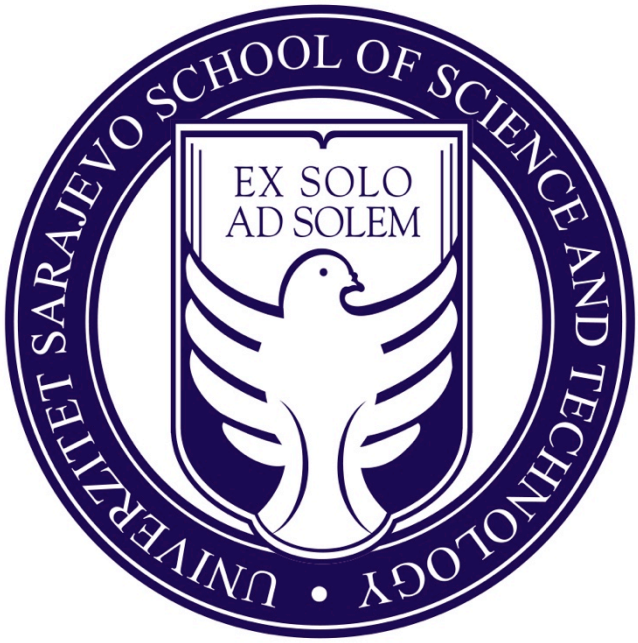


ELECTRONIC GRADE BOOK
"E-GraB"

Instructor: Husović Bakir

Authors: Šubić Amna
Lokmić Kaid
Pekarić Edah

Sarajevo School of Science and Technology

Sarajevo, October/January 2017/18

Table of Contents

1 Abstract **3**

2 Introduction **4**

3 Project Implementation **5**

3.1 Planning **5**

3.2 Analysis **5**

3.3 Design **5**

3.4 Implementation & Testing **7**

4 Lessons Learned and Conclusions **9**

5 References **10**

1 Abstract

While online grade books are a common thing in the world, this practice is still a big unknown for most schools and universities in Bosnia and Herzegovina, partially because many people do not know of their existance. That is why we decided to create an electronic grade book for a university, which could easily be converted for the use in elementary and high schools. Our electronic grade book offers readily information of students and professor in the university. Professors are able to enroll, edit and delete students from their courses, input grades and attendance, while the students are able to see all those information. In addition to students and professor, admin, or in our case registry office is able to add new students and professors in the university database, as well as add new course assignments for the professors. Moreover, admin can edit and delete currently available students, professors and course assignments. With this project we hope to improve learning experience of the students, as well as make teaching easier.

2 Introduction

Bosnia and Herzegovina has always been a few steps behind, when it comes to technology. While the world's technology is rapidly advancing and improving, and finding its way into the education, Bosnia is still relying on outdated education practices. However, thanks to the youth discovering how education can benefit from technology, and more and more young and talented people going down the computer science path, the education revolution might not be that far away. For this reason, we wanted to contribute to the education with the Electronic Grade Book project, which we hope will ease and make learning fun.

Electronic Grade Book represents a traditional paper grade book, used by professors to keep track of student attendance and grades, available on the internet that can be accessed at any place at any given time. Professors are not limited anymore to entering course related information at school only. Given this approach, we hope to prevent illegal correction of attendance and grades by students. On the other side, students are not required anymore to request from professors to see their grades. Now, they can access their records online. As a result, this could lead to improvement in learning and more motivation towards it.

Apart from making teaching and learning experience easier for the students and professors, registration of students and professors is simpler and faster. Registry official can enter all necessary data for a student or professor and submit it directly to the database, avoiding large amounts of paperwork. Editing and deleting a student or a professor is much more simpler, since now there is no need to search through thousands and thousands of files to correct or dismiss data; now you just choose and option next to the student/professor and work from there.

To conclude, E-GraB was made to aid in education and improve student-professor relationship, while also improving overall school experience and performance. E-GraB was developed using PhpStorm (PHP, HTML, CSS), MySQL, MAMP, and phpmyadmin. For this project, some icons [1], images [2], and a small amount of css design codes from the internet [3] were used.

3 Project Implementation

The development of this project went through 4 stages:

1. Planning
2. Analysis
3. Design
4. Implementation & testing

3.1 Planning

Our initial plan was to create an electronic grade book for professor, students and parents of the students. The electronic grade book will have options for each of the users. The professor can enter grades, attendance and add students to their course. Also the professor can edit attendance and grades already entered. The student is able to see all their courses, students that are enrolled in each course, grades and attendance. The parents are able to see all their children enrolled at the university, their grades, attendance, as well as messages from the faculty.

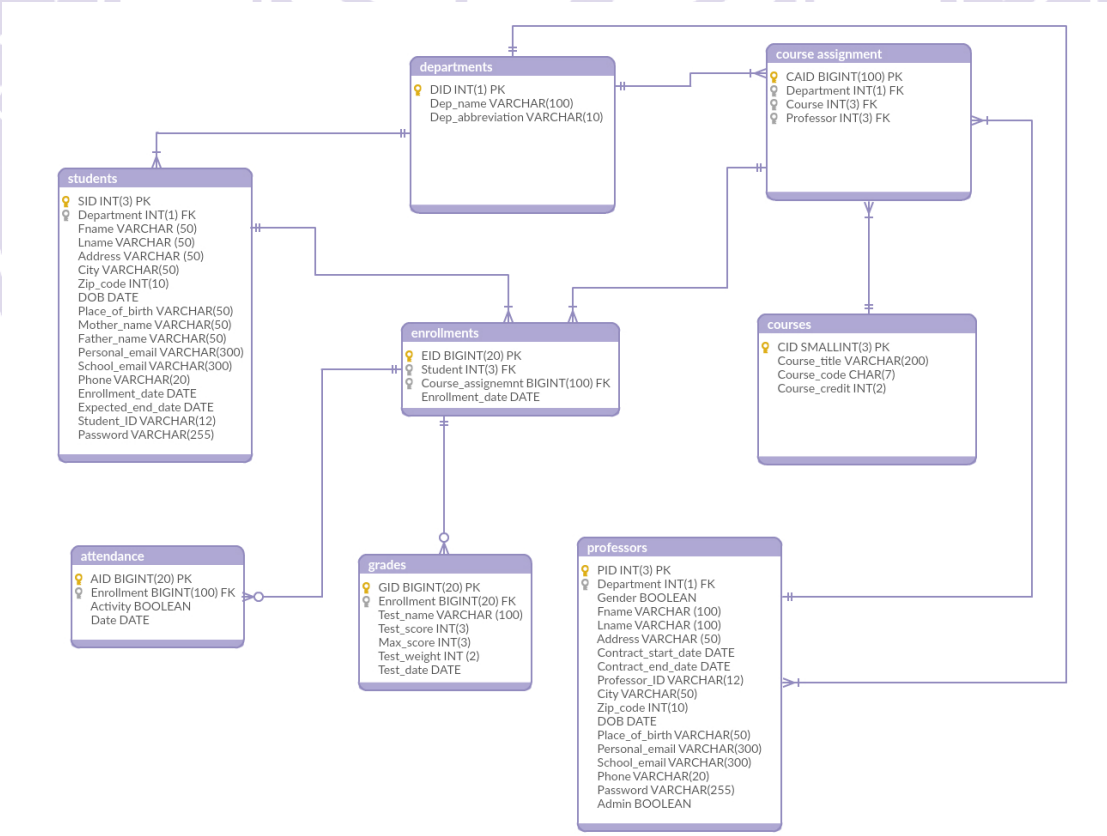
3.2 Analysis

After further analysing the plan, we came to the conclusion that we did not have enough time to implement all featues. Instead we opted out for the two main features: professors and students. Furthermore, as there are too many options that the professor has, and some were too complex to make, we decided to make these options available to the professor: add student to course, edit student enrolled at the course, delete student from the course, add attendance and add grades. The professor can only change and insert data to the course that he or she teaches.

3.3 Design

The next step was design. With the plan and analyzation done, we could start with designing our database and the website layout. The first thing was the entitity relational diagram. We started the ERD by laying out all table relationships:

1. Each department has many courses, each course can belong to many departments (since this is a many to many relationship, we solved this by inserting another table called course assignments)
2. Each course can be taught by many professors; Each professor can teach many courses (we solved this many to many by connecting professors to course assignments)
3. Each department has many students, but one student can belong to only one department.
4. Each student enrolls into many courses. Each course can have many students. (many to many relationship solved by inserting a table enrollment. Enrollment table is connected to course assignment, since that table holds the information of department, course and professor)
5. Each student has many grades and attendance. Each grade and attendance belongs only to only one student. (grades and attendance are directly connected to enrollments, since the grades and attendance are a part of enrollment of a student)



After creating an ERD and adding all needed attributes to the sketch, we created a database using MAMP and phpmyadmin. With the database ready, we then moved on to creating a design of the website. We sketched out the layout of elements and searched the internet for ideas for the profiles [4], and recreated them in HTML and added CSS styles.

3.4 Implementation & Testing

The fourth and last step was implementation of the database and converting HTML code in php to make dynamic pages, as well as testing of the results. This stage was a long and irrative process. Firstly, we started by converting all profiles into php files and creating select queries to display user information. Next we created sessions to enable login options and a connection to the database. There are three possible outcomes of login: one for the professor, one for the student and one for the admin. After a user is signed in, he/she is transferred to the appropriate page: professor profile, student profile or admin profile.

```php
if($_SERVER["REQUEST_METHOD"] == "POST") {
    // username and password sent from form
    $myemail = mysqli_real_escape_string($dbc, $_POST['email']);
    $mypassword = mysqli_real_escape_string($dbc, $_POST['pass']);
    $sql = "SELECT PID, Admin FROM professors WHERE School_email = '$myemail' AND password = '$mypassword'";
    $result = @mysqli_query($dbc, $sql);
    $row = mysqli_fetch_array($result, resulttype: MYSQLI_ASSOC);
    $count = mysqli_num_rows($result);
    if ($count == 1) {
        $_SESSION['login_user'] = $myemail;
        $_SESSION['PID'] = $row['PID'];
        $_SESSION['Admin'] = $row['Admin'];
        if ($row['Admin'] === '1') {
            header( string: "location: admin.php");
            exit();
        } else if ($row['Admin'] === '0') {
            header( string: "location: professor.php");
            exit();
        }
    } else {
        $sql = "SELECT SID FROM students WHERE School_email = '$myemail' AND Password = '$mypassword'";
        $result = mysqli_query($dbc, $sql);
        $row = mysqli_fetch_array($result, resulttype: MYSQLI_ASSOC);
        $count = mysqli_num_rows($result);
        if ($count == 1) {
            $_SESSION['login_user'] = $myemail;
            $_SESSION['SID'] = $row['SID'];
            header( string: "location: student.php");
            exit();
        } else {
            $error = "Your Login Name or Password is invalid";
            echo "<p id='error'>".$error."</p>";
        }
    }
}
```

This image shows the code for checking whose login information was entered and assigning the appropriate header location. With the profiles ready, we continued onto making forms functional. Insert, update and delete queries were created and connected to appropriate forms. These forms took a lot of testing and correcting, but in the end we got all of them right.

We have also managed to make the header and side navigation change depending on the user profile. The image below shows the code.

```php
<?php
if (!empty($_SESSION['login_user'])) {
?>
<body>
<div class="side-navigation side-bar-block side-card side-animate-left" style="..." id="mySidebar">
    <button class="side-bar-item side-button side-xlarge" onclick="side_close()">×</button>
    <?php
    if ($_SESSION['Admin'] === '1') {
        echo "<a href=\"admin.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-user-circle\" aria-hidden=\"true\"></i> Profile</a>";
        echo "<a href=\"all_students.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-graduation-cap\" aria-hidden=\"true\"></i> Students</a>";
        echo "<a href=\"all_professors.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-briefcase\" aria-hidden=\"true\"></i> Professors</a>";
        echo "<a href=\"all_course_ca.php\" class=\"side_bar_item side_button\">  <i class=\"fa fa-book\" aria-hidden=\"true\"></i> CA</a>";
    } else if ($_SESSION['Admin'] === '0') {
        echo "<a href=\"professor.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-user-circle\" aria-hidden=\"true\"></i> Profile</a>";
        echo "<a href=\"courses_p.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-tasks\" aria-hidden=\"true\"></i> Courses</a>";
    } else {
        echo "<a href=\"student.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-user-circle\" aria-hidden=\"true\"></i> Profile</a>";
        echo "<a href=\"courses_s.php\" class=\"side-bar-item side-button\"><i class=\"fa fa-tasks\" aria-hidden=\"true\"></i> Courses</a>";
    }
    ?>
</div>
<div id="page">
<header class="top-navigation">
    <div class="logo">
        <img src="Images/Logo.png" alt="Logo" id="logo">
        <h2>E-GraB</h2>
    </div>
    <div class="buttons">
        <button class="side-button side-xlarge" id="openNav" onclick="side_open()" ><i class="fa fa-outdent"></i>
        </button>
        <a href="logout.php">
            <button class="side-button side-xlarge" id="logOut"><i class="fa fa-sign-out" aria-hidden="true"></i>
            </button>
        </a>
    </div>
</header>
<?php
} else {
?>
<body class="background">
<div class="main_page">
<header class="top-navigation">
    <nav class="main-page-nav">
        <div class="logo">
            <img src="Images/Logo.png" alt="Logo" id="logo">
            <a href="main.php"><h2>E-GraB</h2></a>
        </div>
        <ul>
            <li><a href="main.php#about">About</a></li>
            <li><a href="main.php#contacts">Contacts</a></li>
        </ul>
    </nav>
</header>
```

4 Lessons Learned and Conclusions

While working on this project, we got to a glimse at what it looks like to work on real life projects. Additionally, it helped us further understand web design and development. During the development of the project, we encountered many problems. Most of the problems were query based, however with a lot of trial and error we managed to overcome them. Thanks to this, we learned the neccessary knowledge of HTML, CSS, PHP and MySQL needed to make a functional website.

In conclusion, the aim of this project is to aid students in their learning process, making it easy and fun, while in the same time making student and course managment easier for both professors and registry office. We hope to prove this, and improve this service by adding more planned options in the future.

5 References

[1] D. Gandy, "Font Awesome, the iconic font and CSS toolkit", Fontawesome.io. [Online]. Available: http://fontawesome.io/. [Accessed: 11- Dec- 2017].

[2] "Flaticon, the largest database of free vector icons", Flaticon. [Online]. Available: https://www.flaticon.com/. [Accessed: 11- Dec- 2017].

[3] "How To Create Tabs", *W3schools.com*. [Online]. Available: https://www.w3schools.com/howto/howto_js_tabs.asp. [Accessed: 11- Dec- 2017].

[4] "Resume CV Resume Template #65143", TemplateMonster. [Online]. Available: https://www.templatemonster.com/resumes/resume-cv-template-design-resume-template-65143.html. [Accessed: 06- Nov- 2017].