

# CS 306 Midterm Project Report



**Building and Designing a full website using:**  
HTML/CSS/JavaScript  
jQuery/AJAX/Bootstrap  
&  
**Testing PHP Knowledge through MySQL Database**

*CS 306  
Fall Semester  
Academic year: 2019-2020  
Worked By: Daniel KALEMI  
Received By: Iosif Victoratos*

## TABLE OF CONTENTS

This report will contain:

<b>1. DESCRIPTION.....</b>	<b>3</b>
<b>2. TASKS.....</b>	<b>2</b>
<b>    2.1 SCOPE OF PROJECT.....</b>	<b>3-5</b>
<b>    2.2 UI DESIGN.....</b>	<b>4-11</b>
<b>    2.3 CODE DEVELOPMENT.....</b>	<b>12-16</b>
<b>    2.4 BUSINESS LOGIC.....</b>	<b>16-17</b>

## 1. Project DESCRIPTION

This program in Distributed Applications & Web Development was intended to practice on the learning material as part of the CS306 course. Specifically, the purpose of this project is to create a self-contained, hands-on project integrating all the knowledge gained so far on PHP, MySQL AJAX, jQuery, JS, CSS & HTML.

The name of my project is: “ACT MOODLE”

This website serves as a sub-website of a college like ACT and the students could use it to complete short quizzes online and get marks/points accordingly.

In this project description I’m going to walk you through all the steps and the features of this web application.

## 2. TASKS

The requirements of the program were:

A fully functional Website with HTML pages, external CSS and JavaScript file. What’s important is that covering the PHP requirements we also apply and add to what we have learned about HTML, CSS JavaScript and create an easy-to-read and navigated site.

The site will contain a home page (index.html) and 6 additional pages.

### ACT MOODLE WEBSITE

I have created a fully responsive 11-pages+ website for an online course/quizzes web app.

The pages are not static, they are dynamic due to jQuery Toggle and Bootstrap.

So with the toggle on the accounts page, the web app becomes with nearly 16 pages (considering the walkthrough of the quizzes, the scoring, the leaderboard and the functionality of the admin page)

The pages include:

**HOME (index.html)**

**BLOG (blog.html)**

**MOODLE (index.php)**

**ADMIN LOG IN (admin.php/login/account/logout/update.php)**

**DEVELOPER PAGE (account.php)**

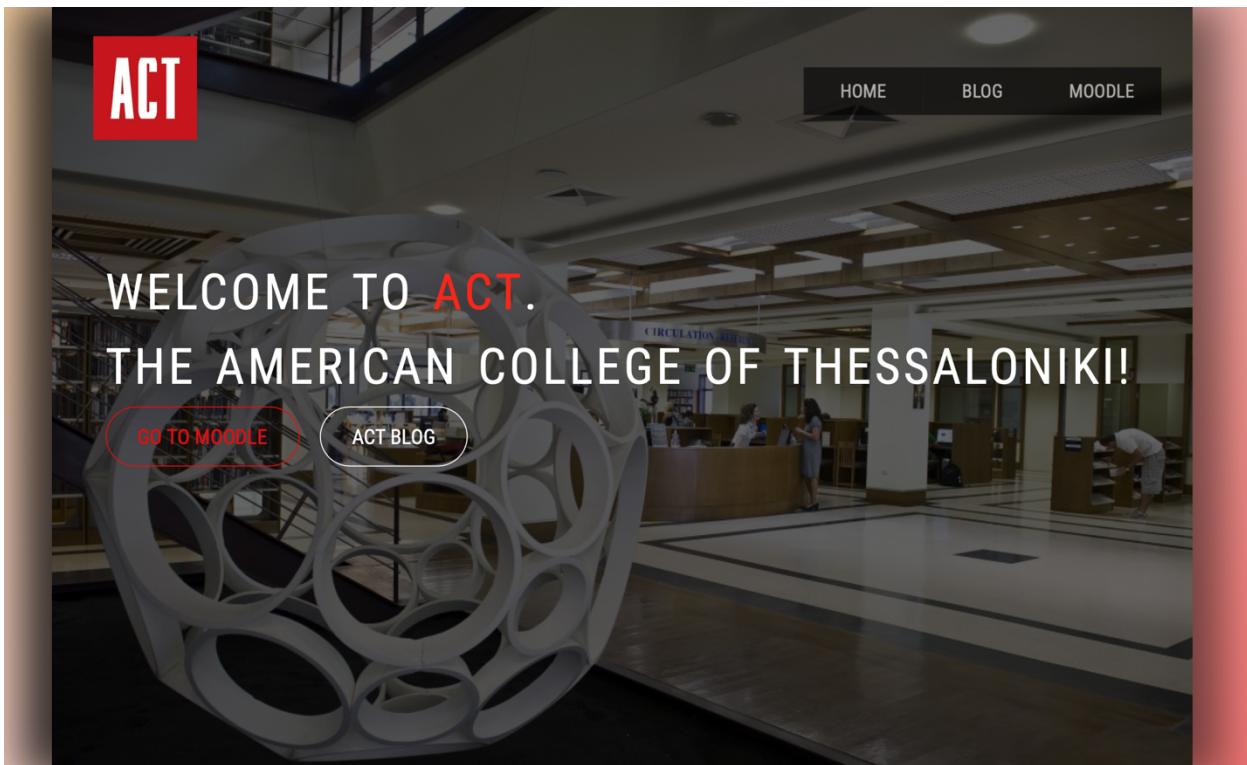
**FEEDBACK (feedback.php/feed.php/account.php)**

**Dash (for the dashboard on account.php & admin)**

## ***DESIGN WALKTHROUGH UI***

By building this project, I was able to follow through the steps of creating a website where the student starts:

1. ***Going to the HOME page***



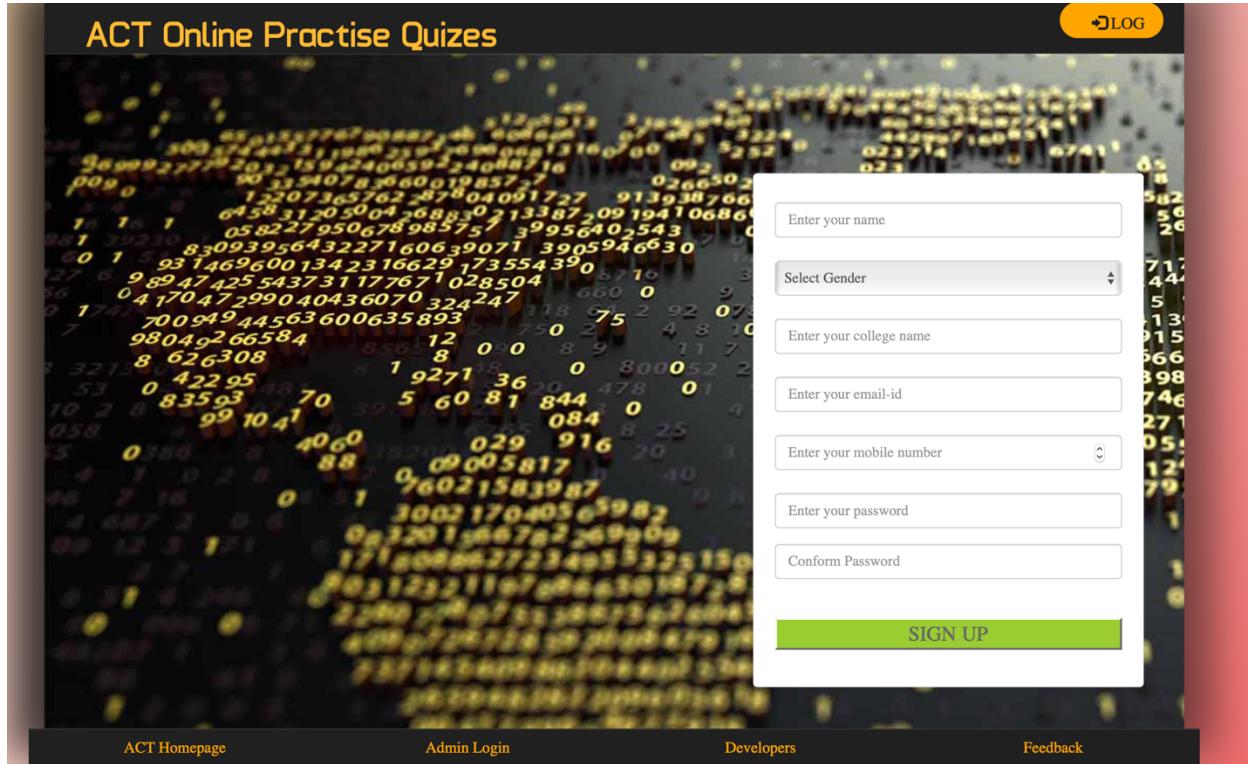
The student is greeted with the main page and a video.  
They can choose to go to MOODLE or follow the BLOG.

2. ***Going to the BLOG page***



Here the students can have a look at the new developments and news of the college.

3. Go to the student portal (ACT MOODLE)



As we can see from the picture above, this page offers many options.

1. If you are an existing user, you can log in to your account
  2. If you are new, you can create an account by filling in the presented form
  3. If you are the admin of the page, log in to see the stats
    - Email: [admin@admin.com](mailto:admin@admin.com)
    - Password: admin
  4. If you want to link with the developer, click to the developer tab and look at the details or click on the name and it will take you to my linkedin account.
  5. For any further questions, the students can refer the feedback section, where they can sent out a form to the admin of the web app with their enquires.
  6. Lastly, you have the option to return to the homepage.
4. My Profile page (account.php)

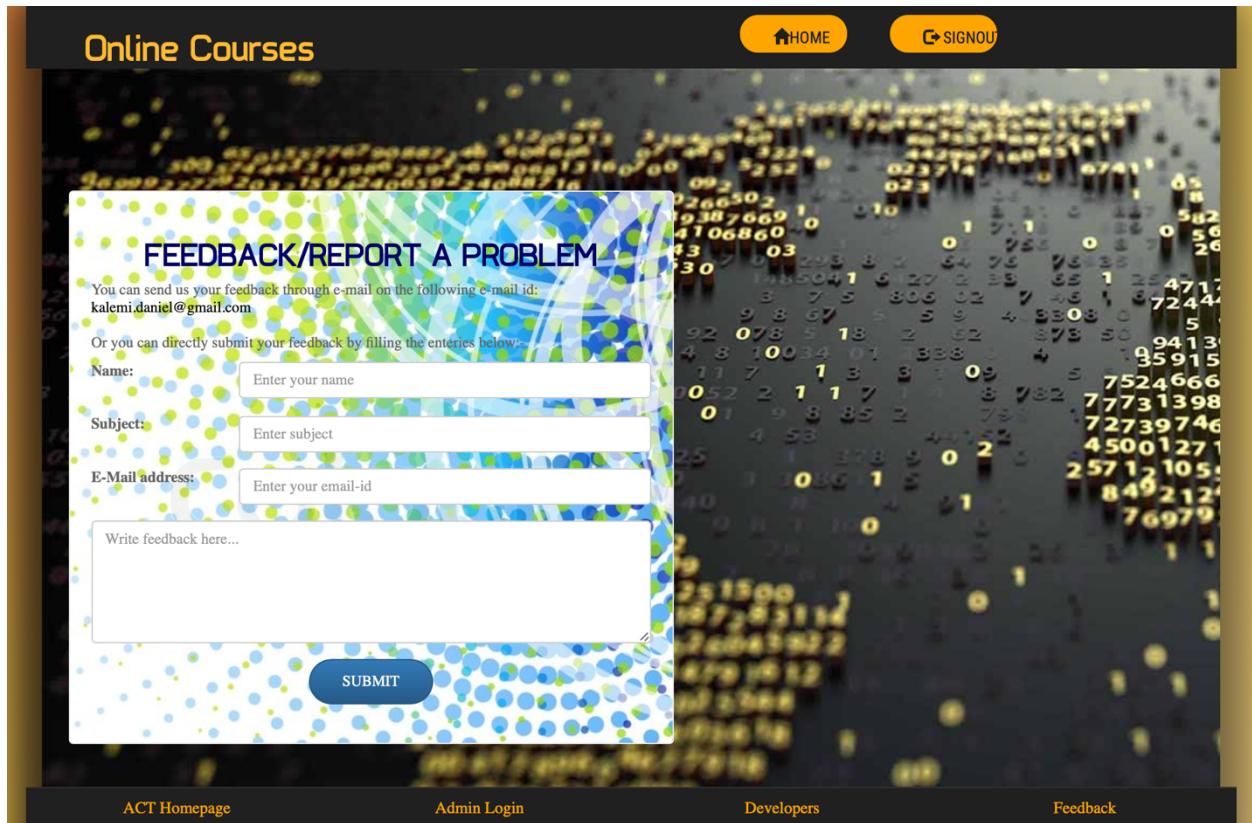
The screenshot shows the ACT Moodle user interface. At the top, there's a navigation bar with links for Home, History, Ranking, and Signout. Below the navigation bar is a search bar with a placeholder 'Enter tag' and a 'SEARCH' button. The main content area displays a table of quizzes:

S.N.	Topic	Total question	Marks	Time limit	Action
1	Linux :vi Editor	5	10	10 min	
2	Linux:startup	5	10	10 min	
3	Networking	2	4	5 min	
4	C++ Coding	2	4	5 min	
5	Php Coding	2	4	5 min	
6	Linux : File Managment	2	4	5 min	

At the bottom of the page, there are links for ACT Homepage, Admin Login, Developers, and Feedback.

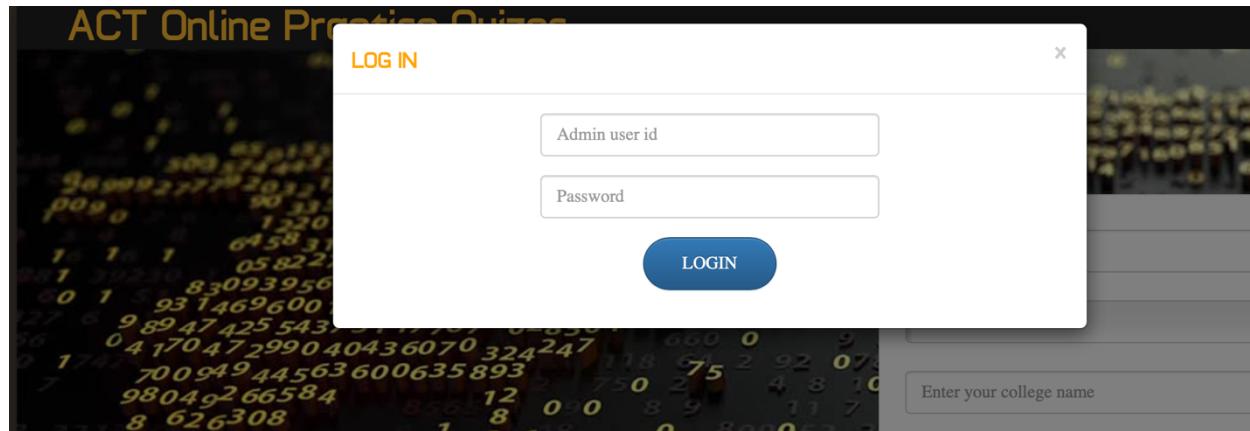
As you can see, after we log in and pass the form validation and the jQuery scripts, we enter on the account page where a user can:

- Take a new quiz (read & interact)
- Retake previous quizzes (They will become red and will be updated every time)
- Look at the history tab all the previous quizzes taken by the student
- Look at the Rankings (Leaderboard) where the student can compare with other students performance
- Or search for the quizzes by tags (the search appears after we click our name as a click on the top-right of the page and search for a tag using ajax)
- 5. The feedback form



**For this one to run I used the php.mailer**

## 6. Admin Dahboard



**The admin has enhanced features that also integrate extensively, complex algorithms, sorting, DOM traversing and CRUD tasks.**

**The Admin can:**

- Delete users

S.N.	Name	Gender	College	Email	Mobile	
1	Avantika	F	Stanford	avantika420@gmail.com	6985068889	
2	Mark Zuckerburg	M	Stanford	ceo@facebook.com	987654321	
3	Dan	M	act	dan@dan.com	7653	
4	Dddd	M	ddd	dd@dd.dd	4444444	
5	Komal	F	MIT	komalpd2011@gmail.com	4185068889	
6	Tom Cruze	M	Hollywood	mi5@hollywood.com	6985068889	
7	Netcamp	M	KNIT sultanpur	netcamp@gmail.com	987654321	

- Can read and delete the feedback request forms

S.N.	Subject	Email	Date	Time	By	
1	trial feedback	vikas@gmail.com	21-06-2019	07:02:31pm	vikas	
2	kcsncsk	sunnygkp10@gmail.com	20-06-2019	01:37:52pm	sunny	
3	testing	sunnygkp10@gmail.com	19-06-2019	09:22:15pm	testing	

trial feedback

- DATE: 21-06-2019 Time: 07:02:31pm By: vikas

triaal feedbak

- Can Create new quizzes  
Enter details, questions, etc.

## Enter Quiz Details

Enter Quiz title

Enter total number of questions

Enter marks on right answer

Enter minus marks on wrong answer without sign

Enter time limit for test in minute

Enter #tag which is used for searching

Write description here...

**Submit**

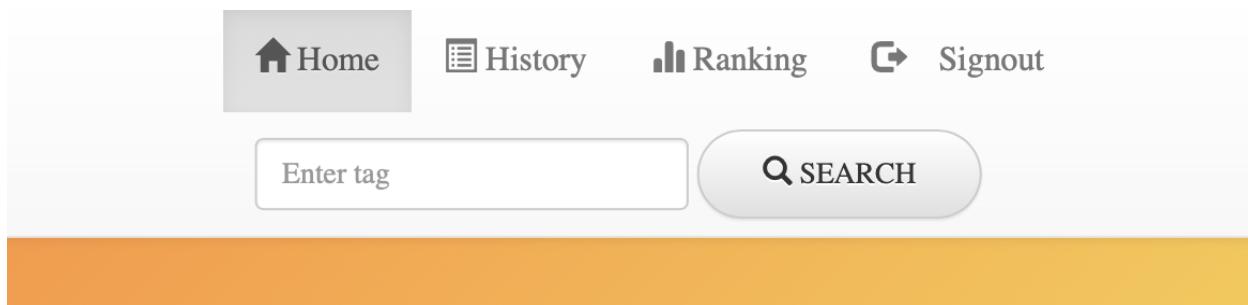
- **Can delete existing quizzes**

S.N.	Topic	Total question	Marks	Time limit	
1	Linux :vi Editor	5	10	10 min	<b>Remove</b>
2	Linux:startup	5	10	10 min	<b>Remove</b>
3	Networking	2	4	5 min	<b>Remove</b>
4	C++ Coding	2	4	5 min	<b>Remove</b>

- **And the update functionality happens when a student wants to retake a quiz**

S.N.	Topic	Total question	Marks	Time limit	
1	Linux :vi Editor ✓	5	10	10 min	<span style="color: red;">↻ REDO</span>
2	Linux:startup	5	10	10 min	<span style="color: green;">↻ GO</span>
3	Networking	2	4	5 min	<span style="color: green;">↻ GO</span>
4	C++ Coding	2	4	5 min	<span style="color: green;">↻ GO</span>

When someone retakes the quiz, they update the points that they receive from that sections and a resorting takes place both on their history and on the leader board, which triggers the advanced functions to take place (resorting and jQuery commands that turn red the buttons, green the section with a tick when that particular section was tried).



Using Ajax the search bar looks for quizzes that have the same tags attached to them.

#### - Taking the Quiz

Question 1 ::  
Which command is used to close the vi editor?

q  
 wq  
 both (a) and (b)  
 none of the mentioned

SUBMIT

[ACT Homepage](#)    [Admin Login](#)    [De](#)

After answering to the requested questions we get to see the answers and the points won.

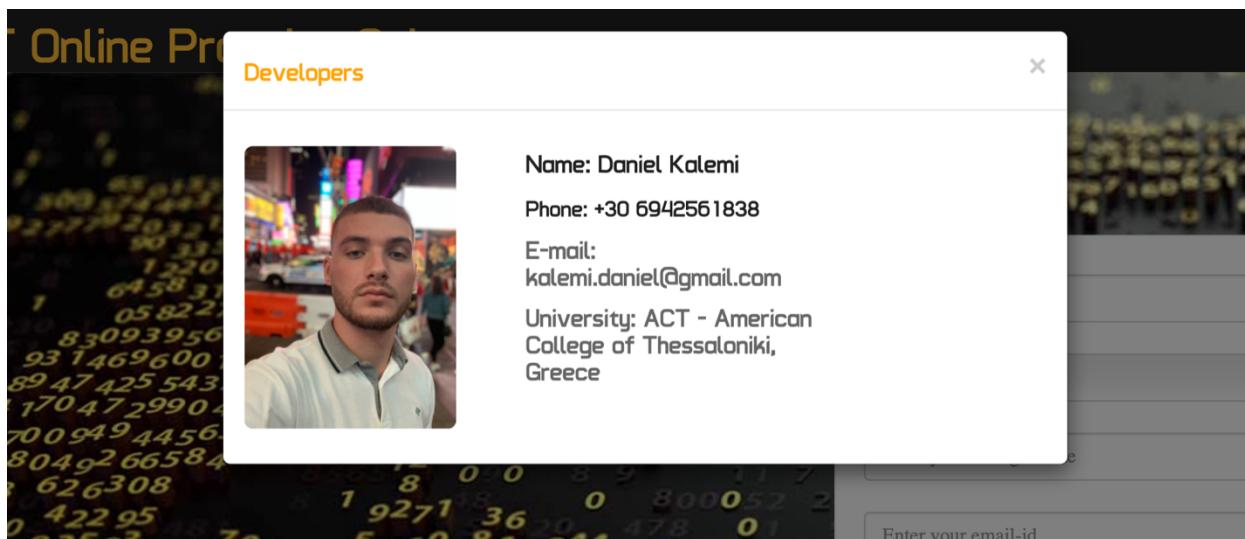
## RESULT

Total Questions	5
right Answer ☑	1
Wrong Answer ✗	4
Score ★	-2

All of them are color-coded and formatted with symbols with the help of:

jQuery  
 DOM traversing  
 Bootstrap  
 JavaScript scripts under the js folder

If anything arises, the developer is at your disposal



**NOTE:** I thought of adding some extra functionality, like an e-book section or a merch cart, but this web application has already 9 tables on the database and the timing given to complete the project was a big challenge to continue with the expansion of it. However this

**project slowly can become like a college website with multi-functionality and richer features.**

## **CODE DESIGN**

### **HTML/CSS/JS/JQUERY/FLEXBOX**

However, there are many more additional links in the form of buttons, clickable multimedia content etc.

There is a wide usage of many of the techniques that we learned in class in addition to many more advanced techniques that I had to search about while I was working on this project.

Some of the techniques are:

#### **HTML/JS/CSS/BOOTSTRAP**

Tables

Forms (with JavaScript & PHP functionality)

Grid

Gallery Slideshow (with JavaScript functionality)

Flexbox

JavaScript functions

Color Gradient using keyframes

Animations and keyframes

Color changing using animation and keyframes

Sticky navigation bar

Importing APIs, iframe and frameworks for fonts, animations and figures

Material Design

Building Parallax Webpages (where the background image is static and the front material of the page is moving creating and animated visual)

CSS Animation Design (typescript, color change, object animation, text manipulation etc.)

**Ajax Requests**

**DOM Traversing & jQuery**

```
CS306_project_Daniel_20170259 > js > JS main.js > ...
1  jQuery(document).ready(function($){
2      //open-close submenu on mobile
3      $('.cd-main-nav').on('click', function(event){
4          if($(event.target).is('.cd-main-nav')) $(this).children('ul').toggleClass('is-visible');
5      });
6  });
7
```

Selectors, Events, Effects, GET/SET (jQuery) – documentation on the js folder and especially implemented on the account.php, feedback.php, login/logout.php

## PHP & MySQL

### Object-Oriented Approach using classes and objects

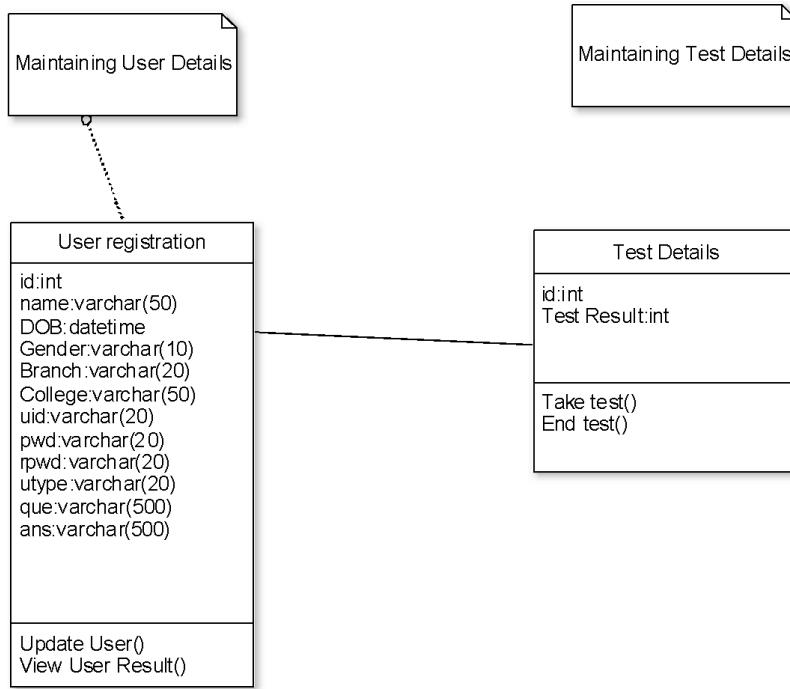
**9 Database tables (also with already inserted data, for the quizzes and some users for the leader board and the feedback notes)**

**C-create new quizzes**

**R- read from tables, history, results, jQuery alerts (feedback form etc)**

**U- update the quiz results by retaking the quiz**

**D – Delete users, feedbacks, existing quizzes**



## Objects & classes

### -Advanced Algorithm

**Under the js folder (the resorting functions) and under the update.php & account.php there is an extensive use of complex functions.**

**Also since there is a huge use of toggles and dynamic pages, made available by “modal contents”**

```
</div></div></div></div>
<!--Footer start-->
<div class="row footer">
<div class="col-md-3 box">
<a href="index.html">ACT Homepage</a>
</div>
<div class="col-md-3 box">
<a href="#" data-toggle="modal" data-target="#login">Admin Login</a></div>
<div class="col-md-3 box">
<a href="#" data-toggle="modal" data-target="#developers">Developers</a>
</div>
<div class="col-md-3 box">
<a href="feedback.php" target="_blank">Feedback</a></div></div>
<!-- Modal For Developers-->
<div class="modal fade title1" id="developers">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
```

### Usability, design, code structure (comments on the code)

- **Using functions.**

All the 11 php/html pages, along with the CSS files have passed the W3C Validation Test. I was testing my code live all the time since in Visual Studio Code there is a plugin from W3C to install and validate while working.

The content is 1100px. (within the 1200px suggested)

The navigation is visible and accessible without scrolling.

Each page links to all other pages.

I have used a “sticky” class as a div around the navigation bar and I used it for creating a sticky menu bar (fixed navigation). However, I have commented it out since it didn’t work in some of the pages due to the extra techniques that they had. Those pages that I used a parallax background the menu bar was disappearing since the background was moving. However, in all the other websites the feature worked properly.

The instructor is encouraged to use the feature by copying this line in CSS:

```
.sticky{position: fixed;}
```

The text is large enough to read and contrasts strongly with the background color.

Each page has at least one heading and at least one paragraph of text (the home page can skip the heading.)

The heading elements and paragraph elements are styled in CSS.

Spelling and grammar are immaculate.

At least one page includes a list coded with HTML list tags.  
At least one page contains a form.

## CODE DESIGN on PHP

On the final output of the code structure and design idea, **efficiency, simplicity & creativity** have been the main drivers along with the main goal to cover all of the material learnt so far.

### CHRONOLOGICAL ORDER OF CONCEPT COVERAGE ILLUSTRATED ON THE PROJECT.

- There is a wide coverage on the .php files on this project of concepts like:
- Escaping special characters (\r,\n,\t)
- Variables (\$amount)
- Datatypes & Common Arithmetic Operators
- String Concatenations (when saving the data to flat files)
- Comparison Operators
- Logical and Assignment operators
- Input using forms
- Controlling the flow of the program (as shown in the design walkthrough)
- Formatting Strings
- Build functions (for the validation, for calculating for the total amount to pay, for the multiple inputs to put them on an array a then pass them to a variable in order to be saved to the database tables.

```
function clean_text($string)
{
    $string = trim($string);
    $string = stripslashes($string);
    $string = htmlspecialchars($string);
    return $string;
}
```

- Working with arrays (Especially with the “select” and “option” inputs as they have multiple variations and can be easily converted to arrays in the PHP part

```

//WE COULD ALSO USE A CSV FILE TO PUT CONTENTS OF THE FIELD ALTERNATIVELY
/* $form_data = array(
   'sr_no'  => $no_rows,
   'name'   => $name,
   'email'  => $email,
   'subject' => $subject,
   'message' => $message
);
fputcsv($file_open, $form_data);*/

```

```

$c = array ( "01" => 1, "02" => 2, "03" => 3,"04" => 4, "05" => 5, "06" => 6,"07" => 7, "08" => 8, "09" => 9,"10"=>10;
$month = $_POST['month'];
foreach ($month as $m) {
| $month1 = $month1 + $c[$m];
}

$c1 = array ( "2020" => 2020, "2021" => 2021, "2022" => 2022, "2023" => 2023, "2024" => 2024, "2025" => 2025);
$year = $_POST['year'];
foreach ($year as $y) {
| $year1 = $year1 + $c1[$y];
}

```

-Use global variables

```

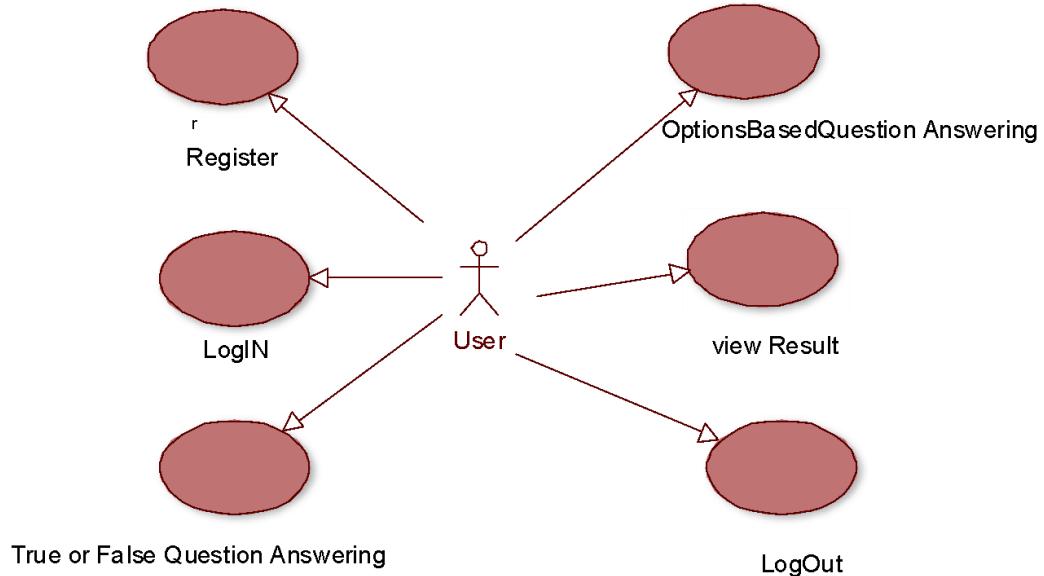
<?php
//customer.php

$error = '';
$name = '';
$email = '';
$subject = '';
$message = '';

```

## ***FINAL STAGE: Business Logic***

USECASE DIAGRAM: Students using the platform



SEQUENCE DIAGRAM: How the system flows and steps to follow

### Sequence Diagram

