**Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

This document outlines the requirements for a **Simple Internship & Placement Dashboard** project. The application will allow the college placement cell to enter and update data, which will then be presented to users (students and staff) as graphs, statistics, and tables.

**1.2 Scope**

* **Data Entry:** The placement cell can input internship/placement data.
* **Data Display:** Users can view the information in visual formats (charts, graphs) and tables.
* **Technology Stack:** The application will be built using the MERN stack (MongoDB, Express, React, Node.js).

**1.3 Overview**

The project aims to provide an easy-to-use dashboard for tracking internship and placement details, offering quick insights through visuals and tabular data. The initial version will have basic data entry and display functionalities without complex interactions.

**2. Overall Description**

**2.1 Product Perspective**

This project is a standalone web application that provides a simple dashboard for displaying placement and internship data. Data is maintained by the college placement cell and consumed by other stakeholders via a user-friendly interface.

**2.2 Product Functions**

* **Data Entry Module:** A form-based interface for the placement cell to add or update records.
* **Dashboard Display:** A view that shows key statistics (e.g., number of placements, internships, success rates) and visualizations (graphs/charts) based on the entered data.
* **Data Table:** A simple table listing all records with basic filtering options.

**2.3 User Classes and Characteristics**

* **Placement Cell Administrators:** Input and update data.
* **Students and Staff:** View the dashboard and analyze data presented in graphs and tables.

**2.4 Operating Environment**

* **Frontend:** React.js will be used to build a responsive single-page application.
* **Backend:** Node.js with Express will serve RESTful APIs.
* **Database:** MongoDB will store all placement and internship data.
* **Deployment:** The application will run on a cloud platform or on-premise server accessible via a modern web browser.

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Data Entry**

* **FR1:** The system shall allow placement cell administrators to log in and access a data entry form.
* **FR2:** The form shall include fields for key data points such as:
  + Company name
  + Internship/Placement type
  + Number of students placed/interned
  + Date of placement/internship
* **FR3:** The system shall save and update records in the database.

**3.1.2 Data Display**

* **FR4:** The system shall present a dashboard that displays:
  + Summary statistics (total placements, internships, etc.)
  + Graphs/charts (e.g., bar chart, pie chart) to visually represent the data.
* **FR5:** The system shall provide a tabular view listing all records.
* **FR6:** The system shall allow basic filtering (e.g., by year, type) on the table view.

**3.2 Non-Functional Requirements**

**3.2.1 Performance**

* **NFR1:** The dashboard should load within 3 seconds on a standard internet connection.

**3.2.2 Usability**

* **NFR2:** The user interface should be simple and intuitive.
* **NFR3:** The application should be responsive and work on desktops, tablets, and mobile devices.

**3.2.3 Security**

* **NFR4:** Only authorized placement cell administrators can access data entry forms.
* **NFR5:** Basic authentication (e.g., username and password) will be implemented for administrative functions.

**3.2.4 Maintainability**

* **NFR6:** The code should be organized and documented to allow for future improvements or additional features.

**4. System Architecture**

**4.1 Frontend**

* **Framework:** React.js
* **Features:** Responsive design, chart libraries (e.g., Chart.js or Recharts) for data visualization.

**4.2 Backend**

* **Framework:** Node.js with Express.js
* **APIs:** RESTful endpoints for data entry, retrieval, and filtering.

**4.3 Database**

* **Database:** MongoDB will store all records related to internships and placements.
* **ORM/ODM:** Mongoose for object modeling and schema management.

**4.4 Deployment**

* **Environment:** Cloud-based or on-premise server.
* **Security:** HTTPS for secure data transmission.

**5. Appendices**

**5.1 Glossary**

* **MERN Stack:** MongoDB, Express, React, Node.js.
* **RESTful API:** An API that uses HTTP requests for CRUD operations.
* **CRUD:** Create, Read, Update, Delete.

**5.2 Future Enhancements (Optional)**

* Additional user roles (e.g., students for personalized views).
* More advanced filtering and search capabilities.
* Detailed analytics and historical trends.