UNIVERSITY AMERICAN COLLEGE SKOPJE

SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Course

Distributed databases

School System 2020

Student: Mark Ruzinov, Mateja Miteva, Marko Stojmenovski
ID: 4187, 4255, 4193

Instructor: *Veno Pachovski, PhD*
Assistant: *Bojan Bogdanovski, MSc,*
Deni Mirchevski, BSc

Academic year 2020/2021

Contents

Introduction

The topic chosen represents a typical education center's record system. One that they might use to keep track of students' details and payment status. This is meant to provide the staff and employees with one central system that lets all staff members access and update records from one location and one database.

The goal here is to give everyone a unified and synchronized experience when keeping track of students in an educational setting. By giving everyone accounts and access to a central online system that can be accessed in a web browser, staff can simply log in anywhere and access their records easily.

The idea is to provide a central and unified platform that can be accessed using a web browser instead of having local records on a computer that have to be used with special software.

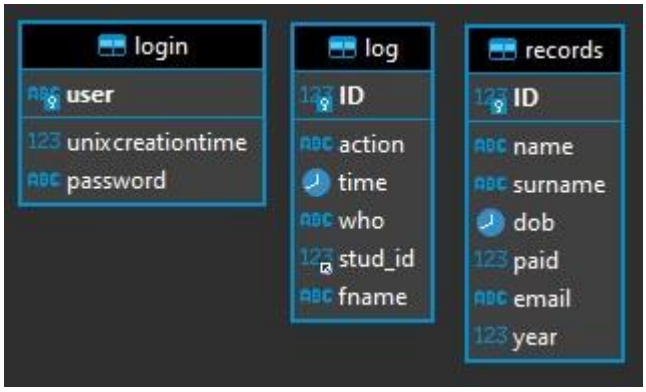
This software's purpose is to give a university's staff and employees/administration access to their student's records, regardless of where they are accessing it on-site, and regardless of the device they are accessing it from.

School System 2020

This school system consists of two main parts – a frontend, consisting of a webserver that serves webpages consisting of HTML and CSS, and a backend that consists of a database server and PHP to process the data on the webpages.

Database Configuration

The database server we used is an instance of MariaDB 10.3.25, however MySQL servers will work fine with this as well. Our database is named schoolSystem, which contains three tables:



- “login”: This table contains the login records for each user of the system. In here, the admin user is one that must exist at all times. In this table, there are two fields:
    - “user”, varchar(100), NOT NULL
    - “unixcreationtime”, bigint(20), NULL
    - “password”, longtext, DEFAULT NULL

  The user field is the primary key of this table, as all accounts must have a name set – one that is unique. The UNIX creation time is used to create a unique salt for each password. The password is hashed using SHA2 with a 512 bit length (also known as SHA-512). This is to prevent anyone but the user from knowing the password.

- “records”: As the name suggests, this table holds all student records present in the system. Because of that, there are several fields which aim to define each student:
    - “name”, varchar(100), NOT NULL
    - “surname”, varchar(100), DEFAULT NULL
    - “dob”, date, DEFAULT NULL
    - “ID”, int(11), NOT NULL, AUTO INCREMENT
    - “paid”, tinyint(1), NOT NULL
    - “email”, varchar(100), DEFAULT NULL
    - “year”, int(11), DEFAULT NULL

  Here, the ID serves as the primary key for this table, as some students may have repeating names, surnames, birthdays, and so on. The other fields are meant to just give each student more defining characteristics, while also keeping their payment status in mind.

- “log”: This is a logging table that holds log entries for actions that are performed on students in the records table. Between updates, creations, and deletions, all of those get logged into this table with a timestamp and a field of who performed the action, along with the ID of the student as well as their name.
    - “action”, varchar(100), DEFAULT NULL
    - “time”, datetime, DEFAULT NULL
    - “who”, varchar(100), DEFAULT NULL
    - “stud_id”, int(11), DEFAULT NULL
    - “ID”, int(11), NOT NULL, AUTO INCREMENT
    - “fname”, varchar(100), DEFAULT NULL

  In this table, the ID field is the primary key, which enumerates each log entry. Originally, the goal was to have “stud_id” have a constraint on it, serving as a foreign key to the “ID” field in the records table – however, there were problems with this idea.

  Removing a record from the records table would either force the database to run in a mode that does not respect constraints, or a “cascade” delete option would have to be set, which would delete the records from the log table as well – something we absolutely do not want happening.

Because of these reasons, we had to make stud_id serve as its own independent field with no constraints. It does however always contain the ID of the student that the action was performed on, even if that student is no longer present in the records table.

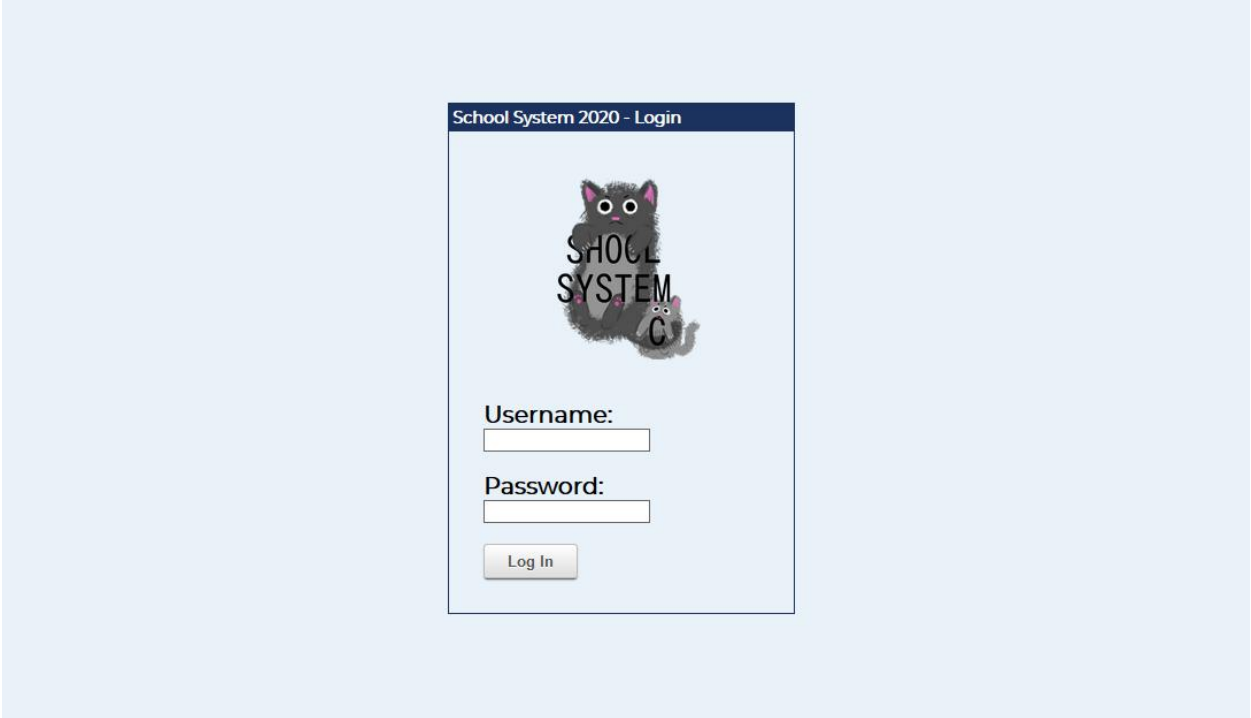
Application Documentation

School System 2020 is made to give its users a simple and familiar way to access their student's information from a web-based user interface. Installation is as simple as adding the files to your web server's working directory, and then creating a database with the corresponding tables on your database server (or running the included SQL script which contains sample data).

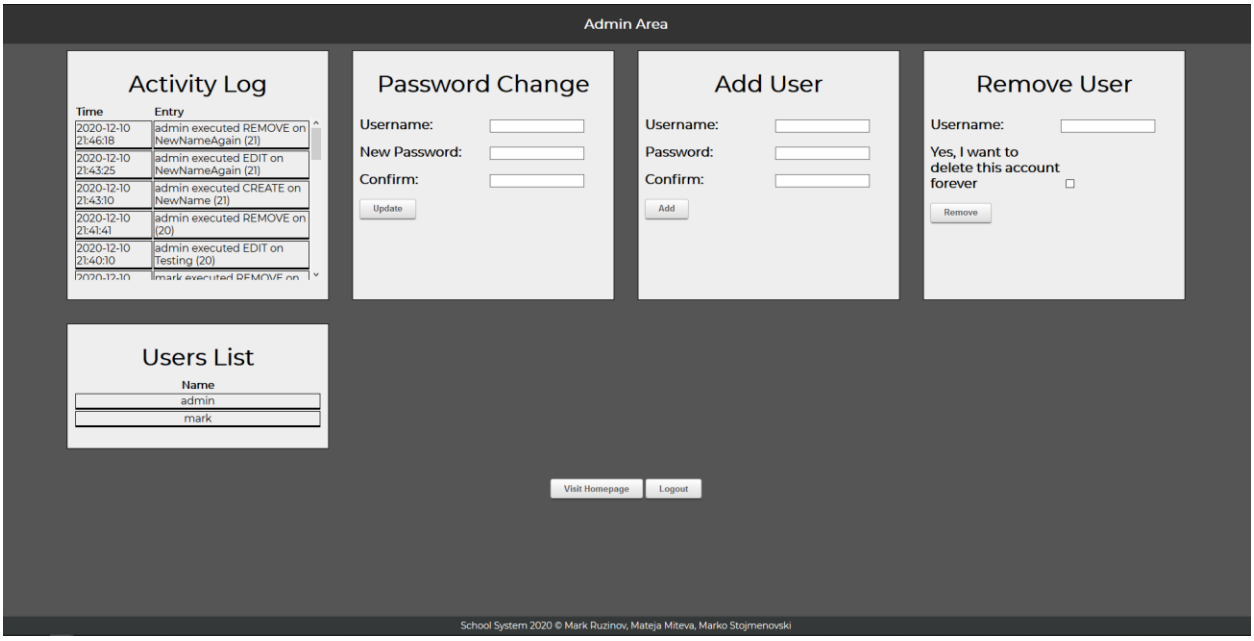
After that, all that is left is to configure your database's credentials and location by editing the "db.php" file in the root folder.

```php
<?php
   define('DB_SERVER', 'localhost:3306');
   define('DB_USERNAME', 'root');
   define('DB_PASSWORD', '');
   define('DB_DATABASE', 'schoolSystem');
   $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

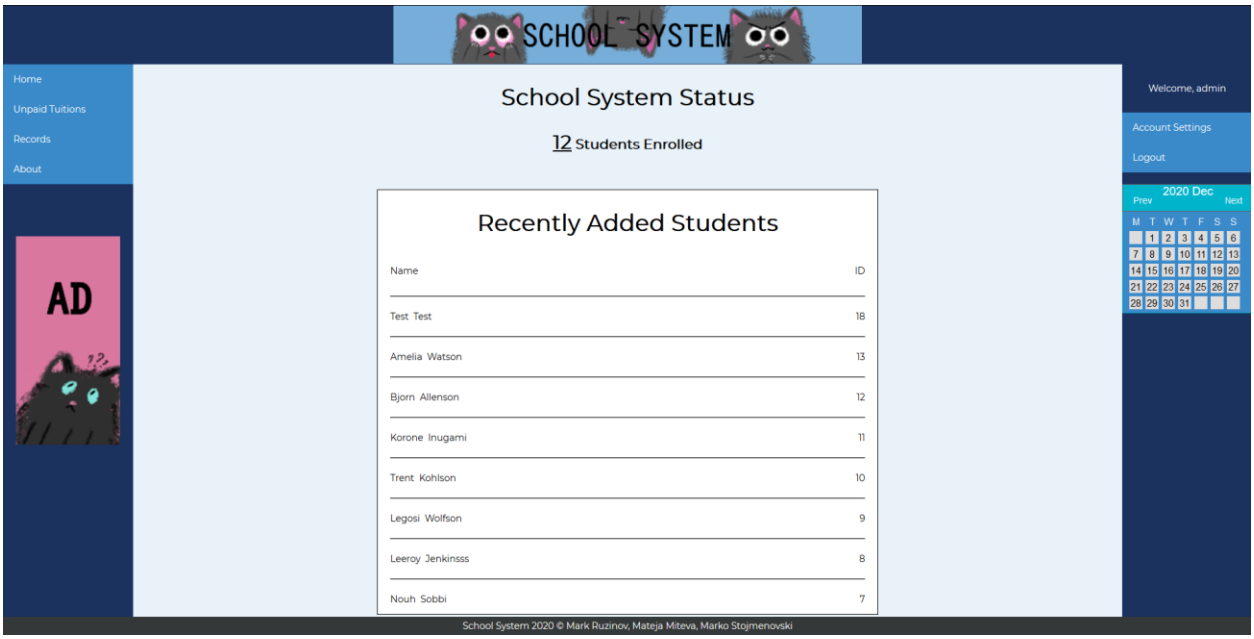
After that, opening login.php in a web browser should take you to the login screen, after which you will be good to go. If not, you can always check your web server's error log to check where the error lies exactly.



From here, you can enter the administrative panel by using the default login username and password, which are “admin”, and “1234”, respectively. After clicking “Log In”, you will be greeted by this panel page:



Now that you are in the administrative panel, you can create users, delete the example user, change the password of the admin account, and visit the main site. This should be all that you need to get up and running.



Our user experience and design language aim to give your users a fun and more playful appearance to what is typically a very dull program. On the left sidebar there is a menu containing all of the places one would need to go to perform actions on a student's record.

- Home – Takes you to this page
- Unpaid Tuitions – Takes you to a page where all students with an unpaid tuition can be seen easily
- Records:
    - Add Record: Add a new record to the system
    - Edit Record: Edits an existing record already present in the system
    - Remove Record: Removes a record from the system (permanent deletion)
    - View All: Shows you all records present in the system in a single table
- About: Opens the about page

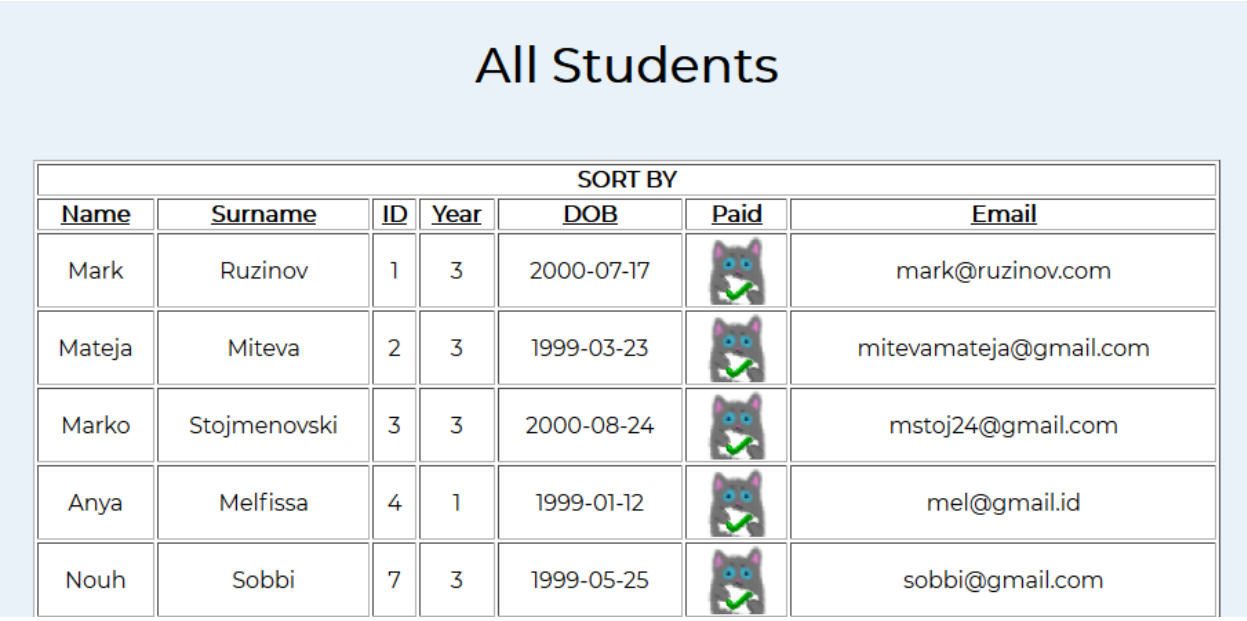
These menu options provide users with 95% of what will usually be done through this system, which is record manipulation. The "ad" included as a joke will cycle through a selection of self-aware ads that poke fun at today's ad-infested Internet. These can be easily removed by removing the PHP block of code in the middle of the header.php file. It is this one:

```php
<?php
$imagesDir = 'ads/';

$images = glob($imagesDir . '*.{jpg,jpeg,png,gif}', GLOB_BRACE);

$randomImage = $images[array_rand($images)];
echo "<img id=\"ad\" draggable=\"false\" src=\"{$randomImage}\"></img>";
?>
```

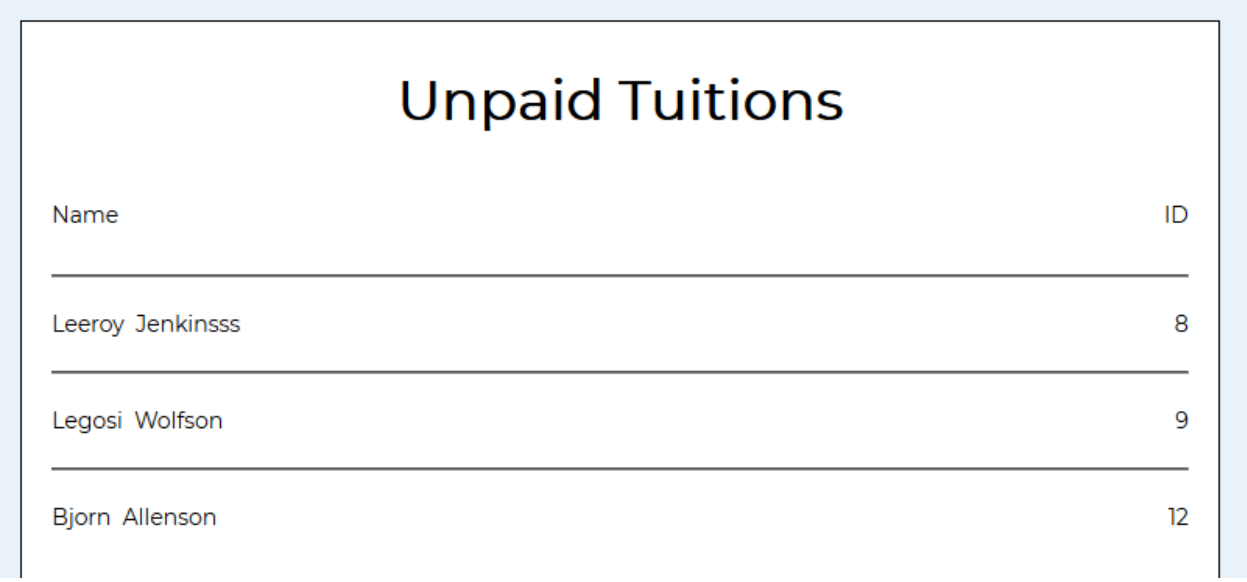
The right-hand sidebar contains a couple of additional features:

- Account Settings: Provides settings for the currently logged in account, as well as a way to enter the admin panel.
- Logout: Logs the current user out from their session
- Calendar: A calendar view of the current month (months can be cycled through with the Prev and Next buttons)

Clicking on the number in the “N Students Enrolled” will also take you to the all students view, which shows you all of the students present in the system. Below that, you will find a list of the 8 most recently added students, together with their full name and ID.

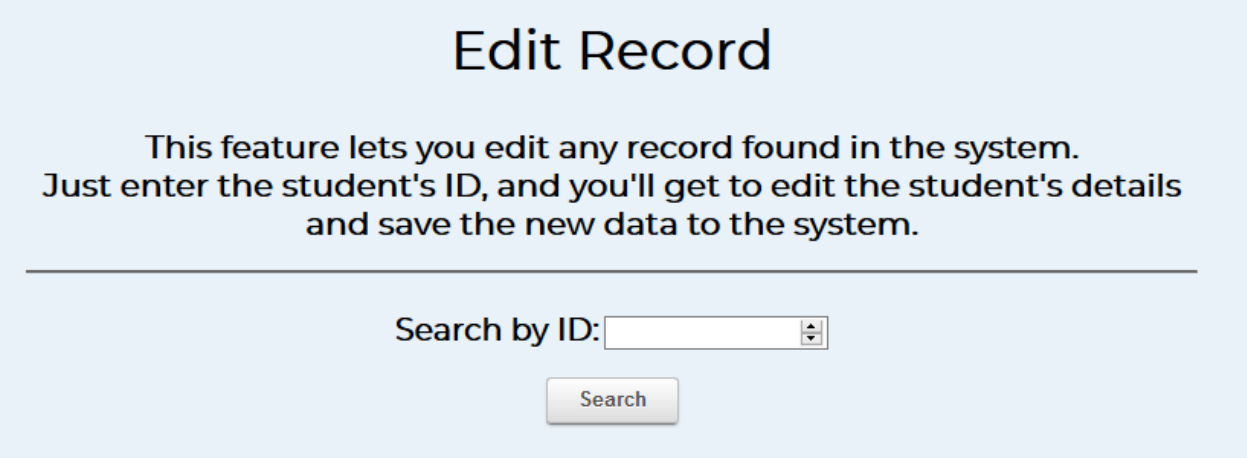


The “All Students” view is an easy way to view and sort your student records to something that works best for you. The table is sorted by ID by default, but this can easily be changed by clicking on any of the table headings. All sorting is done in ascending order (1-9, and A-Z).



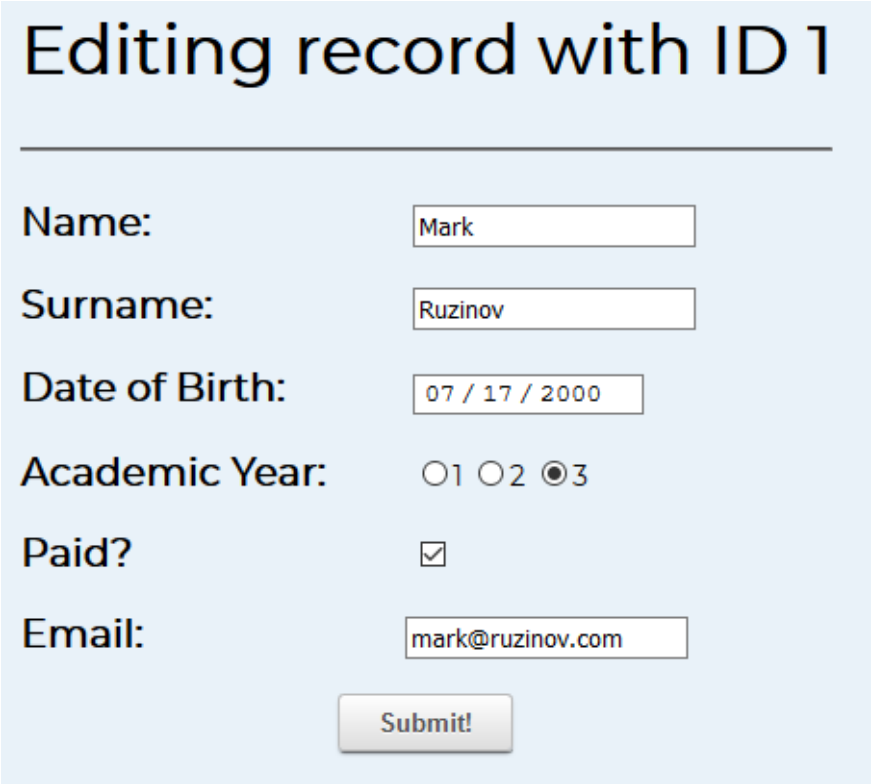
The “Unpaid Tuitions” page gives you an insight on which students still need to pay for their tuition.

The “Add Record” page gives you the ability to add a record to the system. Each field is validated and checked before adding the record to the system. This means that you won’t be able to make a mistake when entering a new student.

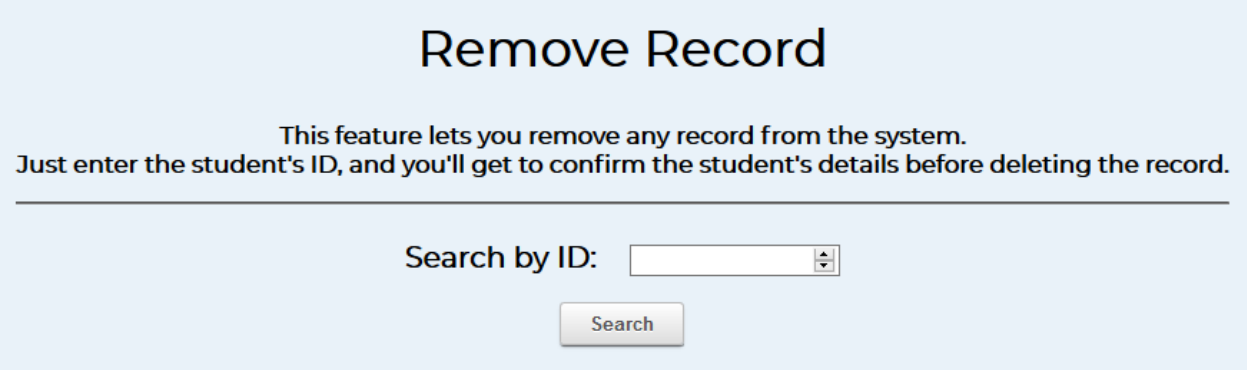


The “Edit Record” page gives you a textbox where you can enter the ID of the student you wish to edit. We debated giving a list of the most recently-added students here, however we figured edits could happen for any student in the database, therefore we left it out.

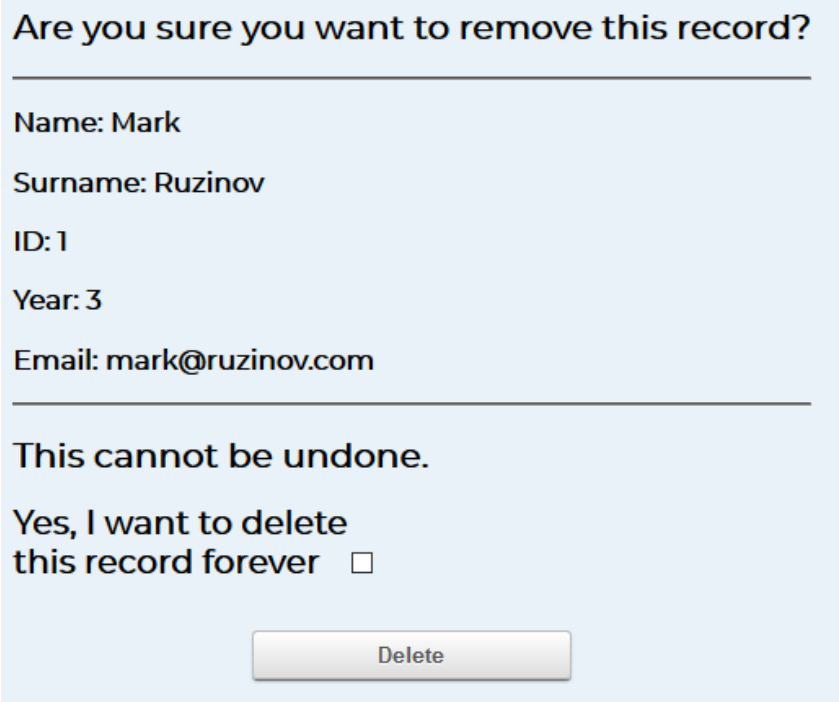
Enter an ID, and click Search. If the ID does not exist, the page will tell you to enter a new ID. If it does, this form will open:



Here, all of the existing details are pre-entered for you. You can now edit them, and then click on Update. The record will save the new values, and then you will be returned to the home page automatically.



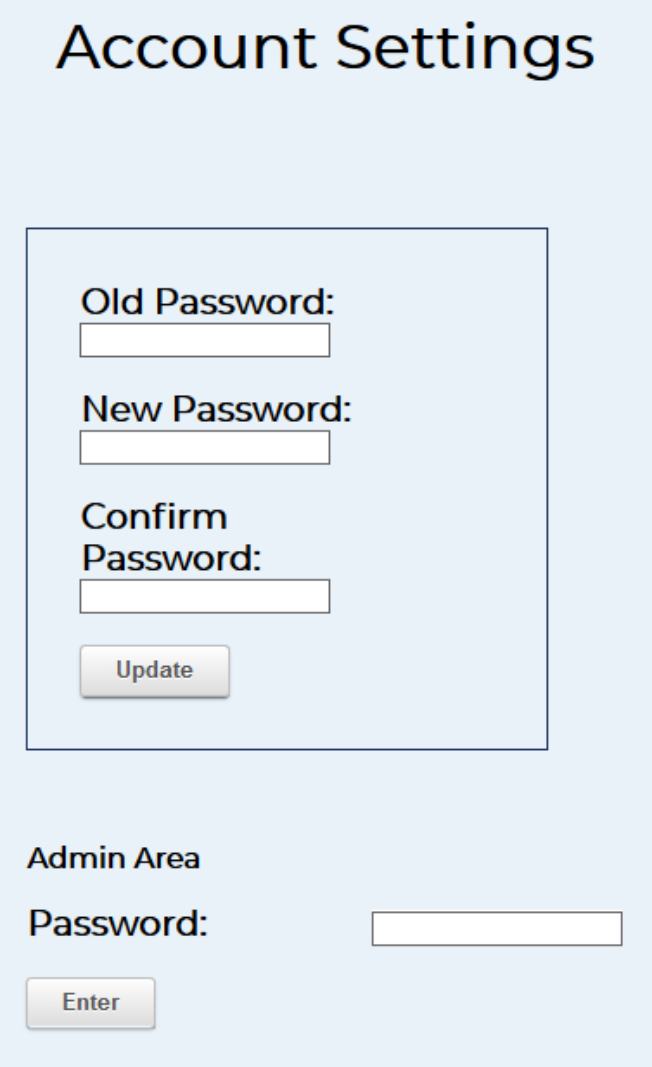
Similar to the Edit Record page, the “Remove Record” page will give you the option to enter a student’s ID. If the ID exists, it will take you to the confirmation page:



Here, you will be able to see that student’s details that are currently present in the database. To delete the record, you must tick the checkbox that asks you to confirm, then you must press the Delete button in order for the deletion to occur.

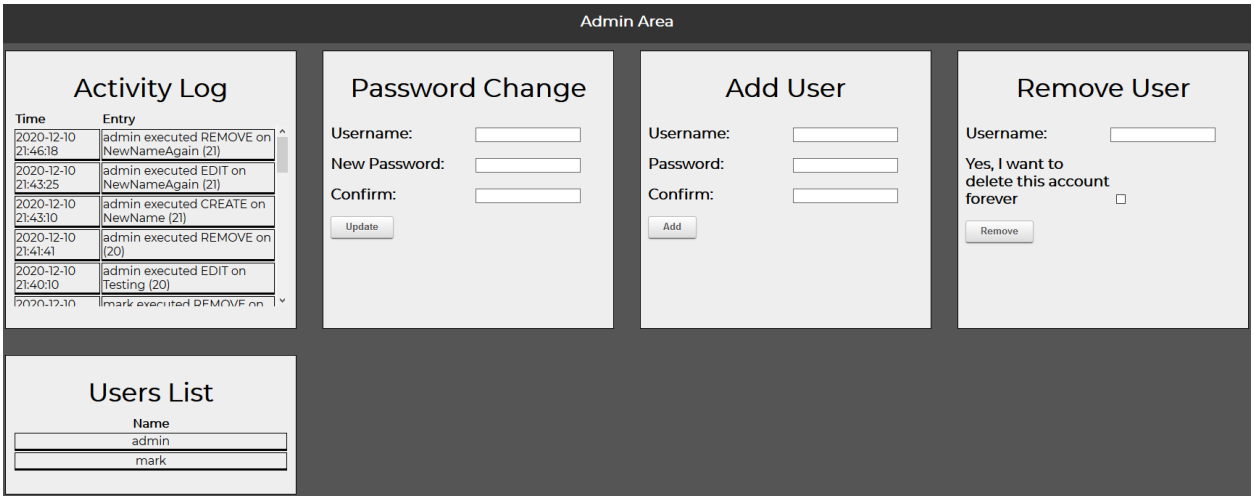
You can leave this page at any time by going to the home page, or by pressing back on your browser window. No destructive actions will occur unless both the box is checked and Delete is pressed.

The “View All” page takes you to the same page as shown earlier regarding the sorting of all students in one table.



Changing your account's password is easy – you can visit the "Account Settings" page and change your password here. In order to change your password as a regular user, you must know your existing password. Simply enter the old password, and your new password twice, and click on Update. You will then be logged out, where you can now log in as the new user.

As an extra feature, you can log into the administrative panel here if you know the admin account's password. Entering this password will sign you in as the admin user for the rest of the session.



The admin area gives the administrator control over the inner workings of the system. From the top left to the bottom right, we have:

- Activity Log: This panel shows you the 50 most recent entries in the log table. This includes student record creations, edits, and deletions, including which account performed the action, which student it was performed on, and the time that it occurred. This is a useful tool when you need to check why a user is missing, or who added a user that doesn't go to the school.
- Password Change: This panel provides the ability to change the password of any registered account without needing to know the existing password. Just enter a registered username, the new password twice, and Update it.
- Add User: This lets you add a new user to the system. Enter a new unique username, and a new password twice, and the user will be created. This panel does not allow duplicate usernames.
- Remove User: You can remove any existing user from the system using this. Just enter their username, check the box to confirm deletion, and click on Remove to delete the account. **NOTE:** You cannot delete the admin account.
- Users List: This shows a list of all currently registered users on the system.

An additional bonus for developers or those willing to expand on the functionality of the admin panel – you'll be delighted to know that each panel is modular and you can add as many as you want to the page – the panels auto align and arrange to form a grid that is as wide as the viewport will allow. With this, you can add new panels that provide new information, change new settings, and more.

Conclusion

With this school system, we have created an easy-to-understand and easy-to-use system that almost anyone can pick up and start using within minutes. With protection in place for erroneous entries, there is no risk of a user entering something that could break the system. Many users can be logged in at the same time with no downsides.

What can be improved on is for sure the design aspect. We are not graphic designers – we designed webpages that are simple and functional – they are not the prettiest webpages for sure. Additionally, we want to enable better security in the future – access control lists, permission systems – something that can limit a user's actions based on their role. Outside of that, there are possibilities for even better features. For example, a fully responsive design, a desktop app if someone prefers that, the ability to search and edit records inline, and more.

This can be adapted to almost any field that requires a record system of some sort – hospitals, the military, the automotive industry, inventory systems, retail stores – the list goes on.