**Project Requirement Document (PRD)**

**Project Name: School Management System**

**1. Introduction**

The School Management System is designed to streamline the administrative tasks of managing student and teacher records, marks entry, and academic analysis. This system will provide an easy-to-use interface for students, teachers, and administrators.

**2. Objectives**

* To provide an efficient way to manage student and teacher details.
* To facilitate the entry and analysis of student marks.
* To provide dashboards for students and teachers with relevant information.
* To generate reports for academic analysis.

**3. Technologies Used**

* **Frontend:** HTML, CSS, JavaScript (Open-source, beginner-friendly technologies)
* **Backend:** Node.js (Open-source, easy to understand and implement)
* **Database:** SQL (Open-source relational database management system)

**4. Features & Modules**

**4.1 Home Page**

* Displays **School Name** and **Logo**.
* Two main buttons: **Students Dashboard** & **Teachers Dashboard**.

**4.2 Student Dashboard**

* **Students List:** Displays student details class-wise.
* **New Student Entry Form:** Form to add new students.
* **Marks Entry Form:** Allows entry of marks for students.
* **Analysis:** Provides performance analysis reports.

**4.3 Teacher Dashboard**

* **Teacher List:** Displays list of teachers.
* **New Teacher Entry Form:** Form to add new teachers.

**4.4 Student Management**

* **Student Details:** Includes Admission Number, Name, Father's Name, Class, Mobile Number, Date of Birth (DOB).
* **Marks Entry:** Class-wise marks entry with exam type section (FA1, FA2, FA3, FA4, SA1, SA2, Final) for all exams. Exam subjects: Telugu, Hindi, English, Maths, Science, Social.
* **Performance Analysis:** Generates class-wise reports based on exam type, including:
  + Average marks
  + Minimum and maximum marks
  + First student
  + Number of students scoring above 75%
  + Number of students scoring 60% to 75%
  + Number of students scoring 50% to 60%
  + Number of students scoring below 50%

**4.5 Teacher Management**

* **Teacher Details:** Includes Name, DOB, Subject Taught, Experience, Class Assignment.
* **Class Assignment:** Assigns teachers to specific classes.

**4.6 Reports & Analysis**

* Generate academic performance reports.
* Downloadable report options.

**Functional Requirement Document (FRD)**

**1. Introduction**

This document describes the functional requirements for the School Management System.

**2. Functional Requirements**

**2.1 Home Page**

* Displays the School Name and Logo.
* Two buttons: **Students Dashboard** & **Teachers Dashboard**.

**2.2 Student Dashboard**

**2.2.1 Student List**

* Displays a list of students based on selected class.
* Fields: Serial No, Name, Father’s Name, DOB, Mobile Number, Class.

**2.2.2 New Student Entry**

* Form to add new student details:
  + Admission No
  + Name
  + Father’s Name
  + Class
  + Mobile Number
  + Date of Birth
* Submit button to save data to the database.

**2.2.3 Marks Entry Form**

* Select class from a dropdown.
* Select exam type from a dropdown (FA1, FA2, FA3, FA4, SA1, SA2, Final).
* All exams include the same subjects: Telugu, Hindi, English, Maths, Science, Social Studies.
* Show students of selected class with mark entry fields.
* Submit button for each student to save marks.

**2.2.4 Analysis Form**

* Select class from dropdown.
* Select exam type from dropdown.
* Generate report button.
* Download report button.

**2.3 Teacher Dashboard**

**2.3.1 Teacher List**

* Displays a list of all teachers.
* Fields: Name, DOB, Subject, Experience, Assigned Class.

**2.3.2 New Teacher Entry**

* Form to add new teacher details:
  + Name
  + Date of Birth
  + Subject Taught
  + Experience (in years)
  + Assigned Class
* Submit button to save details.

**2.4 Reports & Analysis**

* Generate class-wise performance reports.
* Download report as PDF/Excel.

**3. Database Schema**

The database will include the following tables:

**3.1 Students Table**

* student\_id (Primary Key, Auto Increment)
* admission\_no (Unique)
* name
* father\_name
* class
* mobile\_number
* dob

**3.2 Teachers Table**

* teacher\_id (Primary Key, Auto Increment)
* name
* dob
* subject\_taught
* experience
* assigned\_class

**3.3 Marks Table**

* mark\_id (Primary Key, Auto Increment)
* student\_id (Foreign Key referencing Students Table)
* exam\_type
* telugu\_marks
* hindi\_marks
* english\_marks
* maths\_marks
* science\_marks
* social\_marks

**3.4 Reports Table**

* report\_id (Primary Key, Auto Increment)
* class
* exam\_type
* avg\_marks
* max\_marks
* min\_marks
* first\_student
* above\_75\_count
* between\_60\_75\_count
* between\_50\_60\_count
* below\_50\_count

**4. Non-Functional Requirements**

* **User-Friendly Interface:** Simple and easy-to-use UI.
* **Performance:** Should handle a large number of student and teacher records.
* **Security:** Secure login for administrators.
* **Scalability:** System should be scalable to add more features in the future.

**5. Conclusion**

This document outlines the requirements for the School Management System. It serves as a guide for developers to implement the features efficiently. The system aims to improve school administration by automating student and teacher management along with academic analysis.