School Management System

Firstly, this School Management System is a web enabled application developed in PHP and powerful MySQL database backend. Moreover, to implement School Management System, schools do not require expensive hardware and software, all schools need is internet connection and desktops. Additionally, our system works as a centralized database and application that schools can easily access the system from anywhere based on the login credentials. This project is a platform independent system that virtually any user can access from anywhere through a standard internet accessible system. We can also customize E-School for individual school needs.

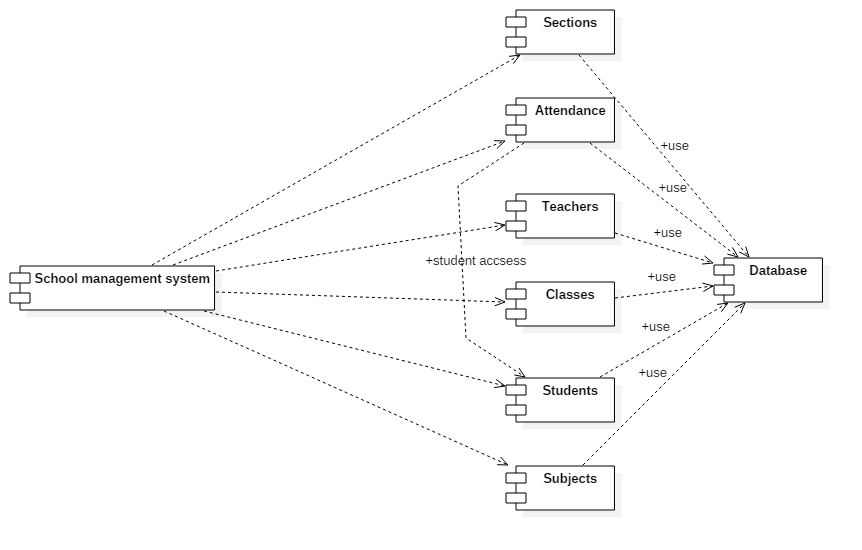
Secondly, this project helps administrators get the most accurate information to make more effective decisions. Teachers and administrators gain time saving administrative tools. Parents gain immediate access to their children’s grades and students can track their own progress. The project has features makes it possible to generate schedules and reports in minutes. This can retrieve attendance records, grade checks, report cards, transcripts, and form letters in just a few clicks.

**Features of the Project:**

* Dashboard: Admin
* Students
* Fees Collection
* Hostels
* Classes
* Sessions
* Notices
* Fees Structure
* Branch
* Teachers
* Subjects
* Streams
* Hostels
* Timetable
* Events
* Notice/Announcements
* Exam Results
* Parents
* Exam Categories
* Sessions
* Student Categories
* Class Attendance
* Fees Structure
* User Control
* Screenshots

**Users:**

* School manager
* School academic affairs
* Students
* Parents
* Teachers



**Coding Standards**

* [Opening and Closing PHP Tags](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#opening-and-closing-php-tags)

### [Single and Double Quotes](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#single-and-double-quotes)

### [Naming Conventions](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#naming-conventions)

### [Interpolation for Naming Dynamic Hooks](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#interpolation-for-naming-dynamic-hooks)

### [Space Usage](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#space-usage)

### [Indentation](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#indentation)

### [Brace Style](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#brace-style)

## [Object-Oriented Programming](https://developer.wordpress.org/coding-standards/wordpress-coding-standards/php/#object-oriented-programming)

الفرزدق احمد

**Main component: Sections & Attendance**

**ؤ Sections**

1. The function name: listSections()
2. Algorithm steps:

* Connect to the database
* Check if search POST value not empty
* Check if order POST values not empty
* Check if POST length value not equal (-1)
* Select all sections
* Store the values in an array
* Echo array output

1. Test result:  **Pass**
2. Error message:
3. The function name: addTeacher ()
4. Algorithm steps:

* Input POST teacher name
* Connect to the database
* Check if POST teacher name values not empty
* Connect to the database
* Insert the data

1. Test result:  **Pass**
2. Error message:
3. The function name: getTeacher()
4. Algorithm steps:

* Input POST teacher id number
* Connect to the database
* Select the value based on teacher id
* Store the data in an array
* Echo array output

1. Test result:  **fail**
2. Error message:

Empty array as a return

1. The function name: addSection ()
2. Algorithm steps:

* Input POST section name
* Check if section name not empty
* Connect to the database
* Insert into section table

1. Test result:  **Pass**
2. The function name: updateSection()
3. Algorithm steps:

* Input POST section id
* Check if section name not empty
* Connect to the database
* update section table

1. Test result:  **Pass**
2. The function name: deleteSection()
3. Algorithm steps:

* Input POST section id
* Check if section name not empty
* Connect to the database
* Delete from section table

1. Test result:  **Pass**

**Attendance**

1. The function name: getStudents()
2. Algorithm steps:

* Input POST class id and section id
* Input attendanceYear, attendanceMonth, attendanceDay
* Connect to the database
* Select attendance based on class id & section id
* Check if attendance select is done
* Echo array output

1. Test result:  **pass**
2. The function name: updateAttendance()
3. Algorithm steps:

* Input attendanceYear, attendanceMonth, attendanceDay
* Connect to the database
* Select attendance based on class id & section id
* Check if attendance select is done
* Check if student id is valid
* Update attendance state value
* Echo "Attendance updated successfully!"

Test result:  **pass**

1. The function name: attendanceStatus()
2. Algorithm steps:

* Input attendanceYear, attendanceMonth, attendanceDay
* Connect to the database
* Select attendance based on class id & section id
* Check if attendance select is done
* echo "Attendance already submitted!";

Test result:  **pass**

مودة محمد حسن

**Main component: Teachers**

**ؤ Sections**

1. The function name: listTeacher()
2. Algorithm steps:

* Connect to the database
* Check if search POST value not empty
* Check if order POST values not empty
* Check if POST length value not equal (-1)
* Select all sections
* Store the values in an array
* Echo array output

1. Test result:  **Pass**
2. Error message:
3. The function name: addTeacher ()
4. Algorithm steps:

* Input POST teacher name
* Connect to the database
* Check if POST teacher name values not empty
* Connect to the database
* Insert the data

1. Test result:  **Pass**
2. Error message:
3. The function name: getTeacher()
4. Algorithm steps:

* Input POST teacher id number
* Connect to the database
* Select the value based on teacher id
* Store the data in an array
* Echo array output

1. Test result:  **fail**
2. Error message:

Empty array as a return

1. The function name: deleteTeacher()
2. Algorithm steps:

* Input POST section id
* Check if section name not empty
* Connect to the database
* Delete from section table

1. Test result:  **Pass**

مثاني ابوسفيان

**Main component: Classes**

1. The function name:
2. Algorithm steps:
3. Test result:
4. Error message:

محمد الفاتح

**Main component: Students**

1. The function name:
2. Algorithm steps:
3. Test result:
4. Error message:

بلسم بشير

**Main component: Subjects**

1. The function name:
2. Algorithm steps:
3. Test result:
4. Error message: