EUROPEAN UNIVERSITY OF LEFKE

Faculty of Engineering

Department of Software Engineering



Name: Lawal Abdulganiy

Student No: 20140002

Department:Software Engineering

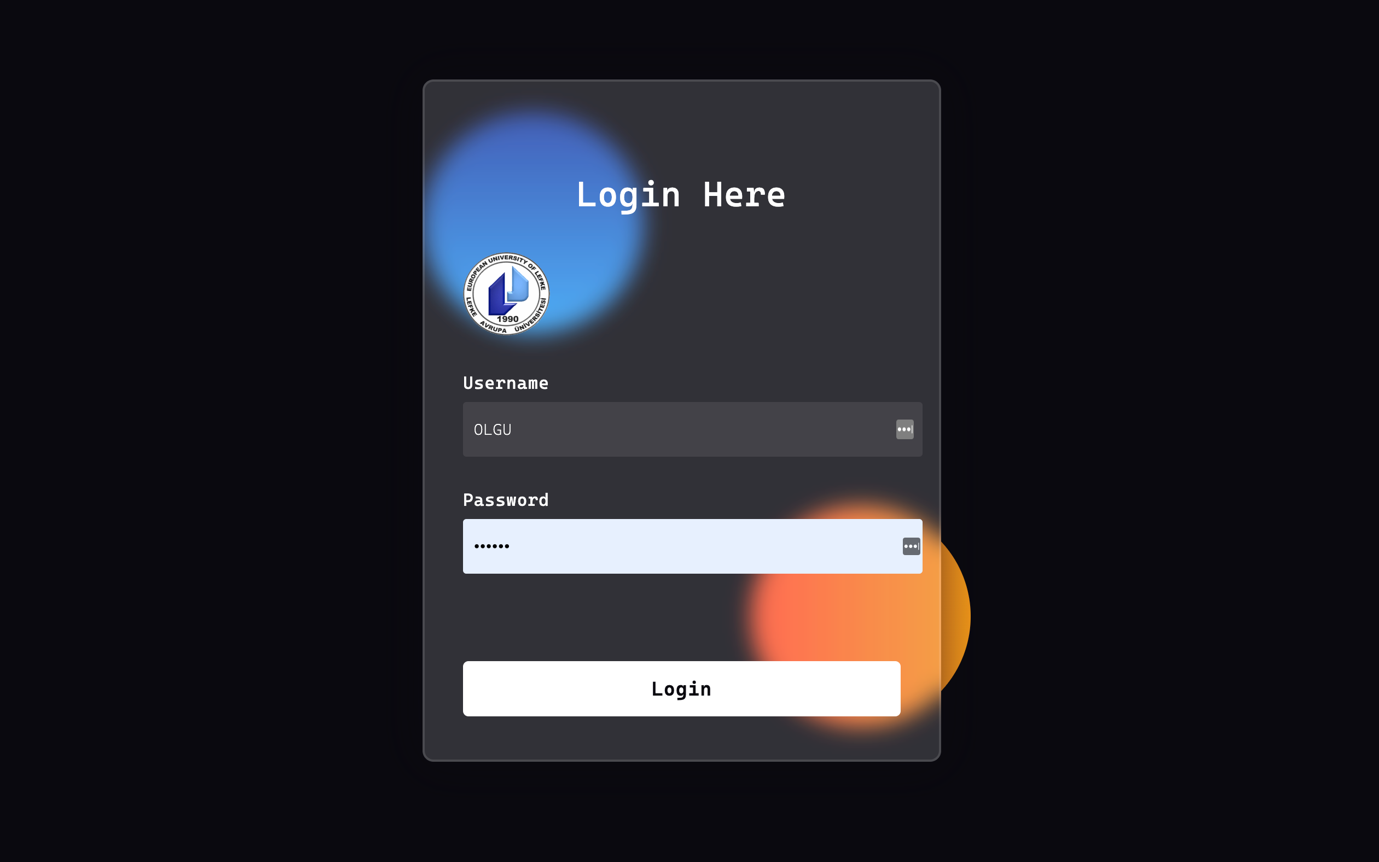
Facaulty: Engineering

Submitted to Dr. Vesile Evrim

NOTE: I Implemented the Bonus Part

**User InterFace**

**Login Page**

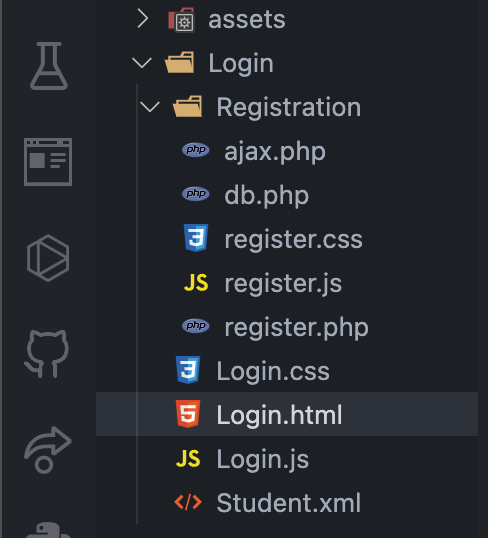
****

**Registration Page**

****

**File Structure**

**This shows a ui of my file structure**

****

**Creating the Database**

**This create the database in phpmyadmin Note: I am using XAMPP**

// credentials for login into myphp admin

*<?php*

*$host* ***=*** *'localhost';*

*$db\_name* ***=*** *'student\_info\_db';*

*$username* ***=*** *'root';*

*$password* ***=*** *'';*

*// Create a database connection*

*$connection* ***=******new*** *PDO("mysql:host=$host;dbname=$db\_name;charset=utf8", $username, $password);*

*// Create the student table if it doesn't exist*

*$createTableQuery* ***=*** *"*

*CREATE TABLE IF NOT EXISTS student (*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*student\_number VARCHAR(10) NOT NULL,*

*name VARCHAR(50) NOT NULL,*

*lastname VARCHAR(50) NOT NULL,*

*photo VARCHAR(255) NOT NULL,*

*department VARCHAR(50) NOT NULL,*

*cgpa DECIMAL(3, 2) NOT NULL,*

*semester VARCHAR(10) NOT NULL*

*)*

*";*

*$connection****->****exec($createTableQuery);*

*// Insert some student.xml data*

*$dummyData* ***=*** *[*

*[*

*'student\_number'* ***=>*** *'140306',*

*'name'* ***=>*** *'OLGU',*

*'lastname'* ***=>*** *'ACUN',*

*'photo'* ***=>*** *'assets/Background.jpeg',*

*'department'* ***=>*** *'BSc IN COMPUTER ENGINEERING',*

*'cgpa'* ***=>*** *3.75,*

*'semester'* ***=>*** *'5th'*

*],*

*[*

*'student\_number'* ***=>*** *'150030',*

*'name'* ***=>*** *'CENK ALEN',*

*'lastname'* ***=>*** *'SEHIROGLU',*

*'photo'* ***=>*** *'assets/Background.jpeg',*

*'department'* ***=>*** *'BSc IN COMPUTER ENGINEERING',*

*'cgpa'* ***=>*** *3.60,*

*'semester'* ***=>*** *'6th'*

*],*

*[*

*'student\_number'* ***=>*** *'164602',*

*'name'* ***=>*** *'FUNGAI',*

*'lastname'* ***=>*** *'GADZIRA',*

*'photo'* ***=>*** *'assets/Background.jpeg',*

*'department'* ***=>*** *'BSc IN SOFTWARE ENGINEERING',*

*'cgpa'* ***=>*** *3.85,*

*'semester'* ***=>*** *'7th'*

*],*

*[*

*'student\_number'* ***=>*** *'170086',*

*'name'* ***=>*** *'MERTCAN',*

*'lastname'* ***=>*** *'ARAS',*

*'photo'* ***=>*** *'assets/Background.jpeg',*

*'department'* ***=>*** *'BSc IN COMPUTER ENGINEERING',*

*'cgpa'* ***=>*** *3.70,*

*'semester'* ***=>*** *'6th'*

*],*

*[*

*'student\_number'* ***=>*** *'170090',*

*'name'* ***=>*** *'MUSTAFA BERKIN',*

*'lastname'* ***=>*** *'CIHAN',*

*'photo'* ***=>*** *'assets/Background.jpeg',*

*'department'* ***=>*** *'BSc IN COMPUTER ENGINEERING',*

*'cgpa'* ***=>*** *3.80,*

*'semester'* ***=>*** *'7th'*

*]*

*];*

*// This shows what database query it is going to perform*

*$insertQuery* ***=*** *"*

*INSERT INTO student (student\_number, name, lastname, photo, department, cgpa, semester)*

*VALUES (:student\_number, :name, :lastname, :photo, :department, :cgpa, :semester)*

*";*

*$statement* ***=*** *$connection****->****prepare($insertQuery);*

*// This performs each query on the individual student.xml dummy data*

***foreach*** *($dummyData* ***as*** *$data) {*

*$statement****->****execute($data);*

*}*

*?>*

**DataBase Image**

****

**Login.Html Validation**

*<div class="form-group">*

*<label for="username">*Username*</label>*

*<input*

*type="text"*

*placeholder="Name or Surname"*

*id="username"*

*oninput="validateUsername()"*

*This calls a function that validate the username anytime the user is typing. This function notifies the user if the user is inputing any non - alphabetic characters*

*/>*

*<div id="usernameError" class="error-message"></div>*

*</div>*

*<div class="form-group">*

*<label for="password">*Password*</label>*

*<input*

*type="password"*

*placeholder="Student NO"*

*id="password"*

*This calls a function that checks if the user password is less than 6 characters*

*oninput="validatePassword()"*

*/>*

*<div id="passwordError" class="error-message"></div>*

*</div>*

*<div class="form-group">*

*This function handles the login logic*

*<button type="button" id="loginButton"*

*onclick="loadDoc()">*

Login

*</button>*

**Login.Js**

**This descripes all the above function implementations**

*// This reads the xml file and returns the xml information*

*function readXml(xml) {*

*const students* ***=*** *xml.getElementsByTagName("Student");*

*const studentData* ***=*** *[];*

***for*** *(let i* ***=*** *0; i* ***<*** *students.length; i****++****) {*

*const studentNo* ***=***

*students[i].getElementsByTagName("StudentNo")[0].textContent;*

*const name* ***=*** *students[i].getElementsByTagName("Name")[0].textContent;*

*const surname* ***=*** *students[i].getElementsByTagName("Surname")[0].textContent;*

*studentData.push({*

*studentNo,*

*name,*

*surname,*

*});*

*}*

***return*** *studentData;*

*}*

*// Validate login*

*function validateLogin(username, password, studentData) {*

*// Loops through the student data*

***for*** *(let i* ***=*** *0; i* ***<*** *studentData.length; i****++****) {*

*const student* ***=*** *studentData[i];*

*// Checks if the current student matches any student in the xml file*

***if*** *(*

*(username* ***===*** *student.name* ***||*** *username* ***===*** *student.surname)* ***&&***

*password* ***===*** *student.studentNo*

*) {*

***return*** *true; // Login successful*

*}*

*}*

***return*** *false; // Login failed*

*}*

*function validateUsername() {*

*const usernameInput* ***=*** *document.getElementById("username");*

*const username* ***=*** *usernameInput.value;*

*// Replaces all the non-alphabetic characters with an empty string from the username*

*const newUsername* ***=*** *username.replace(/[****^****A-Za-z\s]/****g****, "");*

*// Update the input field value with the new value*

*usernameInput.value* ***=*** *newUsername;*

*// Display an Error if the username is not ideal*

*const usernameError* ***=*** *document.getElementById("usernameError");*

***if*** *(username* ***!==*** *newUsername* ***||******!****/****^****[A-Za-z\s]****+$****/.test(username)) {*

*usernameError.textContent* ***=*** *"Please enter only alphabetic characters";*

*}* ***else*** *{*

*usernameError.textContent* ***=*** *"";*

*}*

*}*

*function validatePassword() {*

*const passwordInput* ***=*** *document.getElementById("password");*

*const password* ***=*** *passwordInput.value;*

*// Check if password length is less than 6*

*const passwordError* ***=*** *document.getElementById("passwordError");*

***if*** *(password.length* ***<*** *6) {*

*passwordError.textContent* ***=*** *"Password must be at least 6 characters long";*

*}* ***else*** *{*

*passwordError.textContent* ***=*** *"";*

*}*

*}*

*// This loads the xml file and is responsible for seguing user to registration page*

*function login() {*

*const username* ***=*** *document.getElementById("username").value;*

*const password* ***=*** *document.getElementById("password").value;*

*// Load and parse the XML data*

*const xhttp* ***=******new*** *XMLHttpRequest();*

*xhttp.onreadystatechange* ***=*** *function () {*

***if*** *(this.readyState* ***===*** *4* ***&&*** *this.status* ***===*** *200) {*

*const xmlDoc* ***=*** *this.responseXML;*

*const studentData* ***=*** *readXml(xmlDoc);*

*// Validate login credentials*

*const loginSuccess* ***=*** *validateLogin(username, password, studentData);*

*// If successful segue the user to the register.php page with the username and password this will be useful in the register.php page*

***if*** *(loginSuccess) {*

*window.location.href* ***=*** *`Registration/register.php?username=${username}&password=${password}`;*

*}* ***else*** *{*

*alert("Failed Login. Please check your username and password.");*

*}*

*}*

*};*

*xhttp.open("GET", "Student.xml", true);*

*xhttp.send();*

*}*

**Register.PHP**

*<?php*

*// Check if the user is logged in*

***if*** *(****!****isset($\_GET['username'])* ***||******!****isset($\_GET['password'])) {*

*// Redirect to the login page*

*header("Location: Login.html");*

***exit****();*

*}*

*// Retrieve the username and password from URL parameters*

*$username* ***=*** *$\_GET['username'];*

*$password* ***=*** *$\_GET['password'];*

*// // credentials for login into my database*

*$servername* ***=*** *"localhost";*

*$dbname* ***=*** *"student\_info\_db";*

*$username\_db* ***=*** *"root";*

*$password\_db* ***=*** *"";*

*// This connect to the existing database*

*$conn* ***=******new*** *PDO("mysql:host=$servername;dbname=$dbname", $username\_db, $password\_db);*

*// Prepare and execute the SQL query to fetch user information*

*$stmt* ***=*** *$conn****->****prepare("****SELECT******\*******FROM*** *student* ***WHERE******name******=*** *?* ***OR*** *lastname* ***=*** *?* ***AND*** *student\_number* ***=*** *?");*

*$stmt****->****execute([$username, $username, $password]);*

*$studentData* ***=*** *$stmt****->****fetchAll(PDO****::****FETCH\_ASSOC);*

*?>*

*<!DOCTYPE html>*

*<html>*

*<head>*

*<title>*Registration*</title>*

*<link rel="stylesheet" type="text/css" href="register.css" />*

*</head>*

*<body>*

*<div class="container">*

*<div class="section">*

*<h2>*User Information*</h2>*

*<!-- // Loop through the user information gotten from the database -->*

*<?php* ***foreach*** *($studentData* ***as*** *$student) : ?>*

*<div class="user-card">*

*<div class="user-image">*

*<!-- // I am using a picture that is stored in the asset folder, it is retrieved from my database -->*

*<img src="/<?php echo $student['photo']; ?>" alt="Profile Image">*

*</div>*

*<div class="user-details">*

*<p><span>*Student Number:*</span>* *<?php echo $student['student\_number']; ?></p>*

*<p><span>*Name:*</span>* *<?php echo $student['name']; ?></p>*

*<p><span>*Surname:*</span>* *<?php echo $student['lastname']; ?></p>*

*<p><span>*Department:*</span>* *<?php echo $student['department']; ?></p>*

*<p><span>*CGPA:*</span>* *<?php echo $student['cgpa']; ?></p>*

*<p><span>*Semester:*</span>* *<?php echo $student['semester']; ?></p>*

*</div>*

*</div>*

*<?php* ***endforeach****; ?>*

*</div>*

*<div class="section">*

*<h2>*Course Selection*</h2>*

*<ul id="courseList">*

*<!-- Courses will be dynamically added here -->*

*</ul>*

*<!-- This functionality calls the addCourses function in register.js-->*

*<button onclick="addCourses()">*Add*</button>*

*</div>*

*<div class="section">*

*<h2>*User Courses*</h2>*

*<ul id="UserCourseList">*

*<!-- Selected courses will be dynamically added here -->*

*</ul>*

*</div>*

*<div class="section comment-section">*

*<h2>*Comment*</h2>*

*<textarea id="comment" rows="4" cols="50"></textarea>*

*<!-- This calls the submit course function in register.js -->*

*<button onclick="submitComment()">*Submit*</button>*

*</div>*

*</div>*

*<script src="register.js"></script>*

*</body>*

*</html>*

**Register.JS**

*// Function to fetch the course list from the server*

*function fetchCourseList() {*

*const courseList* ***=*** *document.getElementById("courseList");*

*// Send an AJAX request to fetch the course list*

*const xhr* ***=******new*** *XMLHttpRequest();*

*xhr.onreadystatechange* ***=*** *function () {*

***if*** *(xhr.readyState* ***===*** *XMLHttpRequest.DONE) {*

***if*** *(xhr.status* ***===*** *200) {*

*// Parse the JSON response*

*const response* ***=*** *JSON.parse(xhr.responseText);*

*// Clear the course list*

*courseList.innerHTML* ***=*** *"";*

*// This loops through the json response and display each item with a checkbbox*

*response.forEach((course) => {*

*const listItem* ***=*** *document.createElement("li");*

*const checkbox* ***=*** *document.createElement("input");*

*checkbox.type* ***=*** *"checkbox";*

*checkbox.value* ***=*** *course;*

*listItem.appendChild(checkbox);*

*listItem.appendChild(document.createTextNode(course));*

*courseList.appendChild(listItem);*

*});*

*}* ***else*** *{*

*console.error("Error fetching course list:", xhr.status);*

*}*

*}*

*};*

*xhr.open("GET", "ajax.php?action=getCourseList", true);*

*xhr.send();*

*}*

*// Function to add selected courses to the user courses section and a delete functionality*

*function addCourses() {*

*const courseList* ***=*** *document.getElementById("courseList");*

*const UserCourses* ***=*** *document.getElementById("UserCourseList");*

*// Get all selected checkboxes*

*const checkboxes* ***=*** *courseList.querySelectorAll(*

*'input[type="checkbox"]:checked'*

*);*

*// Check if the number of course is > than 5*

***if*** *(checkboxes.length* ***+*** *UserCourses.childElementCount* ***>*** *5) {*

*alert("You are only allowed to add 5 courses.");*

***return****;*

*}*

*// Iterate over the selected checkboxes*

*checkboxes.forEach((checkbox) => {*

*// Create a list item for the selected course*

*const listItem* ***=*** *document.createElement("li");*

*const courseName* ***=*** *checkbox.parentNode.textContent.trim();*

*listItem.innerText* ***=*** *courseName;*

*// Create a delete button for the courses in the usercourse list section*

*const deleteButton* ***=*** *document.createElement("button");*

*deleteButton.innerText* ***=*** *"Delete";*

*// Adding it’s functionality*

*deleteButton.addEventListener("click", function () {*

*// Remove the course from the*

*UserCourses.removeChild(listItem);*

*// Append the course back to the course list with is defaulted ui*

*const courseItem* ***=*** *document.createElement("li");*

*const courseCheckbox* ***=*** *document.createElement("input");*

*courseCheckbox.type* ***=*** *"checkbox";*

*courseCheckbox.value* ***=*** *courseName;*

*courseItem.appendChild(courseCheckbox);*

*courseItem.appendChild(document.createTextNode(courseName));*

*courseList.appendChild(courseItem);*

*});*

*// Out side of the delete functionality*

*// Append the delete button to the list item*

*listItem.appendChild(deleteButton);*

*// Add the course to the usercourse section with the coursename and a delete button*

*UserCourses.appendChild(listItem);*

*// Remove the course checkbox from the course list*

*checkbox.parentNode.parentNode.removeChild(checkbox.parentNode);*

*});*

*}*

*function submitComment() {*

*var commentText* ***=*** *document.getElementById("comment");*

*// Get the comment value*

*var comment* ***=*** *commentText.value;*

*// Validate if the comment is not empty*

***if*** *(comment.trim()* ***!==*** *"") {*

*// Prepare the mailto link*

*var mailtoLink* ***=***

*"mailto:Vesile@gmail.com"* ***+***

*"?subject=Comment"* ***+***

*"&body="* ***+***

*encodeURIComponent(comment);*

*// Open the default email client with the mailto link*

*window.location.href* ***=*** *mailtoLink;*

*// Clear the comment textarea*

*commentText.value* ***=*** *"";*

*}* ***else*** *{*

*// Display an alert for empty comment*

*alert("Please enter a comment");*

*}*

*}*

*// Fetch the course list when the page loads*

*window*.*onload* ***=*** *function () {*

*fetchCourseList();*

*}*;

**AJAX.PHP**

*<?php*

*// This is responsibble for returning the course list to fetchCourseList() function in register.js*

***if*** *($\_GET['action']* ***===*** *'getCourseList') {*

*// An Array of example courses*

*$courseList* ***=*** *[*

*"Introduction to Algorithms",*

*"Data Structures and Algorithms",*

*"Object-Oriented Programming",*

*"Software Engineering Principles",*

*"Web Development",*

*"Mobile App Development",*

*"Database Systems",*

*"Operating Systems",*

*"Network Security",*

*"Machine Learning"*

*];*

*// Return the course list as JSON*

*echo json\_encode($courseList);*

*}*

*?>*

**Initial Setup**

****

**The user can mark some courses**

****

**When the add button is clicked it append it to the User Course Section with a buttun to it’s side**

**When you clicked on a button it deletes a course and append it to the course selection section**

**Submit UI**

****