School Management System Documentation

Abstract

This document discusses the *School Management System*. Worldwide technology has continuously improved the quality of education and assisted in utilizing groups, departments, and various users, additionally it has increased the productivity and performance of such educational institutions which has helped them maximize the potential of students. Integrating the latest technologies into current systems can significantly improve educational activities, staff productivity, and user satisfaction. By implementing the right functionalities, institutions can offer increased performance and deliver the key functionalities that make processes easier for administrators, staff, students, and lecturers. Our system effectively address the needs required in the education sector.

Why Choose Us?

- We make it possible to:
- Manage all school's Courses, Resources, and Timetables for study.
- Locate lecturers, units, resources, and students in few seconds.
- Print lectures and the units that they teach for any academic year.

Standout Functionalities

- School Administration:

    i. Lecturers, Staff & Students Management.
    ii. Course Management.
    iii. Unit Management.
    iv. Timetable Management.
    v. Graduation Completion.
    vi. Course Management.

- Lecturer:

    i. Materials Management.
    ii. Send Notifications to the Class.

- Staff:

    i. View Staff Messages.
    ii. Send Staff Messages.

- Students:

    i. Register for a Course.
    ii. Apply for Graduation.

System Usefulness

Our system is environmentally & user-friendly, not only does it help make management of a school easier, but it also gives a competitive edge over its competitors. The School Administrator will have greater control over the operations and processes in the school. Our system can easily be integrated to comply with the regulations of schools in relation to the admission process, resources, timetables and up until graduation. It ensures that the reports are informative and precise, complying with school requirements. Students and lecturers can view their personal and academic information through the system. We offer a centralized secure data storage server.

Growth

Our system will be able to accommodate any future institution growth. We invest heavily in the developing of new functionalities & features constantly. We stay aware of new technologies too.

Performance

The system design and platform utilize the latest developments in computer technology. The system has a graphical user interface, it is client-server architecture being web based & user friendly.

Tools Utilized

Tool	Purpose
Sublime Text 3	This was the IDE used to code & debug the system.
MySQL Server	This was the database management system used to securely store the data and user credentials needed for the system. Hosted via the XAMPP Program.
Draw.io	This was the online tool used to design the diagrams.
Laravel	This is a powerful MVC PHP framework, designed for developers who need a simple and elegant toolkit to create full-featured web applications.
Composer	This is a tool which includes some dependencies and libraries. It allows a user to create a project with respect to the mentioned framework.
Node.js	This tool sends the task to the computer's file system, readies the computer to handle the next request and when the file system has opened and read the file, the server returns the content to the client.

Table 1 : Tools Used

Crud and API Functionalities

1. The **delete_resource** function – This function is used to **DELETE** a resource from the database by the lecturer. Once the Resource ID has been found it will delete the row from the table in the database and redirect the lecturer back to the homepage.

```
54      public function delete_resource(Request $request){
55          $id = $request->input('uid');
56          $data = resource::find($id);
57          $data->delete();
58          return redirect()->back();
59      }
60
61  }
62
```

Figure 1 : Delete A Resource

2. The **reg_grad** function – This function is used to **INSERT** a graduation record into the database by a student. When the Student ID and Course fields have been filled the table is saved, and the student is redirected back to the Homepage.

```
12          public function reg_grad(Request $request){
13          $graduation = new graduation;
14
15          $email = $request->input('email');
16          $course = $request->input('course');
17
18          $graduation->email=$email;
19          $graduation->course=$course;
20
21          $graduation->save();
22
23          return redirect()->back();
24      }
```

Figure 2 : Insert A Graduation

3. The **update** function – This function is used to update a user password, once the required fields have been filled and validated, it requests for the User ID. Updates the password columns and saves the users table, then redirects the user back to the My Profile Page.

```
    public function update($user, array $input)
    {
        Validator::make($input, [
            'current_password' => ['required', 'string', 'current_password:web'],
            'password' => $this->passwordRules(),
        ], [
            'current_password.current_password' => __('The provided password does not match your current password.'),
        ])->validateWithBag('updatePassword');

        $user->forceFill([
            'password' => Hash::make($input['password']),
        ])->save();
    }
}
```

Figure 3 : Update User Password

4. The **Select** function – This is a basic PHP function to output arrays of the requested table within columns & rows of a HTML Table.

```
<?php
$conn = mysqli_connect("localhost", "root", "", "school_system");
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT `id`, `name`, `phone_number`, `course`, `email`, `profile_photo_path`, `created_at`, `updated_at` FROM `
  users` WHERE `user_type` = '0'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
// output data of each row
while($row = $result->fetch_assoc()) {
echo "<tr><td>" . $row["id"] . "</td><td>" . $row["name"] . "</td><td>" . $row["phone_number"] . "</td><td>" . $row["course
  "] . "</td><td><a class='none' href='mailto:" . $row["email"] . "'>" . $row["email"] . "</a></td><td>" . $row["created_at
  "] . "</td><td>" . $row["updated_at"] . "</td><td><img src= 'images/" . $row["profile_photo_path"] . "' width = 150
  title='" . $row["profile_photo_path"] . "'></td></tr>";
}
echo "</table>";
} else { echo "No results"; }
$conn->close();
?>
```

Figure 4 : Select A Table from The Database

5. The **Route** function – This is a functionality that is used to redirect the user to a particular View or Controller when requested from by a user form or button.

```php
Route::get('/',[HomeController::class,'index']);

Route::get('/home',[HomeController::class,'redirect']);

Route::middleware([
    'auth:sanctum',
    config('jetstream.auth_session'),
    'verified'
])->group(function () {
    Route::get('/dashboard', function () {
        return view('dashboard');
    })->name('dashboard');
});

Route::post('/add_user',[AdminController::class,'upload']);

Route::post('/add_course',[AdminController::class,'upload1']);

Route::post('/add_timetable',[AdminController::class,'upload2']);

Route::post('/delete_user',[AdminController::class,'delete_user']);

Route::post('/update_graduation',[AdminController::class,'update_graduation']);

Route::post('/add_resource',[LecController::class,'add_resource']);

Route::post('/delete_resource',[LecController::class,'delete_resource']);

Route::post('/send_message',[LecController::class,'send_message']);

Route::post('/send_message1',[StaffController::class,'send_message1']);

Route::post('/reg_course',[StudController::class,'reg_course']);

Route::post('/reg_grad',[StudController::class,'reg_grad']);
```

Figure 5 : Route Functions

Design Diagrams

1. **Class Diagram** – Describes the attributes and operations of a class and the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages:

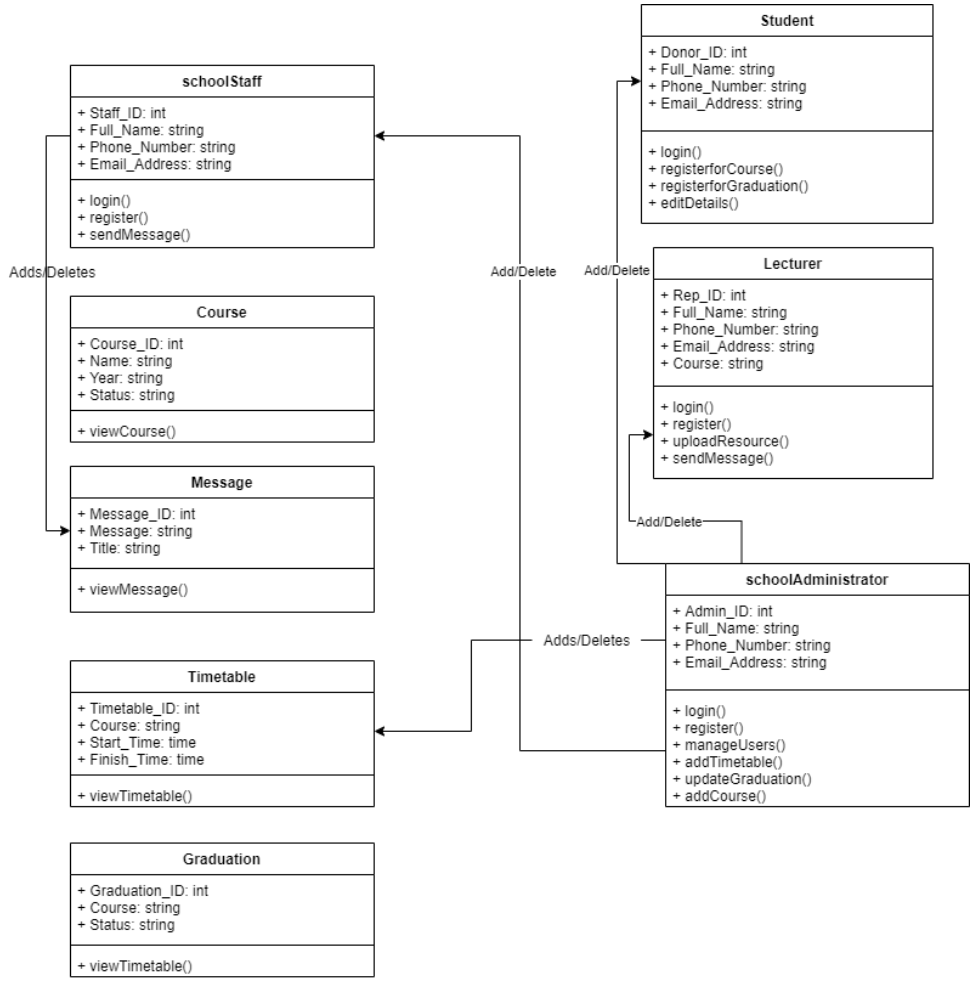

Figure 6 : Class Diagram

2. **Database Schema** – This is a is considered the “blueprint” of a database which describes how the data may relate to other tables or other data models:

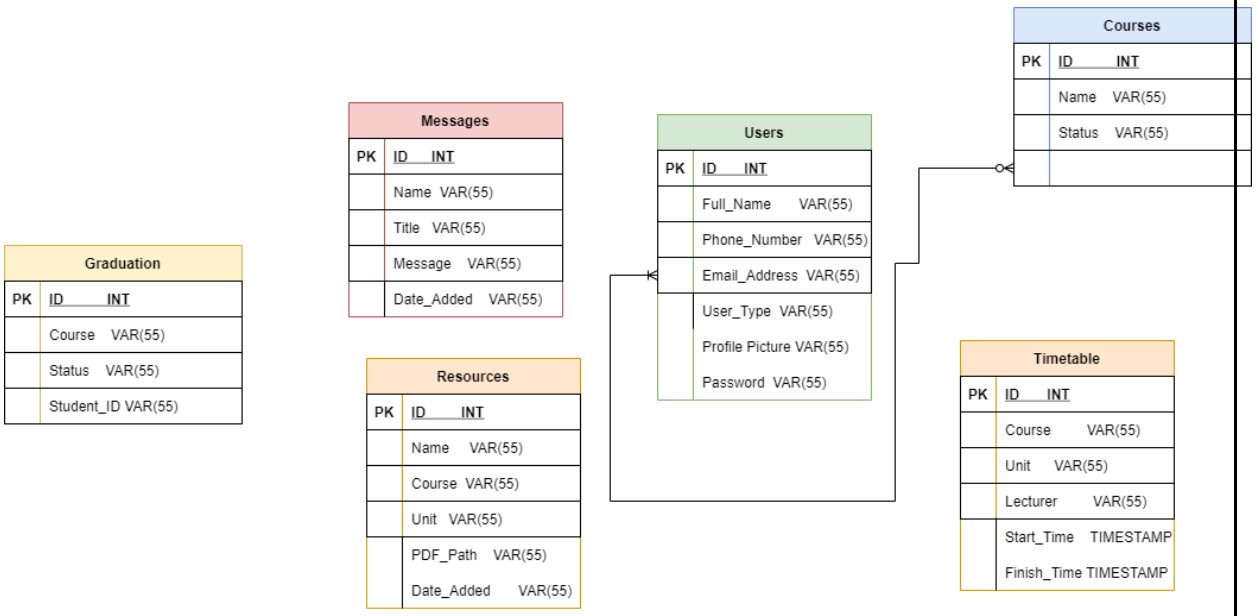

Figure 7 : Database Schema

3. **Entity Relational Diagram** – This is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system:

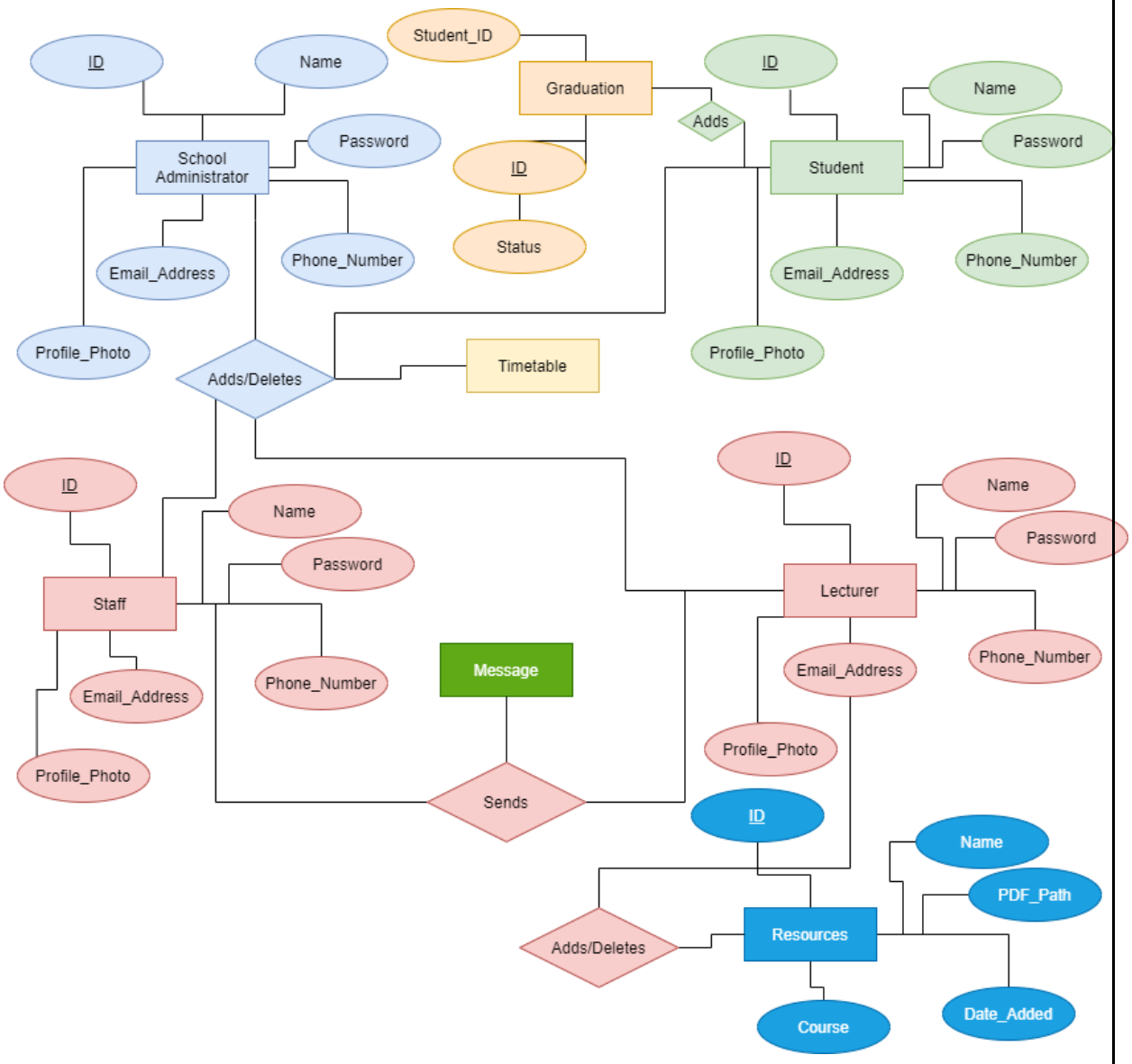

Figure 8 : ER Diagram

Further Advancements

- Our system meets the technical requirements of the education sector, such as lecture, staff, resources, students & courses management. But to make it easier for the users to access the system features an upgraded graphical user interface is needed to have an easier to use page layout.