Student Management System - Epita International Programs

Table of Content

- Overview
- File Structure
- Dependencies
- Application Flow
- Database Schema
- Architecture
- Functionality (Code Snippets)
- Limitations

Overview

The purpose of this technical documentation is to provide developers and users with guidelines and information for building the Student Management System website. The website extracts data from the international programs database, presents the data in an organized manner, and provide functionalities to manage populations, courses, and grades.

File Structure

An overview of the directory containing the project.

```
C:.
├───doc
│       readme.md
│
├───src
│   │   login.php
│   │   logout.php
│   │   register.php
│   │   welcome.php
│   │
│   ├───grades
│   │       AIs-2020-grade.php
│   │       AIs-2021-grade.php
│   │       CS-2020-grade.php
│   │       CS-2021-grade.php
│   │       DSA-2020-grade.php
│   │       DSA-2021-grade.php
│   │       ISM-2020-grade.php
│   │       ISM-2021-grade.php
│   │       SE-2020-grade.php
│   │       SE-2021-grade.php
│   │
│   ├───phpFiles
│   │       addCourses.php
│   │       addStudent.php
│   │       dbConnection.php
│   │       deleteGrades.php
│   │       deleteStudent.php
│   │       editGrades.php
│   │       editStudent.php
│   │       searchAddCourse.php
│   │       searchCourses.php
│   │       searchStudent.php
│   │
│   ├───populations
│   │       AIs-2020.php
│   │       AIs-2021.php
│   │       CS-2020.php
│   │       CS-2021.php
│   │       DSA-2020.php
│   │       DSA-2021.php
│   │       ISM-2020.php
│   │       ISM-2021.php
│   │       SE-2020.php
│   │       SE-2021.php
│   │
│   ├───queries
│   │       attendance.php
│   │       courses.php
│   │       grades.php
│   │       populations.php
│   │       students.php
│   │
│   └───static
│       │   script.js
│       │   styles.css
│       │
│       └───resources
│               logo-epita-en.png
│
└───video
        codebaseDemo.mp4
        websiteDemo.mp4
```

Dependencies

The project uses external libraries and frameworks for its User Interface and Experience.
These are some used:

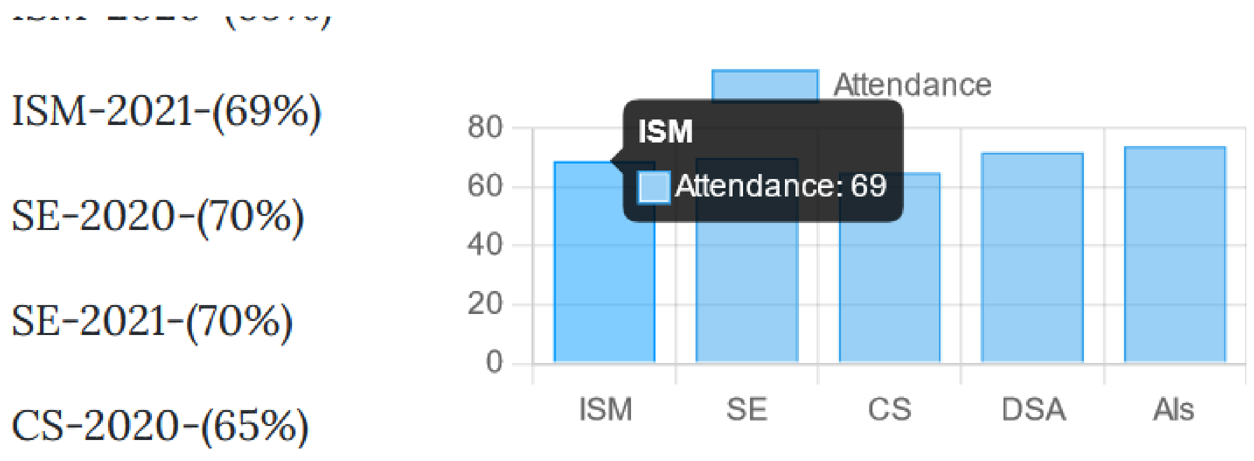
Bootstrap CSS and JavaScript

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boot
strap@5.3.0/dist/css/bootstrap.min.css">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dis
t/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe
1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bGBq41eYeM/fS" crossorigin="anon
ymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.
8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNI
PjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="a
nonymous"></script>
```

Chart.js

This JavaScript library is used to make the interactive charts seen in the welcome page.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```


ISM-2021-(69%)

SE-2020-(70%)

SE-2021-(70%)

CS-2020-(65%)

Google Fonts

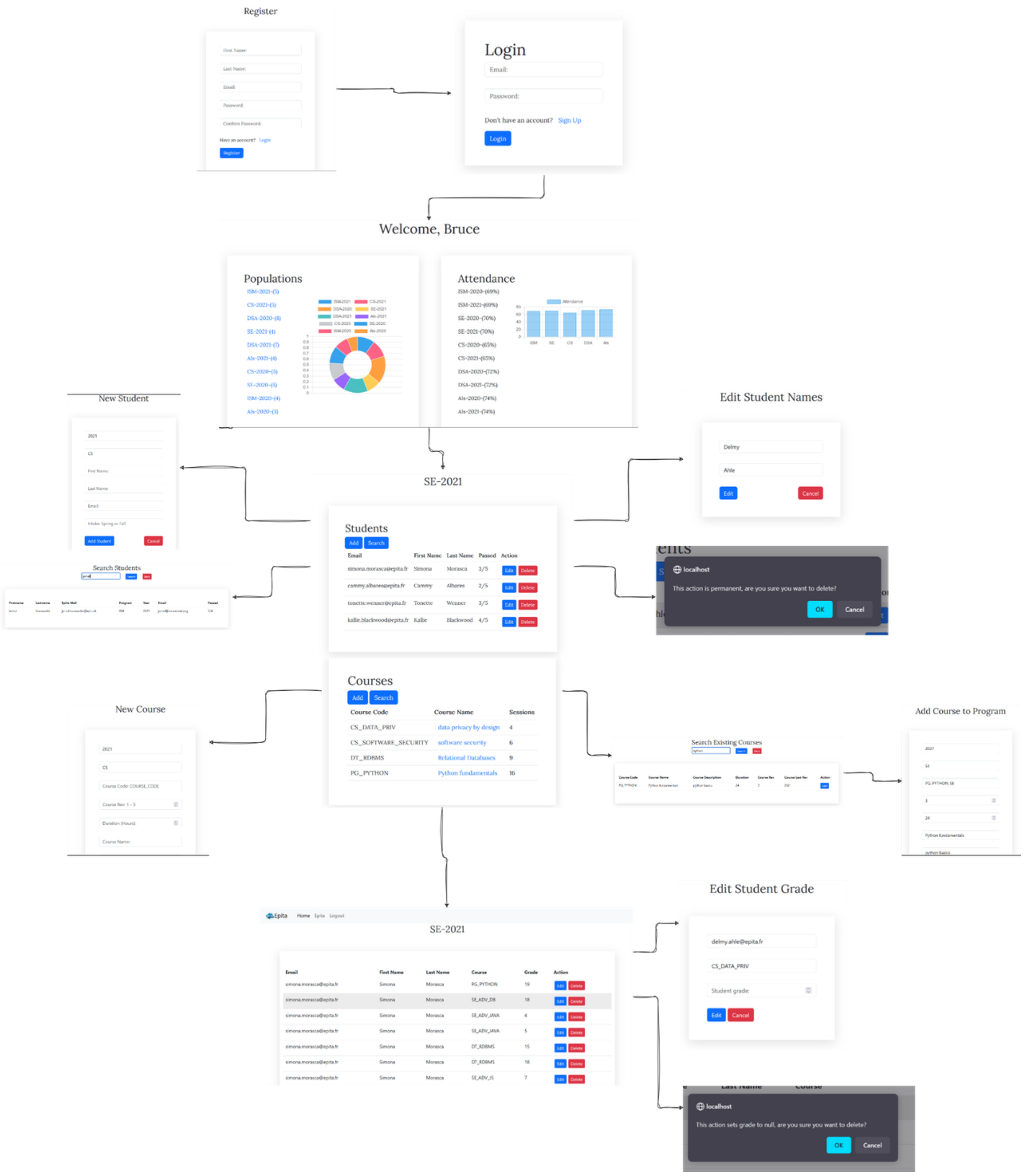
The font used is called Lora, for a smooth user interface and satisfying user experience.

```
<link rel="preconnect" href="https://fonts.googleapis.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crosso
rigin><link href="https://fonts.googleapis.com/css2?family=Lor
a&display=swap" rel="stylesheet">
```

Application Flow

An overview of the flow of the website, from registration to deleting student grades.



Login

Email:

Password:

Don't have an account? Sign Up

SE-2021

Students

Email	First Name	Last Name	Passed	Action
simona.morasca@epita.fr	Simona	Morasca	3/5	Edit Delete
cammy.albares@epita.fr	Cammy	Albares	2/5	Edit Delete
tonette.wenner@epita.fr	Tonette	Wenner	3/5	Edit Delete
kallie.blackwood@epita.fr	Kallie	Blackwood	4/5	Edit Delete

Courses

Course Code	Course Name	Sessions
CS_DATA_PRIV	data privacy by design	4
CS_SOFTWARE_SECURITY	software security	6
DT_RDBMS	Relational Databases	9
PG_PYTHON	Python fundamentals	16

localhost

This action is permanent, are you sure you want to delete?

SE-2021

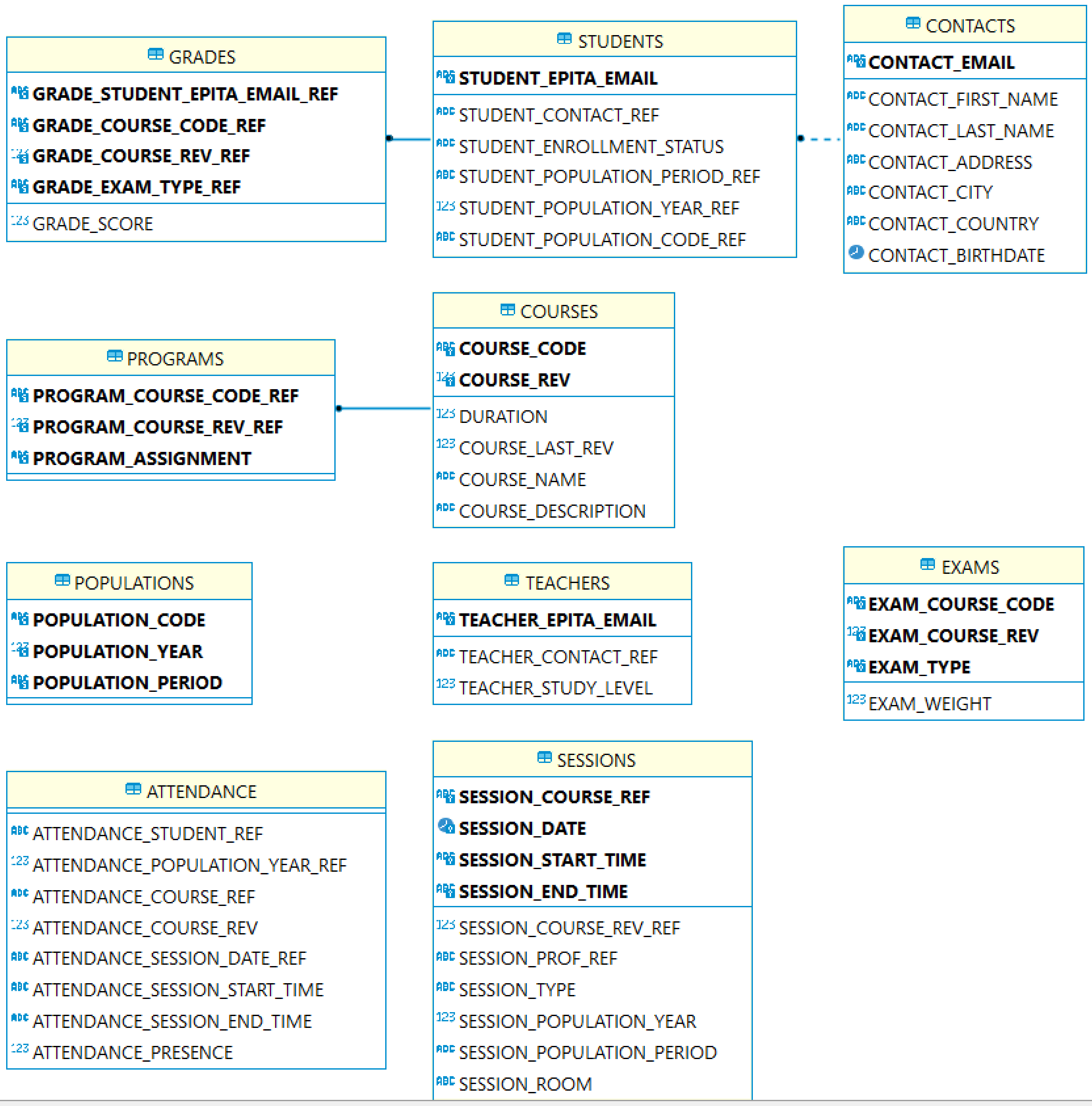
Email	First Name	Last Name	Course	Grade	Action
simona.morasca@epita.fr	Simona	Morasca	PG_PYTHON	19	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	SE_ADV_DB	18	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	SE_ADV_JAVA	4	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	SE_ADV_JAVA	5	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	DT_RDBMS	15	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	DT_RDBMS	18	Edit Delete
simona.morasca@epita.fr	Simona	Morasca	SE_ADV_JS	7	Edit Delete

localhost

This action sets grade to null, are you sure you want to delete?

Database Schema

This is the structure of the database, showing the stored data and the relationships between tables.



Architecture

The website is built using a server-side language, PHP, and a front-end framework, Bootstrap, for the user interface. It communicates with a MySQL database to Create, Read, Update and Delete data.

Functionality (Code Snippets)

Registration
When registering a new user, user data is entered into the database using prepared statements, to prevent an SQL injection attack

```
$sql = "INSERT INTO users (firstname, lastname, email, userPas
sword)
                            VALUES (?, ?, ?, ?)";
                    // initialize statement variable
$stmt = $conn->stmt_init();
$stmt->prepare($sql);
// error catching
if (! $stmt->prepare($sql)){
    die("SQL Error");
}
// bind parameters
$stmt->bind_param("ssss", $_POST["fname"], $_POST["lname"], $_
POST["email"], $passwordHash);

// execute statement
$result = $stmt->execute();
```

Authentication and Sessions
During authentication of a user, the table "users" containing the data of all registered users is queried. If the user exists, the provided password is verified using the "password_verify" function.

If the passwords match, a new session is created for the user and firstname is retrieved and stored to be used throughout the website. Upon successful authentication ,the user is redirected to the welcome page.

```
if ($_SERVER['REQUEST_METHOD'] === "POST") {
            $email = $_POST['email'];
            require "phpFiles/dbConnection.php";
            $sql = "SELECT * FROM users WHERE email='$email'";
            $result = $conn->query($sql);
            $user = $result->fetch_assoc();
            if ($user){
                if (password_verify($_POST['password'],  $user
['userPassword'])){
                    session_start();
                    $_SESSION["user"] = $user['firstname'];
                    header("Location: welcome.php");
                    die();
```

Database Connection
The database used in this project is a MySQL database.
The connection to this database is defined once and reused throughout the code as a variable “$conn”.

```
<?php
$servername = "localhost";      //server
$username = "root";             //username
$password = "";                 //password
$dbname = "epitaproject";               //database
#create db connection
$conn = new mysqli($servername, $username, $password, $dbnam
e);  //connection to MySQL
#check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Functions (Queries, Lists and Tables)
MySQL queries are made in the form of functions with corresponding arguments and return the retrieved table in the event of a successful query.

```
function grades($connection, $year, $program) {
    $result = $connection->query("SELECT g.GRADE_STUDENT_EPITA
_EMAIL_REF as EMAIL , grade_course_rev_ref as COURSE_CODE_REV
,grade_exam_type_ref as EXAM_TYPE , c.CONTACT_FIRST_NAME AS FI
RSTNAME, c.CONTACT_LAST_NAME AS LASTNAME, g.GRADE_COURSE_CODE_
REF AS COURSE, GRADE_SCORE AS GRADE
                                FROM GRADES g
                                JOIN STUDENTS s ON g.GRADE_STU
DENT_EPITA_EMAIL_REF = s.STUDENT_EPITA_EMAIL
                                JOIN CONTACTS c ON s.STUDENT_C
ONTACT_REF = c.CONTACT_EMAIL
                                WHERE s.STUDENT_POPULATION_COD
E_REF = '$program'
                                AND s.STUDENT_POPULATION_YEAR_
REF = $year ;");
    return $result;
}
```

These table results are then used to render HTML content to the page.

```
function makeGradesTable($queryResults){
    while ($row = $queryResults->fetch_assoc()) {
        echo "<tr>".
                "<td>". $row['EMAIL']. "</td>".
                "<td>". $row['FIRSTNAME']. "</td>".
                "<td>". $row['LASTNAME']. "</td>".
                "<td>". $row['COURSE']. "</td>".
                "<td>". $row['GRADE']. "</td>".
                "<td>".
                    "<a class='btn btn-primary btn-sm li' href
='../phpFiles/editGrades.php?epitaMail=". $row['EMAIL']."&"."c
ourseCode=".$row['COURSE']."&"."courseRev=".$row['COURSE_CODE_
REV']."&"."examType=".$row['EXAM_TYPE']."'>"."Edit"."</a>".
                    "<a onClick=\" javascript:return confirm
('This action sets grade to null, are you sure you want to del
ete?');\" class='btn btn-danger btn-sm li' href='../phpFiles/d
eleteGrades.php?epitaMail=". $row['EMAIL']."&"."courseCode
=".$row['COURSE']."&"."courseRev=".$row['COURSE_CODE_RE
V']."&"."examType=".$row['EXAM_TYPE']."'>"."Delete"."</a>".
                "</td>".
            "</tr>";
    }


}
```

This is the case for all Read queries; Populations, Attendance, Students, Courses and Grades.

Page Links
The links between pages of the website are made using URL arguments, which are passed while making the HTML code.
The following is an example.

```
function makeCoursesTable($queryResults, $program, $year){
    while($row = $queryResults->fetch_assoc()){
        echo "<tr>".
                "<td>". $row['COURSE_CODE']. "</td>".
                "<td>"."<a href='../grades/$program-$year-grad
e.php?year=$year&program=$program'".">". $row['COURSE_NAME'].
"</a></td>".
                "<td>". $row['SESSIONS']. "</td>".
            "</tr>";
    }
}
```

The URL arguments are then accessed using the PHP global variable GET, and passed to their respective functions.

```
<tbody>
    <?php
    require("../phpFiles/dbConnection.php");
    require("../queries/grades.php");
    $year = $_GET['year'];
    $program = $_GET['program'];
    $results = grades($conn, $year, $program);
    makeGradesTable($results);
    $conn->close();
    ?>
</tbody>
```

CRUD Actions

- **Adding new Student or Courses:**
  URL arguments of the specific course and program are retrieved using the GET global variable.
  In both cases the user fills in a form with data. Upon validation, the data is inserted into the respective tables

```
if ($_SERVER['REQUEST_METHOD'] == 'POST'){
    // get values from form
    $firstName = $_POST['fname'];
    $lastName = $_POST['lname'];
    $email = $_POST['email'];
    $year = $_POST['year'];
    $program = $_POST['program'];
    $popPeriod = strtoupper($_POST['popPeriod']);
    $epitaMail = strtolower($firstName) . "." . strtolower($la
stName) . "@epita.fr";
    $status = 'completed';
    $examType = 'Project';
    // check if values are empty
    do {
        if (empty($firstName) || empty($lastName) || empty($em
ail) || empty($popPeriod)) {
```

```
            array_push($errorMsg ,"All fields are required");
            break;
        }
        // create new students (sql)
        require("dbConnection.php");
        // insert names into contacts
        $contact = $conn->query("INSERT INTO contacts (contact
_email, contact_first_name, contact_last_name)
                                    VALUES ('$email', '$firstN
ame', '$lastName')");
        if (!$contact){
            array_push($errorMsg, "An error occured adding new
student.");
        }


        // insert into students
        $students = $conn->query("INSERT INTO students (studen
t_epita_email, student_contact_ref, student_enrollment_status,
student_population_period_ref, student_population_year_ref, st
udent_population_code_ref)
                                    VALUES ('$epitaMail', '$em
ail', '$status', '$popPeriod', $year, '$program');");
        if (!$students){
            array_push($errorMsg, "An error occured adding new
student.");
        }
        // get courses for this program
        $getCourses = "SELECT c.COURSE_CODE, c.COURSE_REV
                        FROM COURSES c
                        JOIN PROGRAMS p ON c.COURSE_CODE = p.P
ROGRAM_COURSE_CODE_REF
```

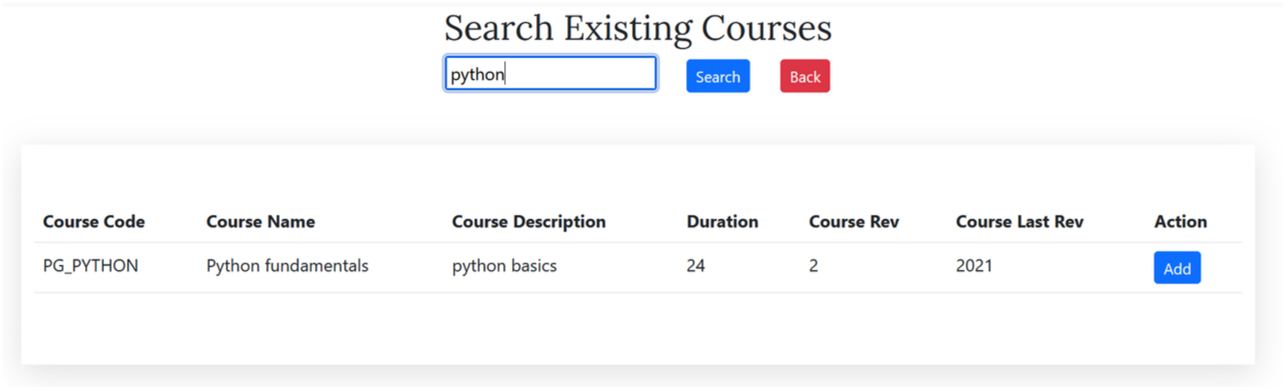
```
                        WHERE p.PROGRAM_ASSIGNMENT = '$progra
m';";
        $result = $conn->query($getCourses);
        if (!$result){
            array_push($errorMsg, "An error occured adding new
student.");
        }

        while ($row = $result->fetch_assoc()){
            $courseCode = $row['COURSE_CODE'];
            $courseRev = $row['COURSE_REV'];
            // insert values into grades table
            $insertGrades = $conn->query("INSERT INTO grades
(grade_student_epita_email_ref, grade_course_code_ref, grade_c
ourse_rev_ref, grade_exam_type_ref, grade_score)
                                                            VA
LUES('$epitaMail', '$courseCode', $courseRev, '$examType', NUL
L)");
        }
        $_SESSION['message'] = "Student added successfully";
        echo "<script>history.go(-2)</script>";


    } while (false);
```

- **Searching for an existing Student or Course**
  Upon clicking the search buttons in the program pages, the user is taken to the search page. Data is displayed in a table for all data matching the search criteria.

```
<div class="container my-5">
        <?php
        require("dbConnection.php");
        if(isset($_POST['submit'])){
            $year = $_POST['year'];
            $program = $_POST['program'];
            $search = $_POST['searchCourses'];
            if(!empty($search)){
                $sql = "SELECT course_code as courseCode,
                                course_name as courseName,
                                course_description as courseDe
scription,
                                duration,
                                course_rev as courseRev,
                                course_last_rev as courseLastR
ev
                                FROM courses
                                WHERE course_name like '%$sear
ch%'";
                $result = $conn->query($sql);
                if($result){
                    if(($result->num_rows) > 0){
                        echo "<table class='table'>".
                                    "<thead>".
                                        "<tr>".
                                            "<th>"."Course Cod
e"."</th>".
                                            "<th>"."Course Nam
e"."</th>".
                                            "<th>"."Course Des
cription"."</th>".
                                            "<th>"."Duratio
n"."</th>".
                                            "<th>"."Course Re
v"."</th>".
                                            "<th>"."Course Las
t Rev"."</th>".
```

```
                                            "<th>"."Action"."
</th>".
                                        "</tr>".
                                    "</thead>".
                                    "<tbody>";
                        while($row = $result->fetch_assoc()){
                            echo
                                    // "<tbody>".
                                        "<tr>".
                                            "<td>". $row['cour
seCode']. "</td>".
                                            "<td>". $row['cour
seName']. "</td>".
                                            "<td>". $row['cour
seDescription']. "</td>".
                                            "<td>". $row['dura
tion']. "</td>".
                                            "<td>". $row['cour
seRev']. "</td>".
                                            "<td>". $row['cour
seLastRev']. "</td>".
                                            "<td>".
                                                "<a class='btn
btn-primary btn-sm' href='../phpFiles/searchAddCourse.php?cour
seCode=". $row['courseCode'] ."&"."courseName=".$row['courseNa
me']."&"."courseDescription=".$row['courseDescription']."&"."d
uration=".$row['duration']."&"."courseRev=".$row['courseRe
v']."&"."year="."$year"."&"."program="."$program"."'>Add</a>".

                                            "</td>".
                                        "</tr>";
                        }
                        echo
```

```
                            "</tbody>".
                            "</table>";
```



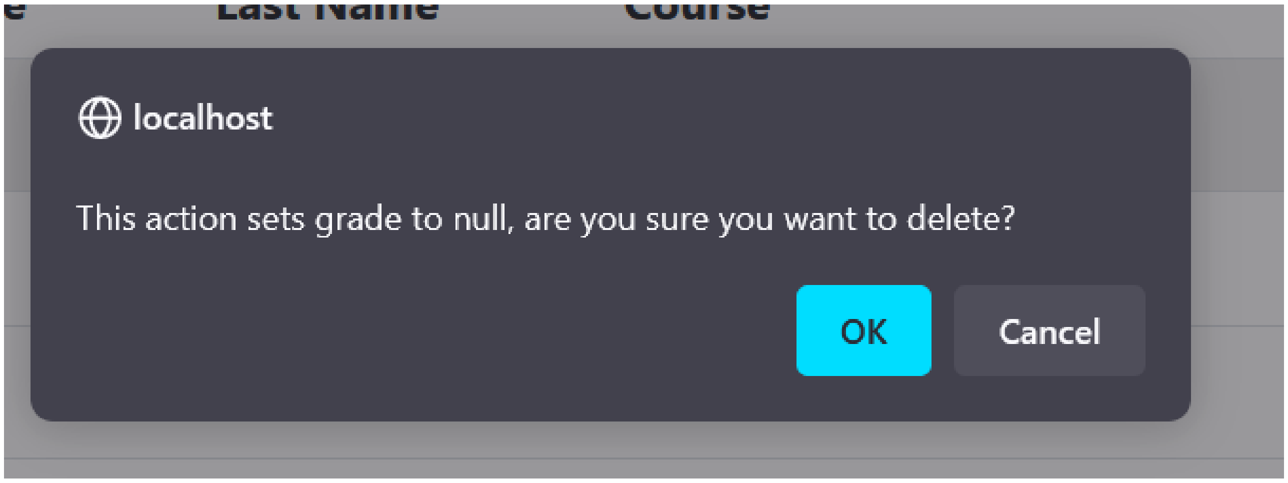
Course Code	Course Name	Course Description	Duration	Course Rev	Course Last Rev	Action
PG_PYTHON	Python fundamentals	python basics	24	2	2021	Add

Search Course 'Python'

- **Deleting an existing Student or Grade**
  There are action buttons on every row in the students and grades table. When the user clicks the delete button, a confirmation modal is shown after which the associated data is deleted or set to null in the case of grades.
  Notification messages are also shown to the user.

```
<?php
// check for url variables
if(isset($_GET['epitaMail']) && isset($_GET['courseCode']) &&
isset($_GET['courseRev']) && isset($_GET['examType'])){
    $epitaMail = $_GET['epitaMail'];
    $courseCode = $_GET['courseCode'];
    $courseRev = $_GET['courseRev'];
    $examType = $_GET['examType'];
    require("dbConnection.php");
    // set grades to null
    $grade = $conn->query("UPDATE grades
                            SET grades.grade_score = NULL
                            WHERE grades.grade_student_epita_e
mail_ref = '$epitaMail'
                            AND grades.grade_course_code_ref =
'$courseCode'
```

```
                                        AND grades.grade_course_rev_ref = 
$courseRev
                                        AND grades.grade_exam_type_ref = 
'$examType';");

}
$_SESSION['message'] = "Student grade deleted successfully!";
echo "<script>history.go(-1)</script>";
?>
```


localhost

This action sets grade to null, are you sure you want to delete?

Confirmation modal for deleting grades

- **Editing Student Names or Grades :**
  The user can also click the edit action button , which would take them to a student or grade editing page.
  The form is filled and validated and then updated in the database

```
// update grade
           require("dbConnection.php");
           $grade = $conn->query("UPDATE grades
                                              SET grades.grade_score = $grad
e
```

```
                                WHERE grades.grade_student_epi
ta_email_ref = '$epitaMail'
                                AND grades.grade_course_code_r
ef = '$courseCode'
                                AND grades.grade_course_rev_re
f = $courseRev
                                AND grades.grade_exam_type_ref
= '$examType';");
        if (!$grade){
            array_push($errorMsg, "An error occured at grade
s");
            break;
        }
        // success message
        $_SESSION['message'] = "Student grade edited successfu
lly";
        echo "<script>history.go(-2)</script>";
```

Limitations

- Sessions can only be in consecutive days.
  Because of the nature of the database and my skill level, sessions are currently only able to be added in consecutive days when adding a new course to the database.

```
$originalDate = $sessionDate;
        // insert first session
        $firstSession = $conn->query("INSERT INTO sessions (se
ssion_course_ref, session_course_rev_ref, session_date, sessio
n_start_time, session_end_time, session_population_year)
                                    VALUES ('$courseCode', $co
urseRev, '$originalDate', '$sessionStartTime', '$sessionEndTim
e', '$year')");
```

```
        if (!$firstSession){
            array_push($errorMsg, "An error occured at first s
ession");
            break;
        }
        // increment the date by the number of sessions
        for ($num = 1; $num < $numOfSession; $num++){
            $tempDate = strtotime("+1 day", strtotime("$origin
alDate"));
            $newDate = date("Y-m-d", $tempDate);
            $sessions = $conn->query("INSERT INTO sessions (se
ssion_course_ref, session_course_rev_ref, session_date, sessio
n_start_time, session_end_time, session_population_year)
                                    VALUES ('$courseCode', $co
urseRev, '$newDate', '$sessionStartTime', '$sessionEndTime',
$year)");

            $originalDate = $newDate;
            if (!$sessions){
                array_push($errorMsg, "An error occured at ses
sions");
                break;
            }
        }
```

- Data can be rendered without reloading the page using AJAX calls