understanding of system goals and constraints.