

CSE233

Agile Software Engineering



Agile Project

Pirates Online Courses Website

| Name | ID |
|---------------------------------|---------|
| Seifeldin Sameh Mostafa Elkholy | 19p3954 |
| Yara Mostafa Ibrahim | 19p1120 |
| Mohamed Amr Fathy Naseef | 20p5552 |
| Omar Mohamed Shams Eldeen | 21p0188 |

Supervised by

Dr Salwa Ossama Selim

Eng. Rana sobhy

Contents

| | |
|--|----|
| Agile Software Engineering | 1 |
| Background | 4 |
| Client report | 5 |
| What the Website Will Accomplish | 5 |
| Prototype | 6 |
| Landing page | 6 |
| Courses in each track (2 prototypes) | 7 |
| Course content page | 9 |
| Log in and registration page | 10 |
| Sql Database | 11 |
| Database schema | 11 |
| Jira | 12 |
| Epics | 12 |
| User stories in epics | 13 |
| Example of items in user stories | 14 |
| Story mapping | 16 |
| Multilevel planning | 17 |
| Code | 21 |
| Php Folder | 21 |
| Database.php | 21 |
| login.php | 29 |
| Sessions.php | 29 |
| Signup.php | 30 |
| strings.php | 31 |
| utils.php | 32 |
| admin.php | 34 |

| | |
|------------------------------|----|
| Styles Folder | 35 |
| JS Folder..... | 35 |
| Admin Pages | 36 |
| User Pages | 36 |
| Login and logout Pages | 36 |
| Pages View..... | 37 |
| Index Page | 37 |
| Courses page..... | 38 |
| Courses Content Page | 39 |
| Login Page..... | 39 |
| Signup Page | 40 |
| Admin Pages | 40 |
| Jira link..... | 42 |
| GitHub link..... | 42 |

Background

Pirates is a non-profit student activity that was founded in 2009. Its main mission is to help bridge the gap between students and the market. Pirates' job is to give students the chance to flourish and prove their worth and find the diamond in the rough. Furthermore, it aims to prepare students for the future.



@Piratesegypt |     

In Egypt, Unemployment is a very critical problem we touch every day. The unemployment rate grew in 2021 reaching 9.33%, assuming we are 106,694,206¹ people living in Egypt then there are about 9,954,570 persons not working which is a huge deal.

At the same time, the ICT industry is the fastest growing sector in Egypt. Its growth rate is estimated to be about 16.3%² which allowed Egypt to create around 280k jobs with 30% women participation – we need to work on it to reach 50% -

What makes the software/ICT industry so special is that employees can fully operate remotely 16% of companies are working fully remotely now³. And you can easily export your software without much hassle other industries face. Which allowed Egypt's IT exports hit \$3.67 Billion Dollars!

Combining the unemployment data with the ICT growth with even the export rate, a new market gap is ready to be filled. Not only we can create a sufficient of software creating in Egypt to reduce the import crisis we are currently facing Egypt⁴ But we can capitalize on this gap to increase our shares of software exports allowing us to increase our reserve of dollars which will allow us to eliminate the dollar crisis.

We aim for a better future for all students. We aim to find the treasure buried inside each student to unlock a better future for everyone.

¹Egypt Population (2022) - <https://www.worldometers.info/world-population/egypt-population>

² FASTEST GROWING SECTOR - <https://itida.gov.eg/English/Programs/Industry-Outlook/Pages/default.aspx>

³ STATISTICS ON REMOTE WORKERS THAT WILL SURPRISE YOU (2022) - <https://www.apollotechnical.com/statistics-on-remote-workers/>

⁴ How Egypt survives its import crisis in 2022: struggle and resilience- <https://businessforwardafrica.com/2022/08/05/how-egypt-survives-its-import-crisis-in-2022-a-story-of-struggle-and-resilience/>

Client report

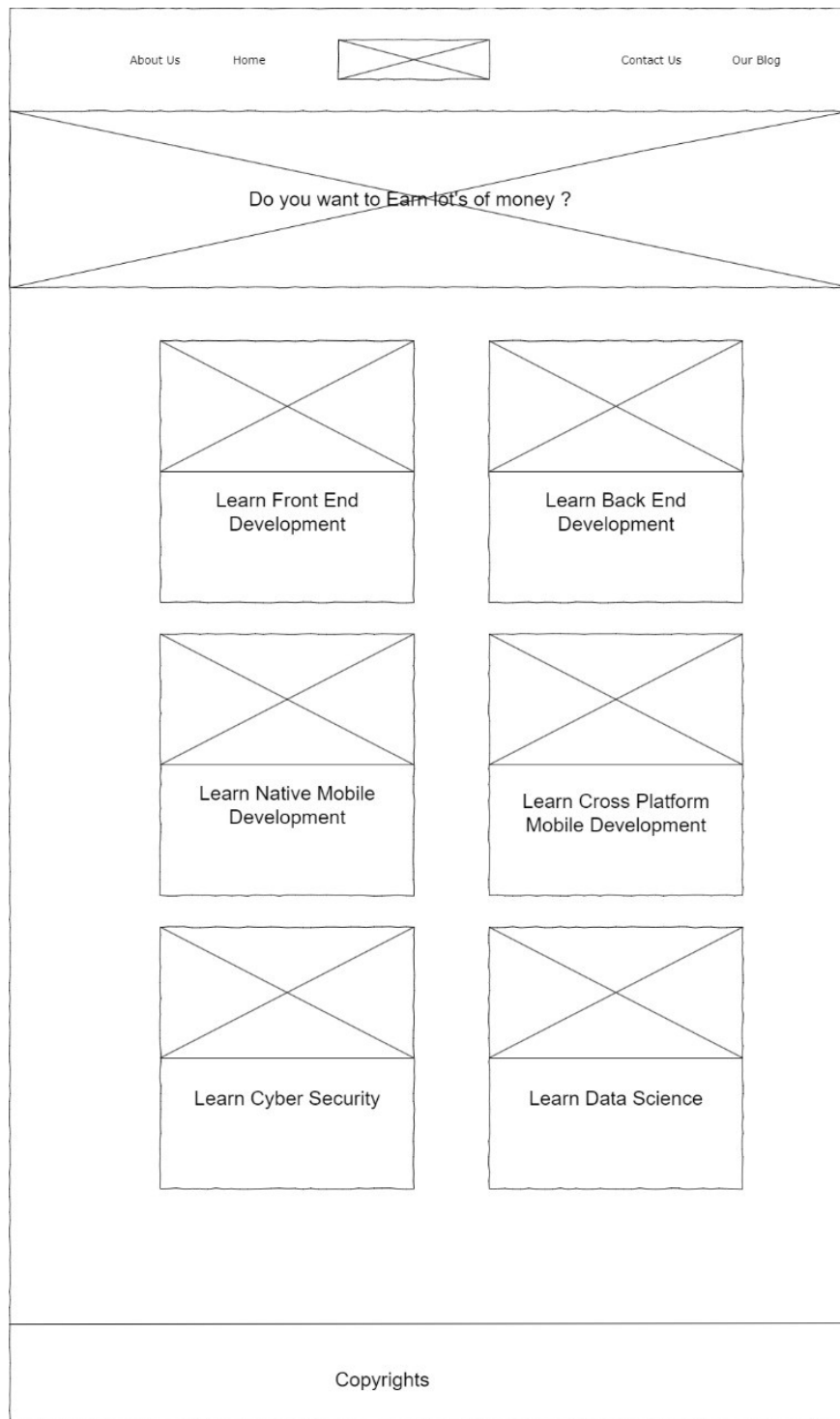
What the Website Will Accomplish

As many people don't know where to begin building their road map to a specific track, the goal of this application is to assist more people in learning and obtaining their dream jobs. Users can log in as students or administrators, and administrators can use their administrative powers to add courses or remove others. Administrators, on the other hand, may monitor their students' academic progress and determine whether they have what it takes to finish their courses.

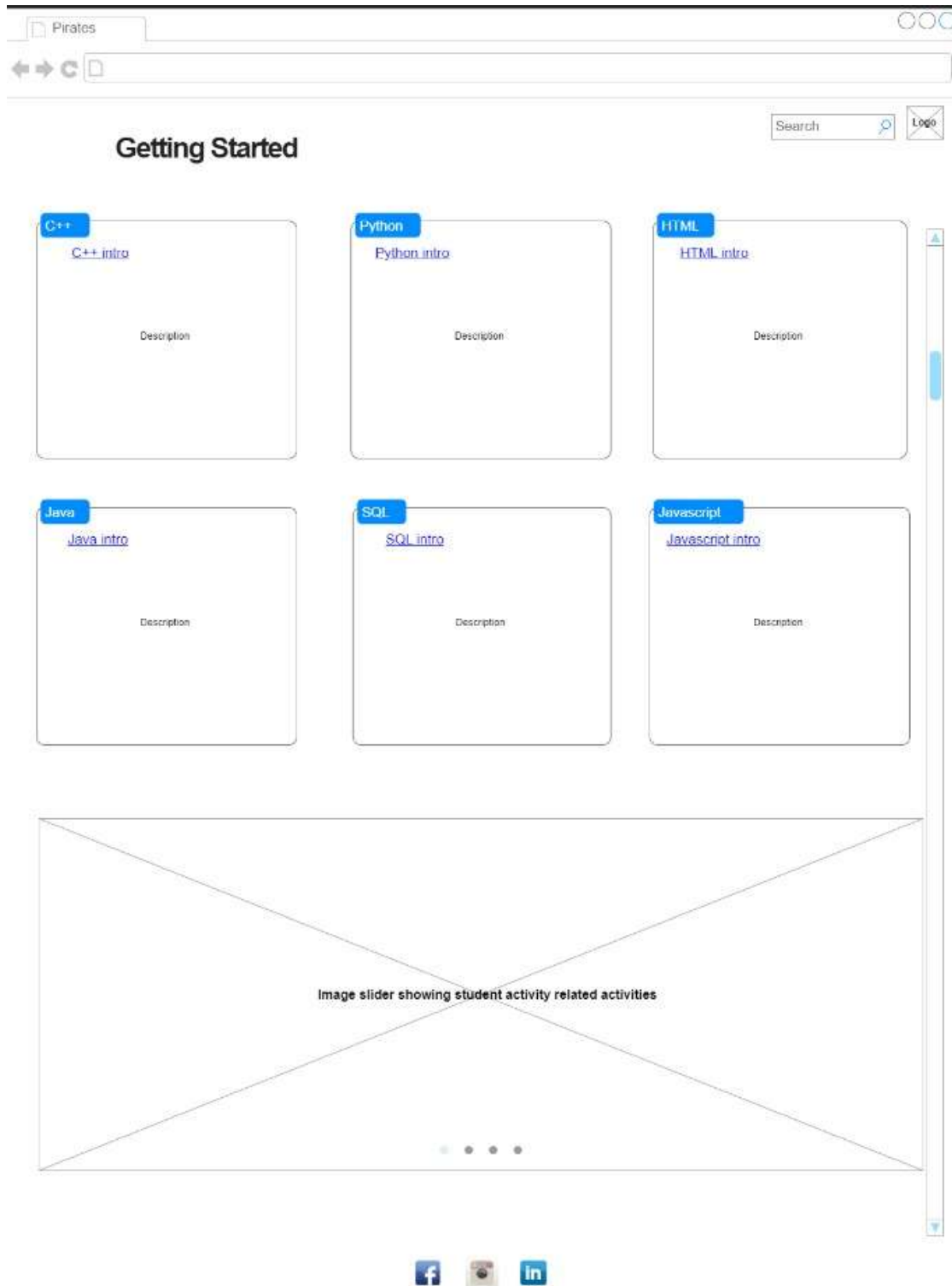
- **Login as admin:** The administrator is the one who adds or deletes courses from the system, as appropriate.
- **User login:** To create an account, students must sign up for the system on their own. Once they have successfully registered, they may log in to the system by providing their email and password.
- **Course Addition and Removal:** The administrator can add courses to the system by entering the course information or delete courses by selecting the desired course.
- **Access tracks and courses:** Users can access any available course from the track they choose.

Prototype

Landing page



Courses in each track (2 prototypes)

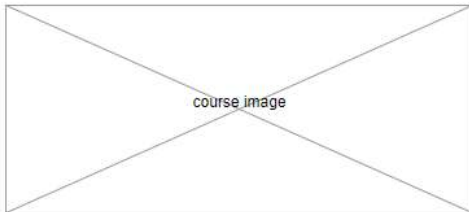
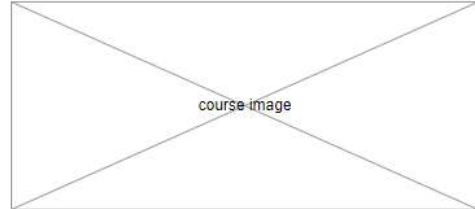


[Home](#)[Log In](#)

Course name

Description

Apply



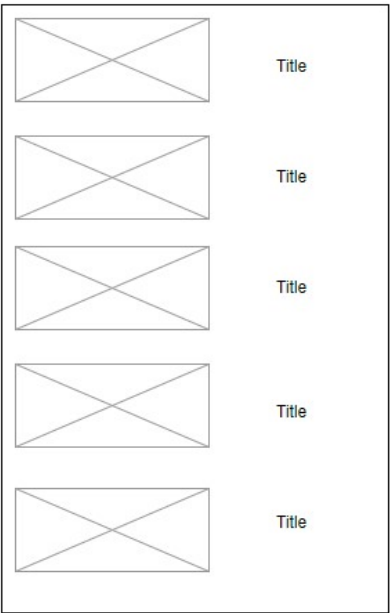
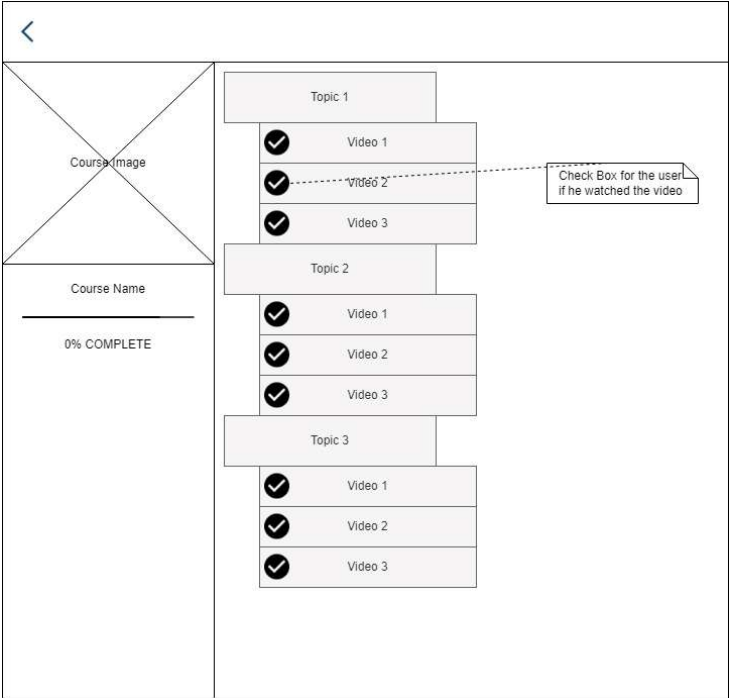
Course name

Description

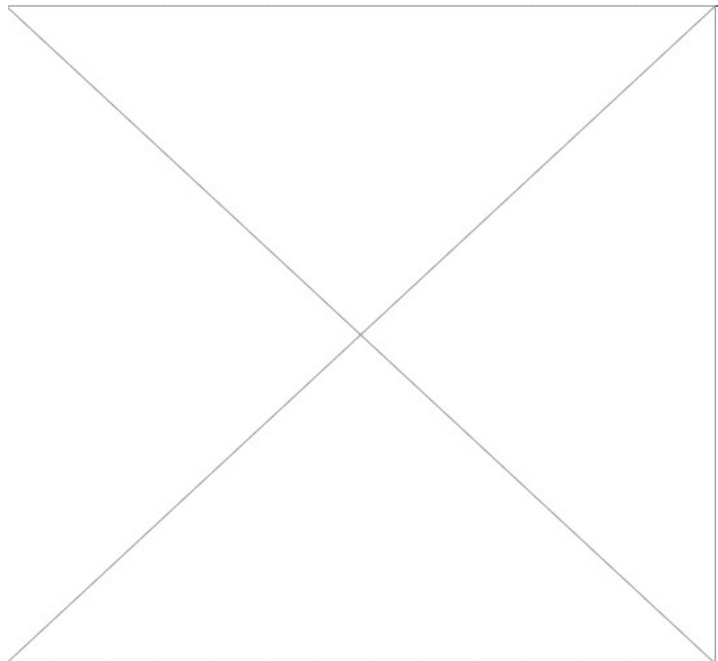
Apply



Course content page



Log in and registration page



Sign In

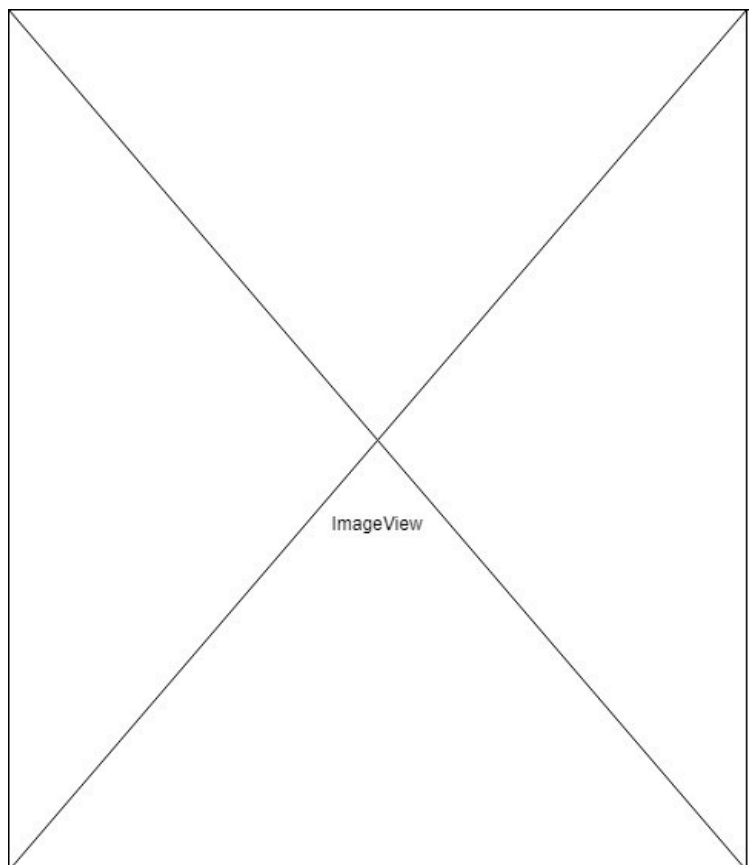
User Name:

Password:

SIGN IN

New User

SIGN UP

ImageView

Sign Up

Full name text box

Username text box

Email text box

Password text box

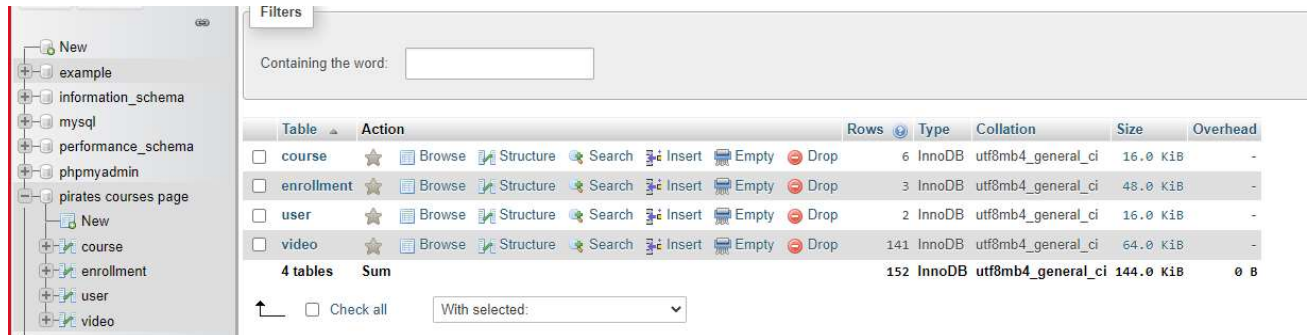
Re-enter Password text box

Address text box

Country text box

Submit button

Sql Database



Database schema

4 tables (user, enrollment, user, video)

Database: pirates courses page, Table: course, Purpose: Table structure

| Column | Type | Null | Default | Comments |
|---------------|--------------|------|---------|----------|
| cid | int(50) | No | | |
| cname | varchar(120) | No | | |
| cdescription | varchar(300) | No | | |
| imglink | varchar(300) | No | | |
| cstudentcount | int(50) | No | | |
| pid | int(50) | No | | |

Database: pirates courses page, Table: enrollment, Purpose: Table structure

| Column | Type | Null | Default | Comments |
|--------|---------|------|---------|----------|
| eid | int(50) | No | | |
| uid | int(50) | No | | |
| cid | int(50) | No | | |

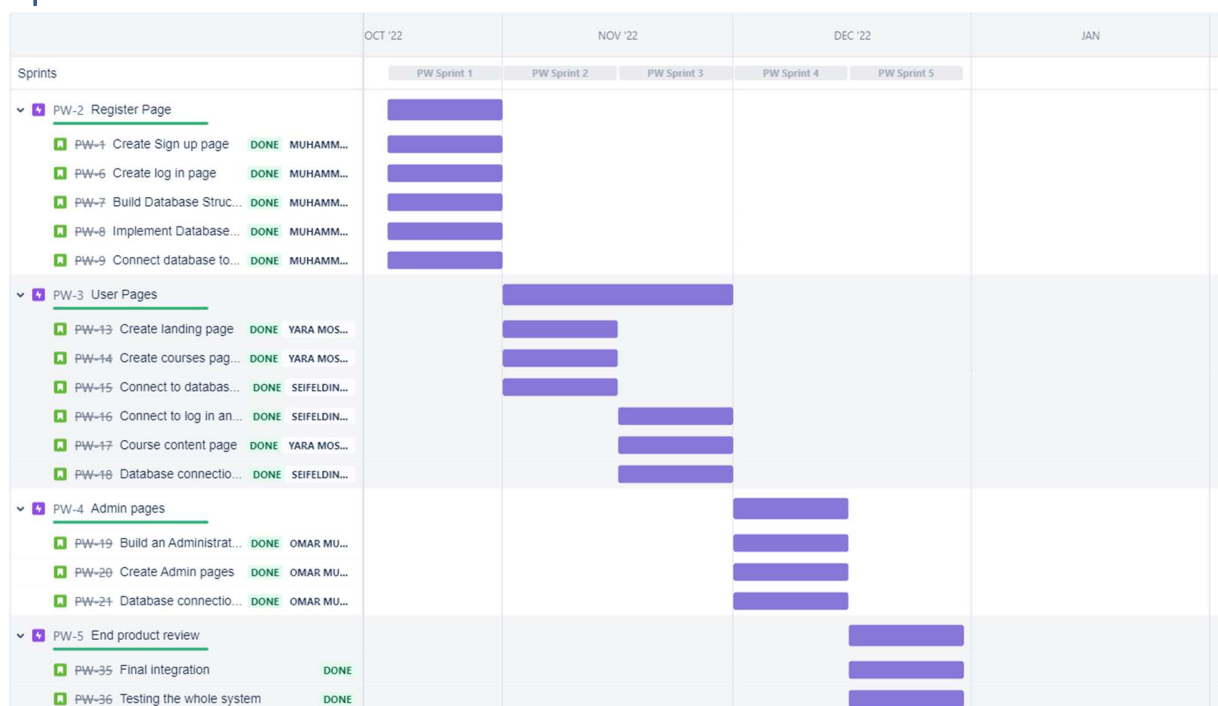
Database: pirates courses page, Table: user, Purpose: Table structure

| Column | Type | Null | Default | Comments |
|-----------|--------------|------|---------|----------|
| uid | int(50) | No | | |
| username | varchar(120) | No | | |
| upassword | varchar(120) | No | | |
| uemail | varchar(120) | No | | |
| uname | varchar(120) | No | | |
| uaddress | varchar(255) | No | | |
| ucountry | varchar(120) | No | | |
| urole | int(1) | No | | |

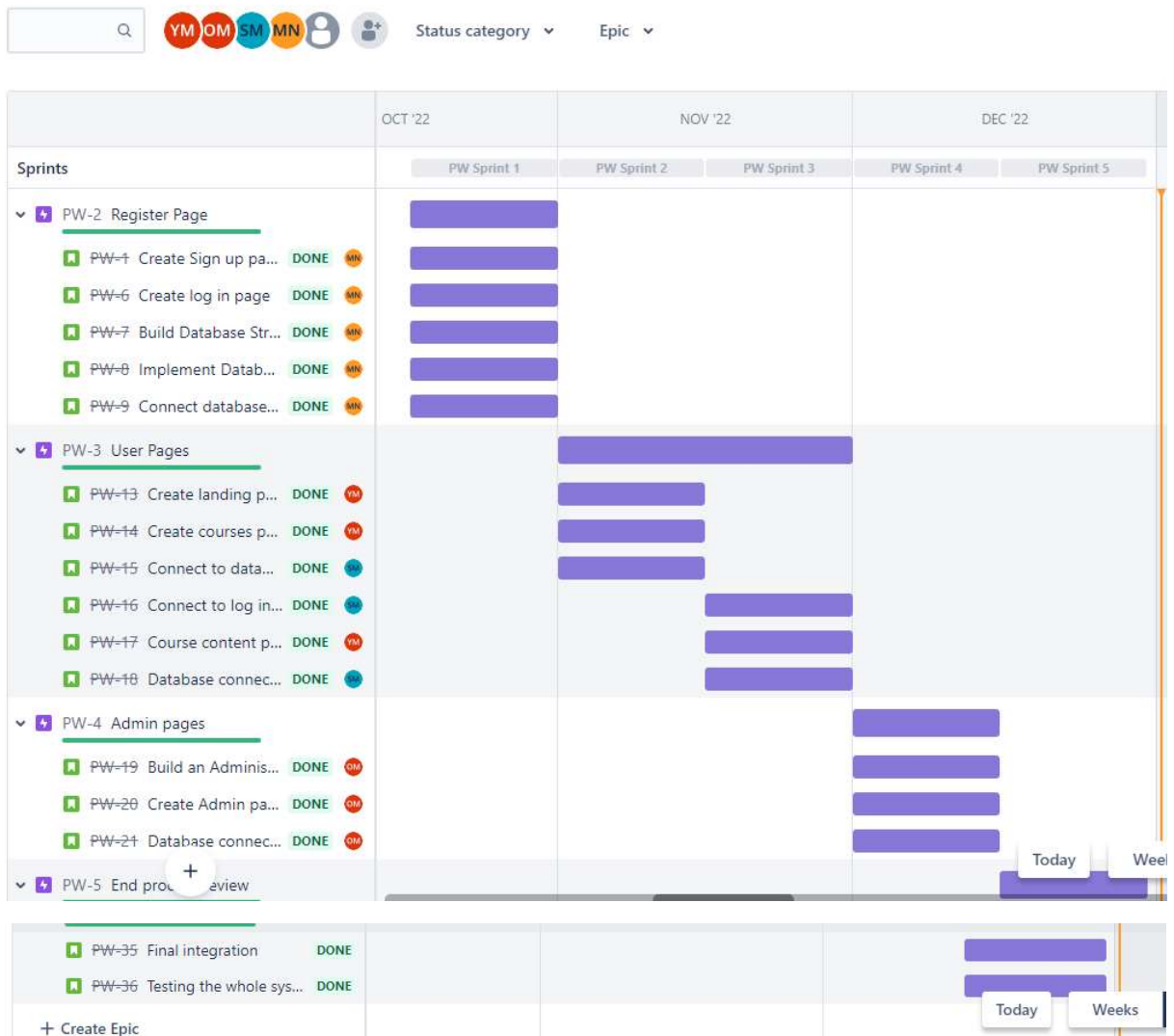
Database: pirates courses page, Table: video, Purpose: Table structure

| Column | Type | Null | Default | Comments |
|--------|--------------|------|---------|----------|
| vid | int(50) | No | | |
| vname | varchar(120) | No | | |
| vlink | varchar(255) | No | | |
| cid | int(50) | No | | |

Jira Epics



User stories in epics



Example of items in user stories

Sprints

PW-2 Register Page

PW-4 Create Sign up pa... DONE

PW-6 Create log in page DONE

PW-7 Build Database Str... DONE

PW-8 Implement Datab... DONE

PW-9 Connect database... DONE

PW-3 User Pages

PW-13 Create landing p... DONE

PW-14 Create courses p... DONE

PW-15 Connect to data... DONE

PW-16 Connect to log in... DONE

PW-17 Course content p... DONE

PW-18 Database connec... DONE

PW Sprint 1

PW Sprint 2

PW Sprint 3

PW Sprint 4

PW-7

Build Database Structure

Done

✓ Done

Description

- A database architecture is required to be created which will host the data of our website.

Child issues

Order by

100% Done

PW-40 Write draft for the data... 2 MN DONE

PW-41 Brainstorm Entities 2 MN DONE

PW-42 Brainstorm Relationships 2 MN DONE

PW-43 Create an EER Diagram 2 MN DONE

Sprints

PW-2 Register Page

PW-4 Create Sign up pa... DONE

PW-6 Create log in page DONE

PW-7 Build Database Str... DONE

PW-8 Implement Datab... DONE

PW-9 Connect database... DONE

PW-3 User Pages

PW-13 Create landing p... DONE

PW-14 Create courses p... DONE

PW-15 Connect to data... DONE

PW-16 Connect to log in... DONE

PW-17 Course content p... DONE

PW-18 Database connec... DONE

PW Sprint 1

PW Sprint 2

PW Sprint 3

PW Sprint 4

PW-16

As a user, I want to be able to navigate to the login page and login and see my name

Acceptance Criteria:

Open the Landing Page then navigate to the courses page then click on enroll button then login then navigate to the courses page then logout

Child issues

Order by

100% Done

PW-57 Implement navigation b... 3 SM DONE

PW-58 Get username from dat... 5 SM DONE

PW-59 Check if user is logged in 2 SM DONE

PW-60 Connect Landing page t... 3 SM DONE

PW-61 Connect Courses Page t... 2 SM DONE

Sprints

PW-2 Register Page

PW-4 Create Sign up pa... DONE

PW-6 Create log in page DONE

PW-7 Build Database Str... DONE

PW-8 Implement Datab... DONE

PW-9 Connect database... DONE

PW-3 User Pages

PW-13 Create landing p... DONE

PW-14 Create courses p... DONE

PW-15 Connect to data... DONE

PW-16 Connect to log in... DONE

PW-17 Course content p... DONE

PW-18 Database connec... DONE

PW Sprint 1

PW Sprint 2

PW Sprint 3

PW Sprint 4

Clicking on 'apply now' button in any track offered from the Landing page open courses offered for this track, then when choosing a course the course content page opens and display videos

Attachments (1)

image-20221230...110.png

30 Dec 2022, 10:51 am

Child issues

Order by

100% Done

PW-31 Create videos playlist vi... 8 YM DONE

PW-34 Each video shall have a ... 5 YM DONE

PW-56 connect page to course... 3 YM DONE

Sprints

PW-2 Register Page

PW-1 PW-1 Create Sign up pa... DONE

PW-6 Create log in page DONE

PW-7 Build Database Str... DONE

PW-8 Implement Datab... DONE

PW-9 Connect database... DONE

PW-3 User Pages

PW-13 Create landing p... DONE

PW-14 Create courses p... DONE

PW-15 Connect to data... DONE

PW-16 Connect to log in... DONE

PW-17 Course content p... DONE

PW-18 Database connec... DONE

PW-4 Admin pages

PW-19 Build an Adminis... DONE

PW-20 Create Admin pa... DONE

PW-21 Database connec... DONE

PW Sprint 1

PW Sprint 2

PW Sprint 3

PW Sprint 4

Create Admin page

Done Done

Description

Add a description...

Child issues

Order by 100% Done

PW-22 Add courses 3 OM DONE

PW-23 Delete course 3 OM DONE

PW-24 Update existing courses 5 OM DONE

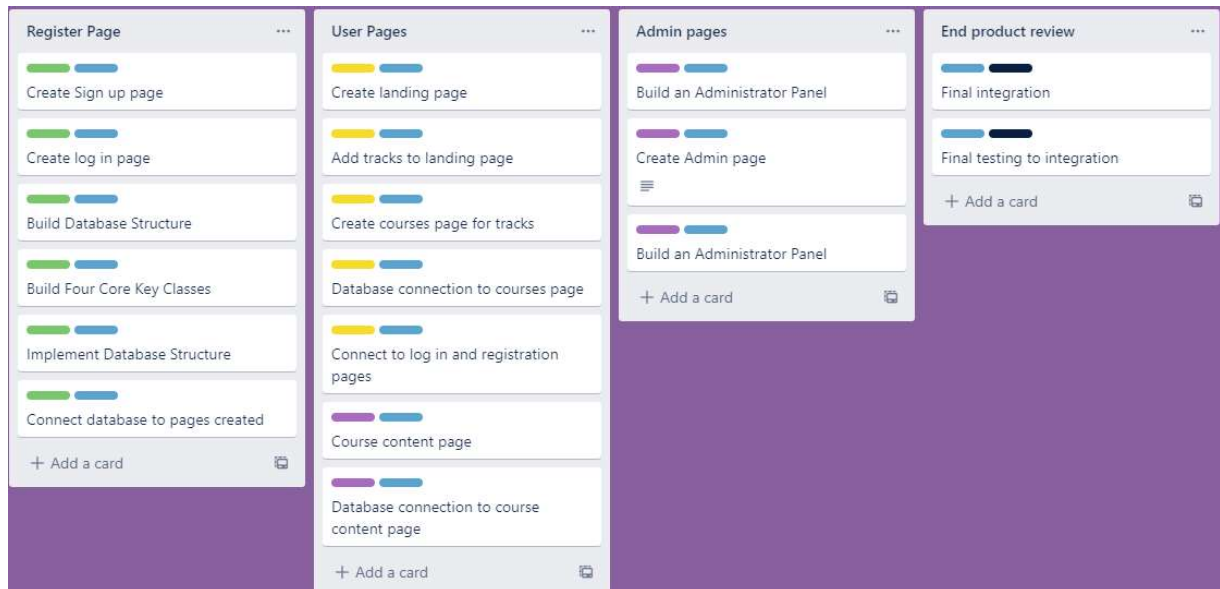
PW-25 Delete courses video(s) 3 OM DONE

Pinned fields

Click on the next to a field label to start pinning.

YM Add a comment

Story mapping



labels

Release 1 -> green

Release 2 -> yellow

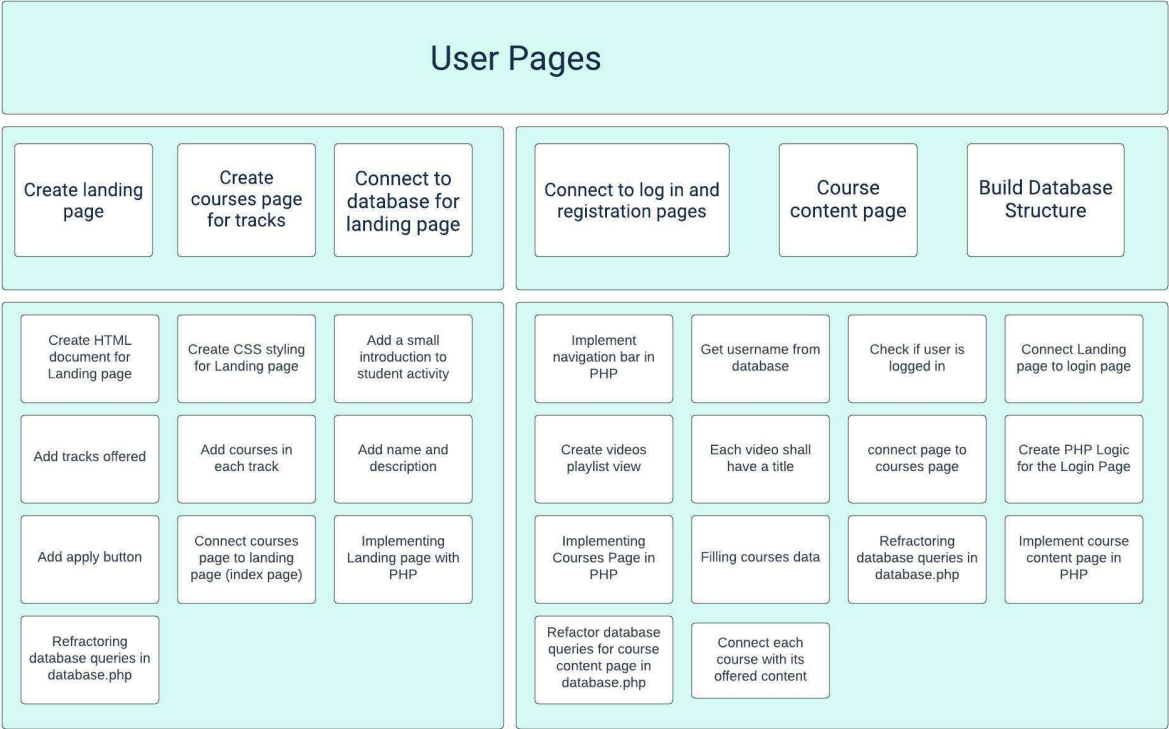
Release 3 -> purple

Release 4 -> black

Done -> blue

Multilevel planning





End product review

Final integration

Deploy the system

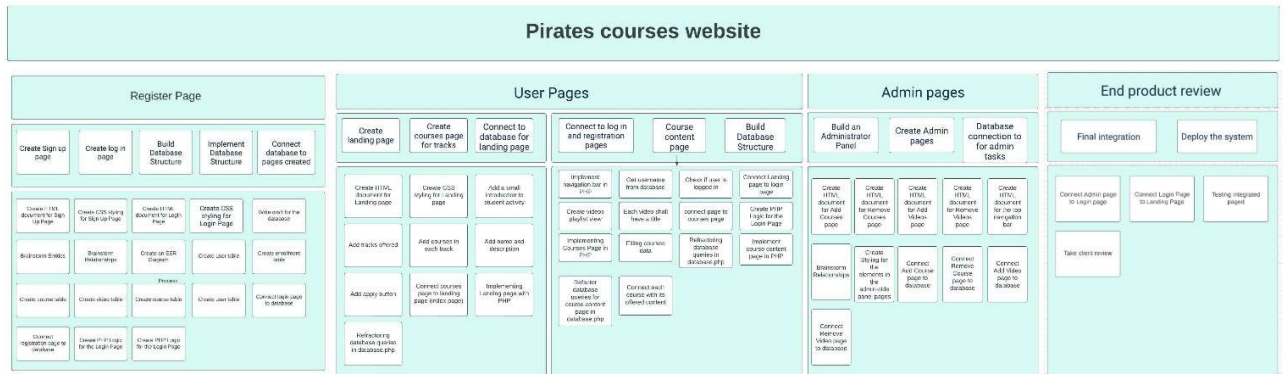
Connect Admin page
to Login page

Connect Login Page
to Landing Page

Testing integrated
paged

Take client review

Whole view of theme then epics then user stories then items.



Code

Php Folder

Database.php

```
database.php M X
php > database.php
1  <?php
2  require "utils.php";
3
4  > class DB { ...
285 }
286 ?>
287
```

Database.php serves as a singleton and acts as the controller that contacts the database either to get data or modify/change data in the database.

```
class DB {
    private $host="localhost";
    private $username="root";
    private $password="";
    private $database="learntoearn";
    public $connection;

    function __construct() {
        $this->connection = mysqli_connect($this->host,$this->username,$this->password,$this->database);
        if(!$this->connection){
            die("Connection Failed");
        }
    }
}
```

The constructor is responsible for the connection to the database. It contains name of host(localhost), username(root), password, our database name(learntoearn).

If the connection fails, Connection Failed is printed out.

```
function signup($submit,$username,$password1,$password2,$email,$address,$country,$name){
    if(isset($submit) && empty_check($username)&&empty_check($password1)&&empty_check($email)&&empty_check($address)
        &&empty_check($country)&&empty_check($name)){
        if($this->isPasswordAvailable($password1,$password2) == 1 &&$this->isEmailAvailable($email) == 1&& $this->
            $query="INSERT INTO user (username, upassword, uemail, uname, uaddress, ucountry, urole) VALUES('$username','$password1','$password2','$email','$address','$country','$name')";
            $result = mysqli_query($this->getConnection(),$query);
            return 1;
        }else{
            return "Your data is duplicated";
        }
    }else{
        return "Please enter complete data";
    }
}
```

The signup function creates a new user in the database. It contains error handling:
 1- Checking if the username entered was entered before, 2- Checking if all data fields were entered.

It also has a layer of security in that it hashes password entered to md5 in the database.

```
function login($submit,$username,$password){
    if(isset($submit) && empty_check($username)&&empty_check($password)==1){
        $query="SELECT * FROM user WHERE username='".$username.'" AND upassword='".md5($password)."'";
        $res = mysqli_query($this->connection, $query);
        if(mysqli_num_rows($res)==1){
            //there exist a user with these credentials
            return [1, $res->fetch_assoc()['urole']];
        }else{
            //there is no user with these credentials
            return "please enter valid credentials";
        }
    }else{
        //missing data
        return "Please enter complete data";
    }
}
```

The login function checks if entered username and password are in the database and are correct.

```
function getConnection(){
    return $this->connection;
}
```

getConnection returns our database connection

```
function isUsernameAvailable($username){
    $query="SELECT * FROM user WHERE username='".$username."'";
    $result = mysqli_query($this->getConnection(),$query);
    if(mysqli_num_rows($result)){
        return 0;
    }else{
        return 1;
    }
}
```

This function checks if username is already used or not and is used in login function.

```
function getUserCount($cid){
    $query= "SELECT COUNT(*) FROM enrollment WHERE cid='".$cid."'";
    $result = mysqli_query($this->getConnection(),$query);
    $data = mysqli_fetch_assoc($result);
    return $data["COUNT(*)"];
}
```

getUserCount takes course id as parameter and returns number of students enrolled in this course

```
function setUserCount($cid,$count){
    $query = "UPDATE course SET cstudentcount='$count' WHERE cid = '$cid'";
    $result = mysqli_query($this->getConnection(),$query);
}
```

setUserCount takes course id and student count and updates it in our database.

```
function isPasswordAvailable($password1,$password2){
    if($password1===$password2){
        return 1;
    }else{
        return 0;
    }
}
```

This function checks if used have not been repeated before in our database


```
function isEmailAvailable($email){
    $query="SELECT * FROM user WHERE uemail='".$email."'";
    $result = mysqli_query($this->getConnection(),$query);
    if(mysqli_num_rows($result)){
        return 0;
    }else{
        return 1;
    }
}
```

isEmailAvailable checks if given email have not been used before to create another account

```
function addCourse($c_name, $c_desc, $c_link, $c_imag, $c_studs, $c_track) {
    $sql_insert_course =
    "INSERT INTO `course` (cname, cdescription, clink, imglink, cstudentcount, pid)
    VALUES('$c_name', '$c_desc', '$c_link', '$c_imag', '$c_studs', '$c_track')";
    if (mysqli_query($this->getConnection(), $sql_insert_course)) {
        return true;
    } else {
        return false;
    }
}
```

addCourse adds course with its information to database. This function is used in admin pages.

```
function addVideo($v_name, $v_link, $c_id) {
    $sql_insert_vid =
    "INSERT INTO `video` (vname, vlink, cid)
    VALUES('$v_name', '$v_link', $c_id)";
    if (mysqli_query($this->getConnection(), $sql_insert_vid)) {
        return true;
    } else {
        return false;
    }
}
```

addVideo adds video with its information to database.


```

function removeCourse($courseID) {
    $this->removeVideo(NULL, $courseID);
    if ($courseID) {
        $sql_remove = "DELETE FROM `course` WHERE `course`.`cid` = $courseID";
        if (mysqli_query($this->getConnection(), $sql_remove)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}
}

```

removeCourse removes course with given courseID from database

```

function removeVideo($videoID, $courseID) {
    if ($videoID) {
        $sql_remove = "DELETE FROM `video` WHERE `video`.`vid` = $videoID";
        if (mysqli_query($this->connection, $sql_remove)) return true;
    } elseif ($courseID) {
        $sql_remove = "DELETE FROM `video` WHERE `video`.`cid` = $courseID";
        if (mysqli_query($this->connection, $sql_remove));
        return true;
    } else {
        return false;
    }
}
}

```

removeVideo deletes course with given videoID and courseID from database

```

function getList($vid1course0) {
    if ($vid1course0 == 1) {
        $sql = "SELECT vid, vname FROM video";
        $col1 = 'vid';
        $col2 = 'vname';
    } else {
        $sql = "SELECT cid, cname FROM course";
        $col1 = "cid";
        $col2 = "cname";
    }
    $r = mysqli_query(
        $this->getConnection(),
        $sql
    );

    $result_col1 = [];
    $result_col2 = [];
    while ($array = mysqli_fetch_array($r)) {
        $result_col1[] = $array[$col1];
        $result_col2[] = $array[$col2];
    }
    $result = [[]];

    for ($i = 0; $i < count($result_col1); $i++) {
        array_push($result,
            [$result_col1[$i], $result_col2[$i]]);
    }
    return array_slice($result, 1);
}

```

getList returns array that contains course name and id and video name and id

```

function getTracks() {
    return
    [
        [1, "Front End"],
        [2, "Back End"],
        [3, "Data Science"],
        [4, "Mobile Developement"]
    ];
}

```

getTrack returns a list with all tracks offered

```

function showCourses(){
    $pid = $_GET["pid"];

    if(isset($_SESSION["username"])){
        $uname = $_SESSION["username"];
        $user = mysqli_query($this -> getConnection(), "SELECT uid FROM user WHERE username='$uname'");
        $urow = mysqli_fetch_array($user,MYSQLI_ASSOC);
        $uid = $urow["uid"];
        $enrolled_course = mysqli_query($this -> getConnection(), "SELECT eid,enrollment.cid FROM course,enrollment,user WHERE user.uid = $uid AND enrollment.user_id = $uid");

        $result = mysqli_query($this -> getConnection(), "SELECT cid,cname,cdescription,imglink FROM course WHERE pid = $pid");
        if(isset($_SESSION["username"])){
            $query = array("result" => $result,
                "pid" => $pid,
                "uname" => $uname,
                "user" => $user,
                "urow" => $urow,
                "uid" => $uid,
                "enrolled_course" => $enrolled_course
            );
        }else{
            $query = array("result" => $result,
                "pid" => $pid,
            );
        }
        return $query;
    }
}

```

showCourses retrieves all courses offered for a specific track, then returns an array containing user info and enrolled courses for this certain user if logged in; else, returns array with courses of selected track.

```

function showCourseName(){
    $cid = $_GET["cid"];
    if(isset($_SESSION["username"])==False){
        header('Location:login.php');
    }
    $uname = $_SESSION["username"];

    $result = mysqli_query($this -> getConnection(), "SELECT cname FROM course WHERE cid = $cid");
    $user = mysqli_query($this -> getConnection(), "SELECT uid,uemail FROM user WHERE username='$uname'");
    $enrolled_course = mysqli_query($this -> getConnection(), "SELECT cid FROM enrollment,user WHERE user.uid = $uid AND enrollment.user_id = $uid");
    $videos = mysqli_query($this -> getConnection(), "SELECT vname,vlink FROM video WHERE cid = $cid");

    $row1 = mysqli_fetch_array($enrolled_course,MYSQLI_ASSOC);
    $cname = mysqli_fetch_array($result,MYSQLI_ASSOC);
    $urow = mysqli_fetch_array($user,MYSQLI_ASSOC);
    if(is_null($row1)){
        $uid = $urow["uid"];
        $usercount= $this->getUserCount($cid);
        $this->enrollUser($uid,$cid);
        $usercount++;
        $this->setUserCount($cid,$usercount,$urow["uemail"]);
    }
    return $cname;
}

```

showCourseName returns coursename

```
function enrollUser($uid,$cid){
    $sql = "INSERT INTO enrollment (eid, uid, cid)
VALUES (NULL, $uid, $cid)";
    mysqli_query($this -> getConnection(), $sql);
}
```

enrollUser enrolls user with uid into course with cid into enrollment table into database

```
function showVideo(){
    $cid = $_GET["cid"];
    $uname = $_SESSION["username"];
    if(isset($_SESSION["username"])==False){
        header('Location:login.php');
    }
    $result = mysqli_query($this -> getConnection(), "SELECT cname FROM course WHERE cid = $cid");
    $user = mysqli_query($this -> getConnection(), "SELECT uid FROM user WHERE uusername='$uname'");

    $enrolled_course = mysqli_query($this -> getConnection(), "SELECT cid FROM enrollment,user WHERE user.uid = enro
    $videos = mysqli_query($this -> getConnection(), "SELECT vname,vlink FROM video WHERE cid = $cid");

    $row1 = mysqli_fetch_array($enrolled_course,MYSQLI_ASSOC);
    $cname = mysqli_fetch_array($result,MYSQLI_ASSOC);
    $urow = mysqli_fetch_array($user,MYSQLI_ASSOC);
    if(is_null($row1)){
        $uid = $urow["uid"];
        $sql = "INSERT INTO enrollment (eid, uid, cid)
VALUES (NULL, $uid, $cid)";
        mysqli_query($this -> getConnection(), $sql);
    }
    return $videos;
}
```

showVideo retrieves videos from video database for certain course with cid

login.php

```
login.php M X
php > login.php
1  <?php
2  require "sessions.php";
3  require "database.php";
4  require "strings.php";
5  $username= $_POST["username"];
6  $password= $_POST["password"];
7
8  $db = new DB();
9
10 $login = $db->login($_POST["submit"],$username,$password);
11
12 if ($login[0] == 1) {
13     $_SESSION["username"] = $username;
14     $_SESSION["userrole"] = $login[1];
15     if ($login[1] == 1) {
16         redirect("/index.php");
17     } elseif ($login[1] == 0) {
18         redirect("/admin/addcourse.php");
19     } else {
20         echo $login[0];
21     }
22 }
23
24 ?>
```

Checks return value. If 1, user is redirected to index page; if 0, user is an admin and is redirected to admin pages

Sessions.php

```
sessions.php X
php > sessions.php
1  <?php
2  session_start();
3  ?>
```

Starts session after user is logged in

Signup.php

```
signup.php M X
php > signup.php
1  <?php
2  require "sessions.php";
3  require "database.php";
4  require "strings.php";
5
6  $username= $_POST["username"];
7  $password1= $_POST["password1"];
8  $password2= $_POST["password2"];
9  $email= $_POST["email"];
10 $address= $_POST["address"];
11 $country= $_POST["country"];
12 $name= $_POST["name"];
13
14 $db = new DB();
15
16 $signup = $db->signup($_POST["submit"],$username,$password1,$password2,$email,$address,$country,$name);
17
18 if ($signup==1){
19     redirect("/login.php");
20 }else{
21     echo $signup;
22 }
23
24 ?>
```

Sends user info to signup to register in the database and assigns user role 1.

strings.php

Contains all strings used in our View.

All strings were refactored to this file

```
strings.php M X
php > strings.php
1  <?php
2  // General
3  $hi = "Hi";
4  $logout = "Log Out";
5  $login = "Log In";
6  $signup = "Sign Up";
7  $showlearn = "Learn";
8  $enrollnow = "Enroll Now";
9  $addcourse = "Add Course";
10 $addtocourse = "Add To Course";
11 $removecourse = "Remove Course";
12 $addvideo = "Add Video";
13 $removevideo = "Remove Video";
14 $selectvideoremove = "Select a video to remove";
15 $selectcourseremove = "Select a course to remove";
16 $videoname = "Video Name";
17 $videolink = "Video Link";
18 $coursename = "Course Name";
19 $courseauthor = "Course Author";
20 $coursedescription = "Course Description";
21 $imagelink = "Image Link";
22 $coursetrack = "Course Track";
23 // Titles
24 $titlesuffix = " | Pirates - Innovate to Lead";
25 $logintitle = "Log In".$titlesuffix;
26 $signuptitle = "Sign Up".$titlesuffix;
27 $coursespages = "Courses".$titlesuffix;
28 $removevideotitle = "Remove Video".$titlesuffix;
29 $removecoursetitle = "Remove Course".$titlesuffix;
30 $addvideotitle = "Add Video".$titlesuffix;
31 $addcoursetitle = "Add Course".$titlesuffix;
32 // Input Fields
```

```

32 // Input Fields
33 $usernameph = "Your Username ...";
34 $passwordph = "Your Password ...";
35 $emailph = "Your Email ...";
36 $password2ph = "Re-enter Your Password ...";
37 $addressph = "Your Address ...";
38 $countryph = "Your Country ...";
39 $fullph = "Your Full Name ...";
40 ?>

```

utils.php

```

utils.php M X
php > utils.php
1 <?php
2 function redirect($url) {
3     header('Location: '.$url);
4     die();
5 }
6
7 function isloggedin($urole){
8     //urole = 0 for admin, 1 for regular user
9     if (!isset($_SESSION["username"]) || !($_SESSION["userrole"]==$urole)) {
10         redirect("/login.php");
11     }
12 }
13
14 function empty_check($field){
15     if(!empty($field)){
16         return 1;
17     }else{
18         return 0;
19     }
20 }
21

```

Contains utility function used in view.

isloggedin checks user role and opens a session for the appropriate user role.


```

function ack_removed($removed, $vid1course0) {
    if ($removed) {
        if($vid1course0==1) {
            echo '<script>alert("Video removed successfully")</script>';
        } else {
            echo '<script>alert("Course removed successfully")</script>';
        }
        header("Refresh:0");
    } else {
        if($vid1course0==1) {
            echo '<script>alert("No videos found")</script>';
        } else {
            echo '<script>alert("No courses found")</script>';
        }
    }
    header("Refresh:0");
}

```

ack_removed returns acknowledgment message for item removed

```

function ack_added($vid1course0) {
    if($vid1course0==1) {
        echo '<script>alert("Video Added successfully")</script>';
    } else {
        echo '<script>alert("Course Added successfully")</script>';
    }
    header("Refresh:0");
}

```

ack_added returns acknowledgment message for item added

admin.php

admin.php U X

php > admin.php

```
1  <?php
2  require "database.php";
3  $db = new DB();
4
5  if (isset($_POST['add_course'])) {
6
7      $c_name = $db->escapeString($_POST['course_name']);
8      // $c_auth = $db->escapeString($_POST['course_author']); // Not currently used
9      $c_desc = $db->escapeString($_POST['course_description']);
10     // $c_link = $db->escapeString($_POST['course_link']);
11     $c_imag = $db->escapeString($_POST['image_link']);
12     $c_track = $db->escapeString($_POST['course_track']);
13
14     $cadd = $db-> addCourse($c_name, $c_desc, $c_link, $c_imag, 0, $c_track);
15     if ($cadd)
16         ack_added(0);
17 }
18
19 if (isset($_POST['add_video'])) {
20     $v_name = $db->escapeString($_POST['video_name']);
21     $v_link = $db->escapeString($_POST['video_link']);
22     $c_id = $db->escapeString($_POST['course_with_vid']);
23
24     $vadd = $db->addVideo($v_name, $v_link, $c_id);
25     if ($vadd)
26         ack_added(1);
27 }
28
29 if (isset($_POST['remove_course'])) {
30     $course_id = $db->escapeString($_POST['courses']);
31     $course_rem = $db->removeCourse($course_id);
32     ack_removed($course_rem, 0);
33 }
```

```

29  if (isset($_POST['remove_course'])) {
30      $course_id = $db->escapeString($_POST['courses']);
31      $course_rem = $db->removeCourse($course_id);
32      ack_removed($course_rem, 0);
33  }
34
35  if (isset($_POST['remove_video'])) {
36      $video_id = $db->escapeString($_POST['videos']);
37      $vid_rem = $db->removeVideo($video_id, NULL);
38      ack_removed($vid_rem, 1);
39  }
40
41  ?>

```

Contains all admin page functionalities (add course, remove course, add video, remove video).

Styles Folder

| ▼ styles | |
|----------------------|---|
| # admin.css | U |
| # courses.css | U |
| # CoursesContent.css | U |
| # Landing-page.css | U |
| # navbar.css | U |
| # nicepage.css | U |
| # users.css | U |
| # watch.css | U |






Contains all CSS style sheets for all pages

JS Folder



| ▼ js | |
|----------------|---|
| JS jquery.js | U |
| JS nicepage.js | U |


Contains all JavaScript files used

Admin Pages

| | |
|--|---|
|  addcourse.php | U |
|  addvideo.php | U |
|  removecourse.php | U |
|  removevideo.php | U |
|  topnav.php | U |

User Pages


| | |
|---|---|
|  courses.php | U |
|  index.php | M |

| | |
|---|---|
|  watch.php | U |
|---|---|

The courses page shows all courses offered in a specific track dynamically.



The index is our main/landing page.


Watch.php is our video page that shows all videos offered in a certain course

| | |
|--|---|
|  navbar.php | U |
|--|---|

Navigation bar used for our pages

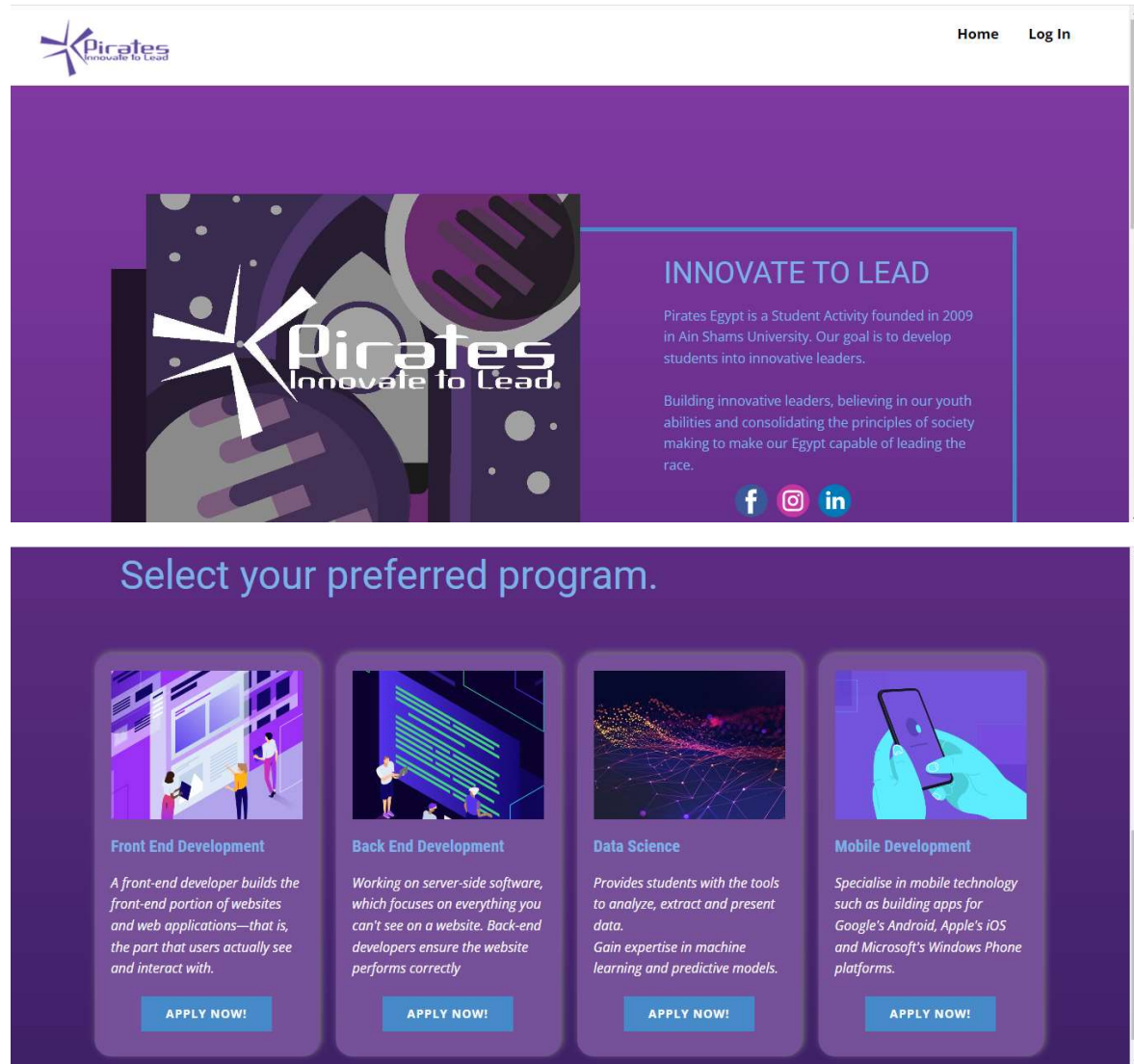
Login and logout Pages

| | |
|--|---|
|  login.php | M |
|  logout.php | U |

| | |
|--|---|
|  signup.php | M |
|--|---|

Pages View


Index Page



Courses page



[Home](#) [Log In](#)




CSS

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. This tutorial will teach you CSS from basic to advanced.

[Enroll Now](#)

HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting.



HTML

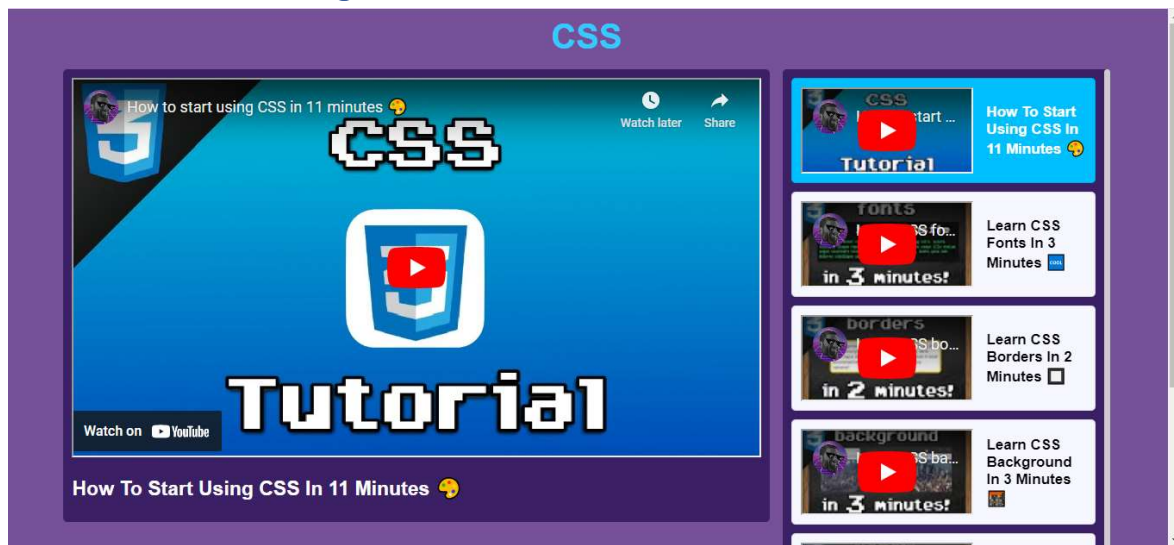
The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

[Enroll Now](#)

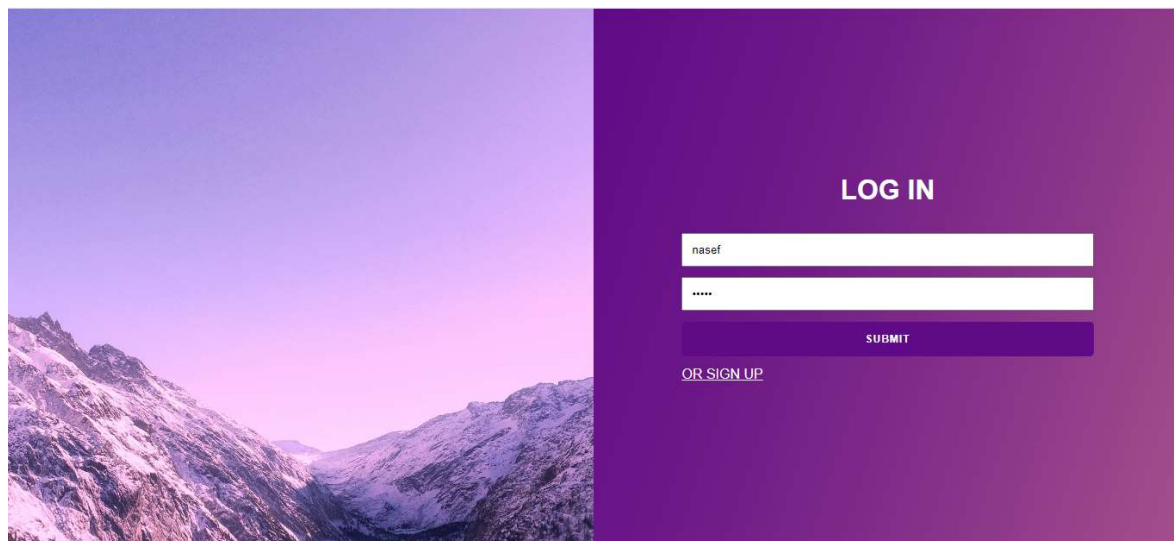


Address: Faculty of Engineering, Ain Shams University

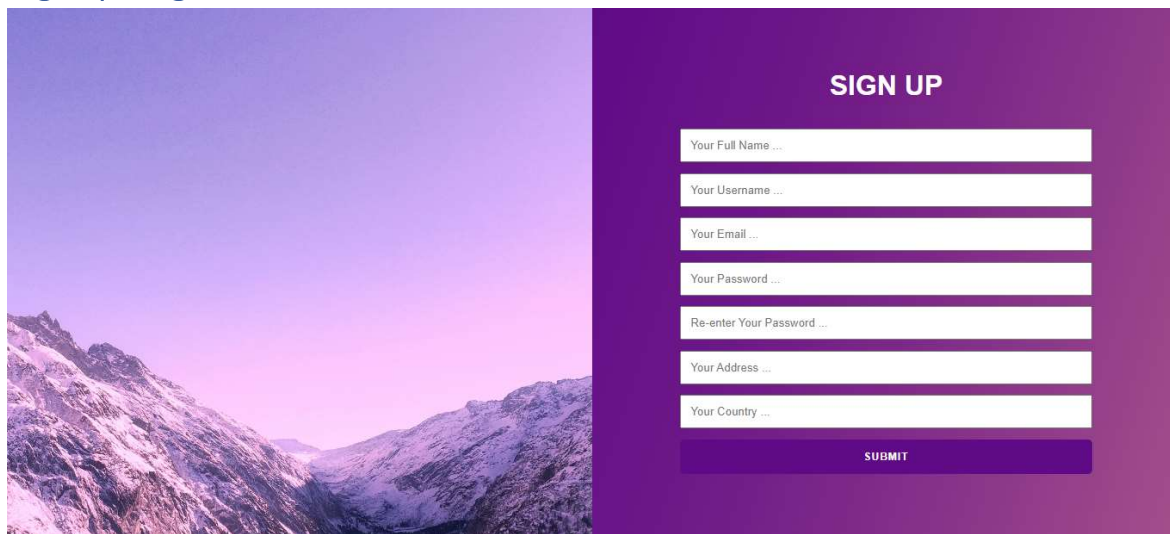
Courses Content Page



Login Page



Signup Page

The image shows a web page for signing up. On the left is a landscape photograph of a mountain valley under a blue sky. On the right is a dark blue vertical panel with the text "SIGN UP" in white. Below the text are seven white input fields with dark blue borders, each with a label: "Your Full Name ...", "Your Username ...", "Your Email ...", "Your Password ...", "Re-enter Your Password ...", "Your Address ...", and "Your Country ...". At the bottom of the panel is a dark blue button with the word "SUBMIT" in white.

SIGN UP

Your Full Name ...

Your Username ...

Your Email ...

Your Password ...

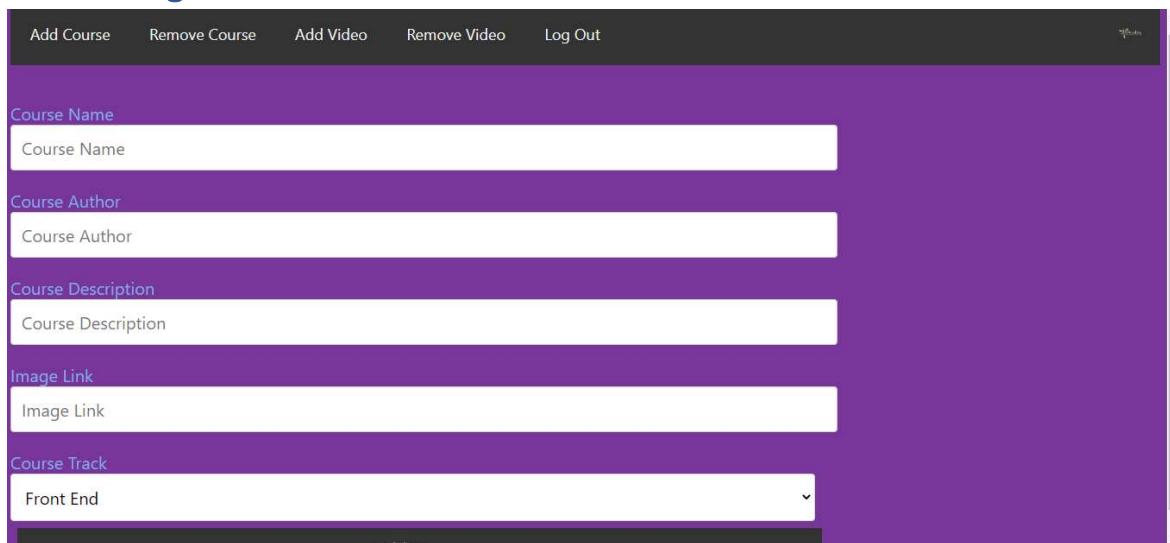
Re-enter Your Password ...

Your Address ...

Your Country ...

SUBMIT

Admin Pages

The image shows an admin interface. At the top is a dark blue header bar with five white links: "Add Course", "Remove Course", "Add Video", "Remove Video", and "Log Out". Below the header is a light blue background. On the left side, there are five labels in blue: "Course Name", "Course Author", "Course Description", "Image Link", and "Course Track". Each label is followed by a white input field with a dark blue border. The "Course Name" field contains the text "Course Name". The "Course Author" field contains the text "Course Author". The "Course Description" field contains the text "Course Description". The "Image Link" field contains the text "Image Link". The "Course Track" field is a dropdown menu with "Front End" selected and a small downward arrow on the right. At the bottom of the page, there is a dark blue bar with the text "Add Course" in white.

Add Course Remove Course Add Video Remove Video Log Out

Course Name
Course Name

Course Author
Course Author

Course Description
Course Description

Image Link
Image Link

Course Track
Front End

Add Course

Add CourseRemove CourseAdd VideoRemove VideoLog Out

Select a course to remove

Java

Remove Course

Add CourseRemove CourseAdd VideoRemove VideoLog Out

Video Name

Video Name

Video Link

Video Link

Add To Course

Java

Add Video

Add CourseRemove CourseAdd VideoRemove VideoLog Out

Select a video to remove

Learn HTML In Arabic 2021 - #01 - Introduction and What I Need To Learn

Remove Video

Jira link

<https://pirates-website.atlassian.net/jira/software/projects/PW/boards/1/roadmap?selectedIssue=PW-35&shared=&atlOrigin=eyJpIjoiYWJiMzY2MjZhZTRhNDE1YmEzYWZmZDBkY2U4NGVjODgiLCJwIjoiajI9>

GitHub link

Includes php code, css code, and js code

<https://github.com/Pirates-Website/Pirates-website>