

PROJECT REPORT FOR PRT 452 SOFTWARE ENGINEERING: PROCESS AND TOOLS ON MNEMONICS DATABASE

NAME	ROLE	Contact details		
Dr. Simon Moss	Product Owner and Client	Simon.Moss@cdu.edu.au		
Dr. Peter Shaw Subject faculty		peter.shaw@cdu.edu.au		
Rahul Rao	Project Team Member	rahulrao.v@gmail.com		
Rahul Shokeen Project Team Member		rahulshokeen94@gmail.com		
Lin Ma	Project Team Member	malin0703lin@gmail.com		
Aman Kathed	Project Team Member	amankanthed1994@gmail.com		

Contents

Project Introduction	3
Requirements	3
Functional requirements	3
Non-functional requirements	5
Security Requirements	5
UI Design	6
Database design	10
E-R Diagram	11
UML Design	12
Class Diagram	12
Activity Diagram	13
Sequence diagram	14
Use Case Diagram	15
Search Mnemonic	15
Signup	16
Profile Use Case Diagram	17
Add Mnemonics Use Case Diagram	18
Login Use Case Diagram	19
Design Pattern	20
Agile process	21
Pair programming	23
User story Cards	23
Testing	24
CODE	27
References	97

Project Introduction

The objective of the project was to develop a database to help people insert and search for relevant mnemonics. A mnemonic, or memory gadget, is any learning strategy that guides data maintenance in the human memory. Mnemonics make utilization of elaborative encoding, recovery prompts, and symbolism as particular devices to encode any given data in a way that takes into consideration proficient capacity and recovery. Mnemonics help unique data in to be related with something which, enables the mind to have better organization of the data. For example, it can be as simple as a "Knuckle Mnemonic", it is learned at an early age to remember the number of days in a month for every knuckle representing a month with 31 days. Our project here deals with mnemonics that are specified into various categories by registered users.

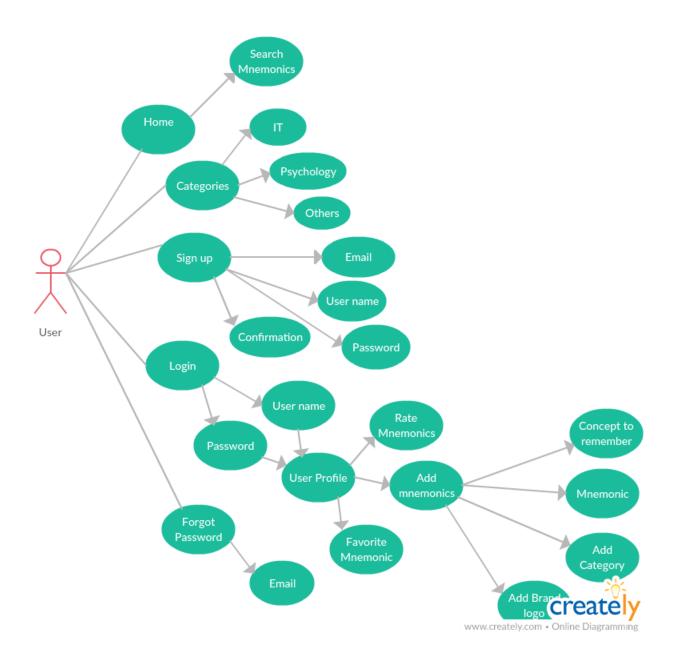
A video that briefs about the project is as follows: https://youtu.be/5AT5QgcyweM

Requirements

Functional requirements

The web application shall

- Take input as mnemonics from registered users.
- Display mnemonics i.e. Take search queries from all who wish to learn
- Allow registered users to rate the mnemonics on their like.
- Allow registered users to favorite a mnemonic they liked.
- Allow registered users to view their profile, which shows in a nutshell all the mnemonics added, favorites etc.
- When people enter a mnemonic, the web application should also prompt them to specify their organization and, if possible, enter the logo of their organization. So, people might then enter mnemonics to promote their organization.
- The web application should provide users with the categories. They shall initially receive
 a drop-down menu with about 10 options: Psychology, IT, law, biology, chemistry, or
 other. If they press "other", they should be prompted to write a category.



PRT452 Project report Mnemonics Database

Non-functional requirements

The web application should

- Be easy to use and the search process be accurate.
- Be available always.
- Not accept any entry of a mnemonic with a swear word in it.
- Be secure with the user details it holds.
- Be able to maintain the huge dataset and flow of mnemonics.
- Support all browsers and should function with ease on all platforms.
- Monitor the quality of mnemonics, it should have automated swear control mechanism.
- The web application's response time for search queries should be less.

Security Requirements

The web application shall

- Accept registration once a sign up is initiated.
- Verify the email id for verification of genuine users.
- Allow users which have successfully verified their email id to log in the portal.
- Allow registered users to change their passwords.
- Registered users to use the forgot password feature if they forget their password.

UI Design

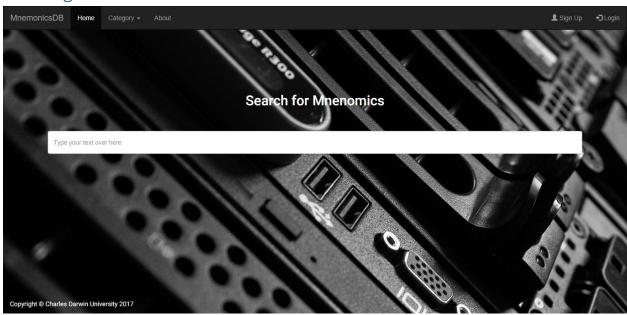


Figure- Home page

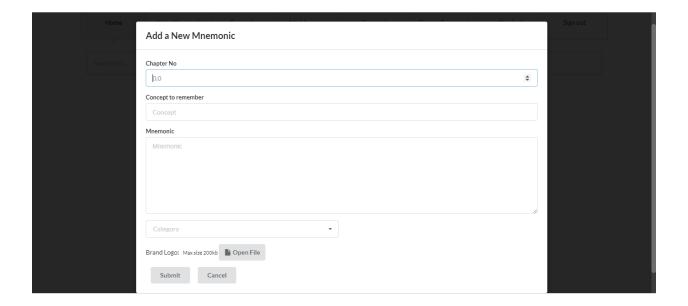


Figure- Add mnemonics

PRT452 Project report Mnemonics Database

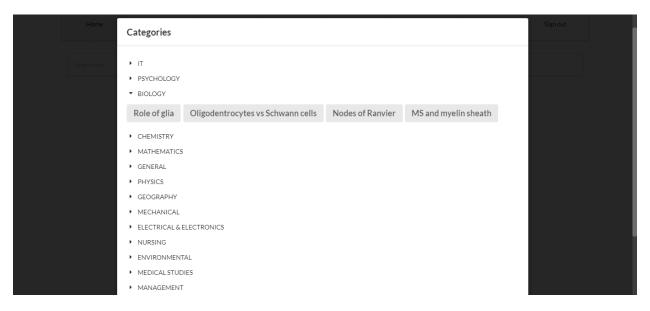


Figure- Categories

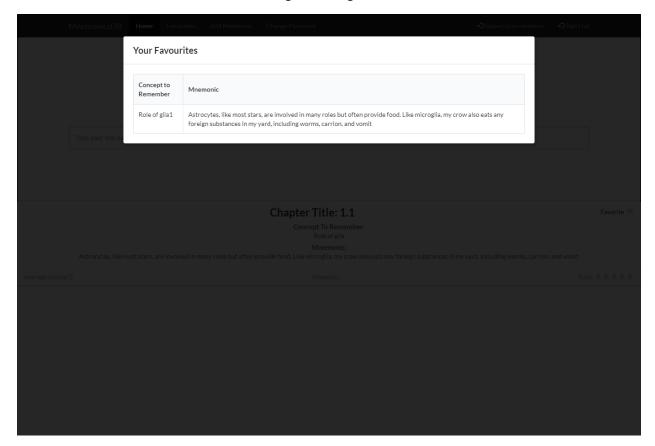


Figure- Favorites

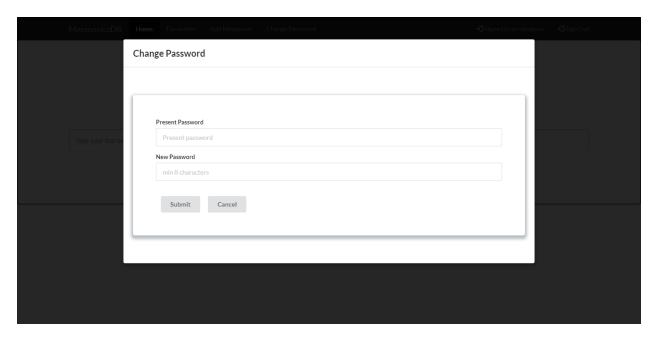


Figure- Change Password

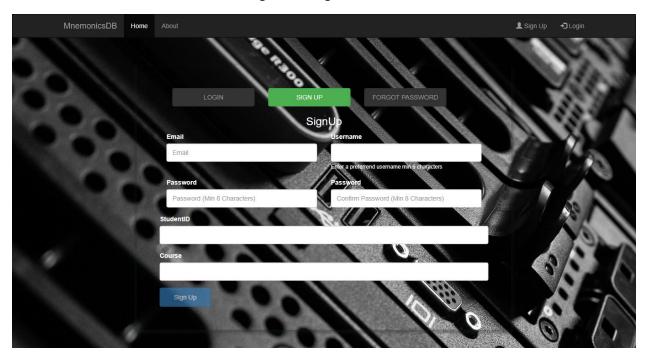


Figure- Signup page

PRT452 Project report Mnemonics Database

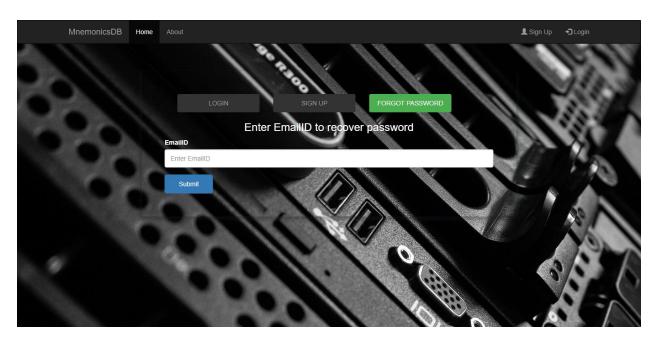


Figure- Forgot password

Database design

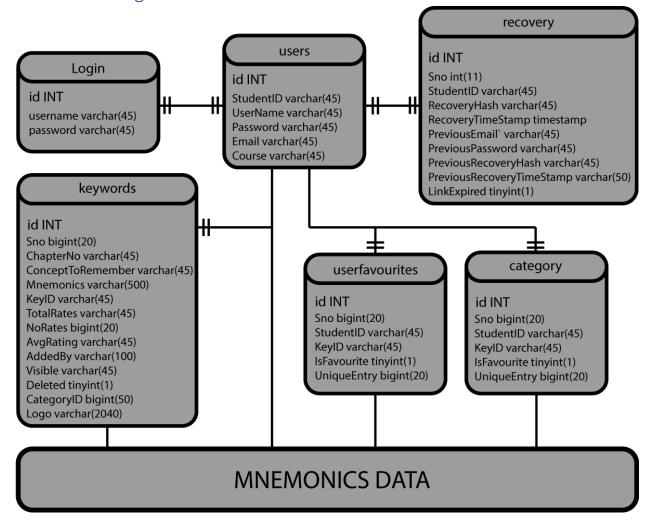


Figure- Database Design

E-R Diagram **EMAIL PASSWORD** USERNAME COURSE STUDENT ID SIGN UP **USER** SEARCH MATCHES CONCEPT **KEYWORD MNEMONICS LOGIN** ID PASSWORD USERNAME **FAVORITE** ADD **RATE**

Figure- 2 E-r Diagram

UML Design

Class Diagram

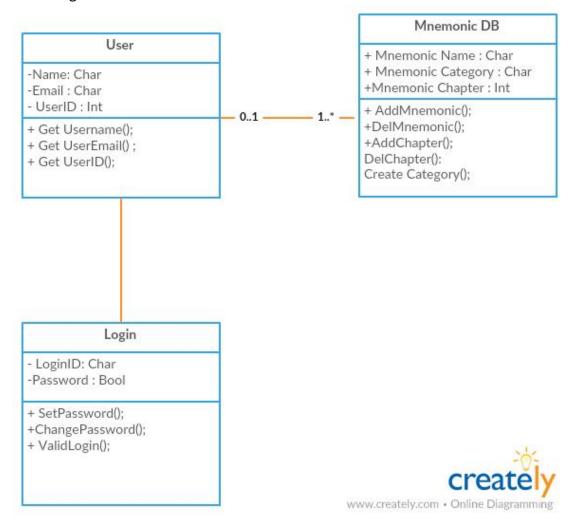


Figure-1 Class Diagram

Activity Diagram

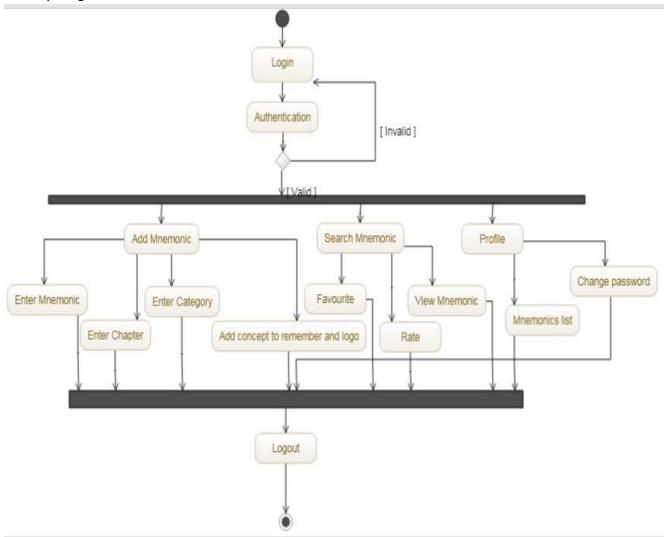


Figure-2 Activity Diagram

Sequence diagram

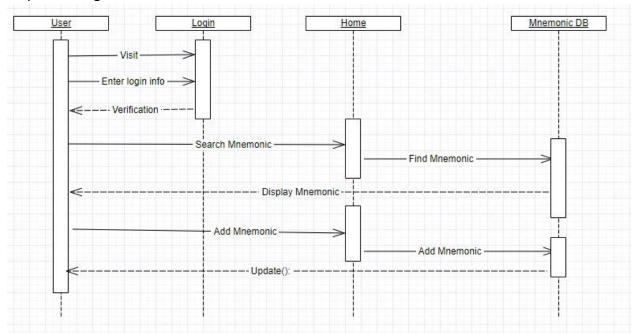


Figure-3 Sequence Diagram

Use Case Diagram
Search Mnemonic

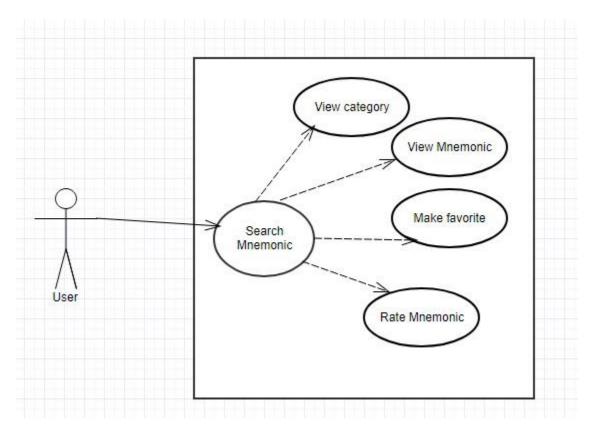


Figure-4 Use Case Diagram for Search Mnemonic

Signup Use Case

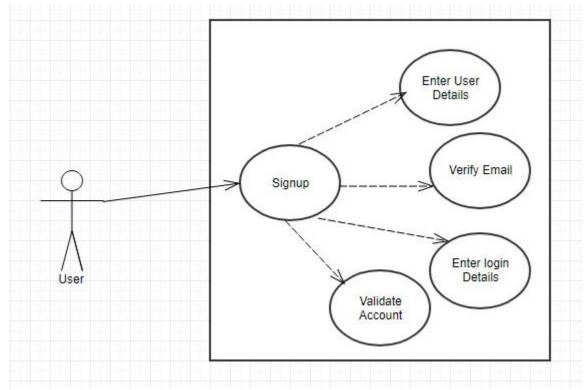


Figure-5 Sign up use case Diagram

Profile Use Case Diagram

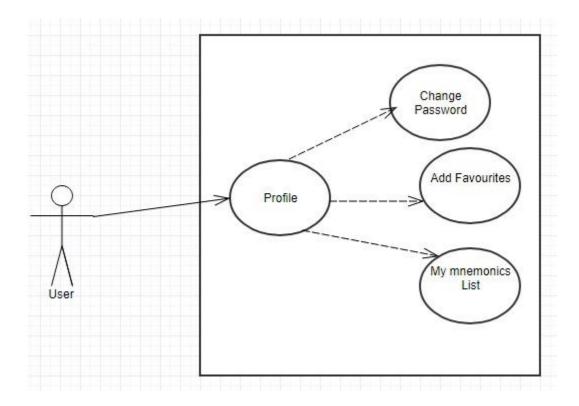


Figure-6 Profile Use Case Diagram

Add Mnemonics Use Case Diagram

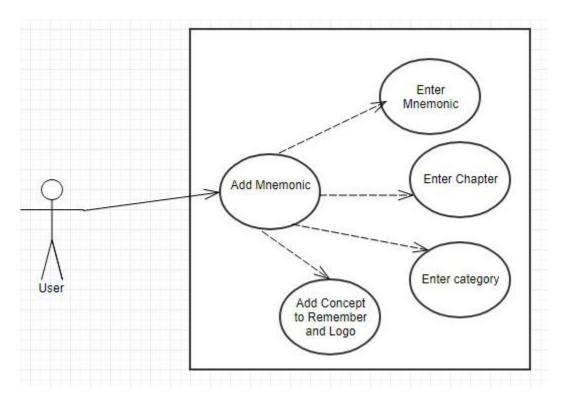


Figure-7 Add Mnemonics Use Case Diagram

Login Use Case Diagram

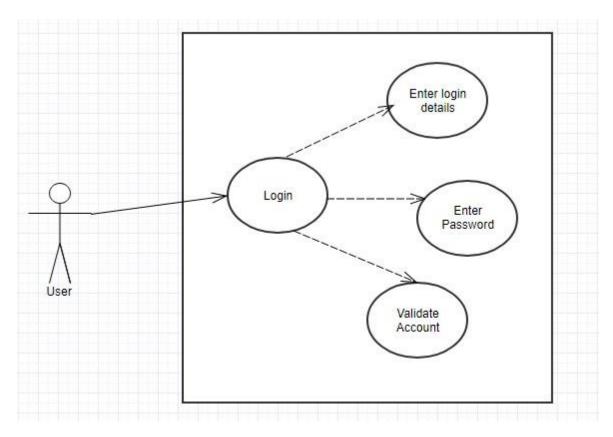


Figure-8 Login Use Case Diagram

Design Pattern

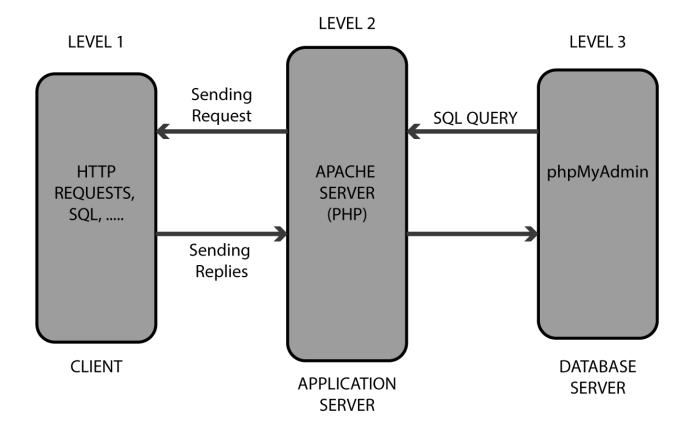


Figure- Design Architecture

Agile process

Agile programming advancement portrays an arrangement of qualities and standards for programming improvement under which prerequisites and arrangements advance through the communitarian exertion of self-sorting out cross-useful teams (Collier, 2011). It advocates versatile arranging, transformative improvement, early conveyance, and ceaseless change, and it energizes quick and adaptable reaction to change (Agile Alliance, 2017). These standards bolster the definition and proceeding with development of numerous product improvement methods (Larman, 2004).

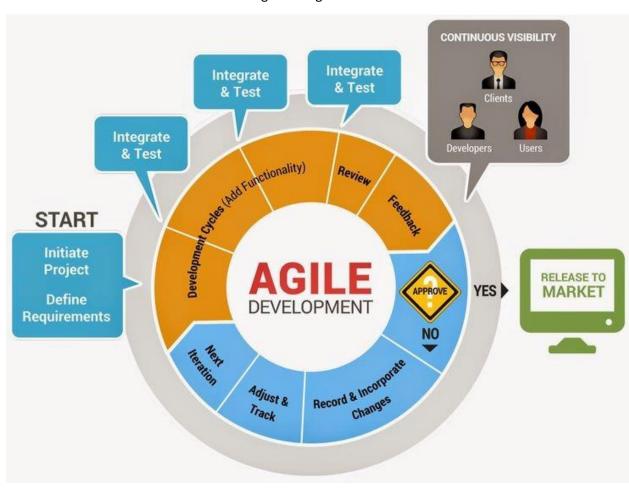


Figure- 1 Agile Process

Our project could be more defined in 4 increments. The first increment was the basic prototype thing which showed the basic functionality of the mnemonics database. Mnemonic favorites, rating of mnemonics and average rating features are added. The second increment of the project witnessed email verification and functioning of the forgot password button. The third increment had a new design with more sharp buttons and an added swear control mechanism. The fourth and final increment of the project saw the categories coming up with a brand logo upload option at the time of new mnemonic addition.

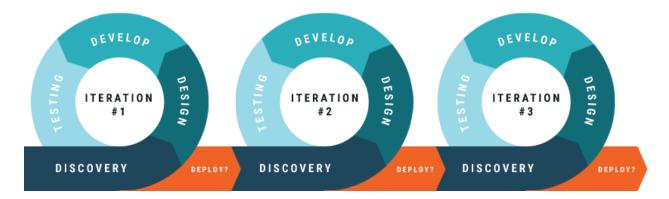


Figure- 2 Agile Iteration

Deliverables at the end of 4 iterations:

Iteration 1: working prototype with working functionalities as rating of mnemonics, favoriting a mnemonic, average rating of mnemonics etc.

Iteration 2: Email verification process and forgot password mechanism. Mnemonics added by feature added in this iteration.

Iteration 3: New design with sharper buttons and a working swear control mechanism came up.

Iteration 4: final increment with added categories option which categorize the mnemonics and a brand logo upload option at the time of adding mnemonic.

PRT452 Project report Mnemonics Database

By utilizing an Agile procedures, we can react to changing needs as the undertaking advances by focusing on profitable highlights, which are discharged in fortnightly emphasess. Working intimately with customers we can build up the highlights which would be of most advantage, an item form can be discharged which gives the most critical effect and returns. Executing Agile process for development of ventures we diminish chance through clearer perceivability, and accordingly lessen time to showcase.

For our project, we had Dr. Simon Moss Associate Professor in Psychology as our client. We had timely client meetings and the first meeting was dedicated towards requirements and then we generated user stories from then.

Pair programming

Pair Programming in software development is a light-footed programming improvement strategy in which two software engineers cooperate at one workstation. One, the driver, composes code while the other, the onlooker or navigator (Williams, n.d.) audits each line of code as it is written in. The two software engineers switch parts regularly. For our code we worked in pair and for example:

Deliverable	Description	Pair Programmers	
Categorization of mnemonics	A detailed categorization	Rahul Rao, Lin Ma	
Order by rating display of mnemonics & Browse categories option	Adding features to the website	Rahul Shokeen, Aman	
Website Development	Start the final design of website based on user stories and requirements.	Rahul Shokeen, Aman	

User story Cards

As a **client**, I want a database for mnemonics as a web application so that it is readily allows people to search for relevant mnemonics. I also want that when people enter a mnemonic, I would hope they could also specify their organization and, if possible, enter the logo of their

organization. So, people might then enter mnemonics to promote their organization. Perhaps, with the categories f mnemonics, we could also prompt people, who enter a mnemonic to specify a category. They might initially receive a drop-down menu with about 10 options: Psychology, IT, law, biology, chemistry, or other. If they press "other", they can then be prompted to write a category. Because they have already seen the previous options, they will hopefully choose a broad category. I would like no swear words in my database entered and there should be some mechanism to prevent that. I would like the users to register and once they register with their email id the email id should be verified for security purposes.

As a **user**, I need a web application which searches for the relevant mnemonics and makes my learning process easier. The web application should be providing me a personal space where I can list all my mnemonics and if necessary I can add some mnemonics for the community as well.

Testing

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC01	Verify customer Email with valid data	 Go to website. Signup here. Enter user Email. Click Submit. Verification mail sent. 	User Email: Rahulrao.v@gmail.com	User Email verification done.	User Email verification done.	Pass
TC02	Verify customer Email with valid data	 Go to website. Signup here. Enter user Email. Click Submit. Verification mail sent. 	User Email: Rahulrao.v@gmail.com	User Email verification done.	User Email verification not done.	Fail

TC03	Verify customer Email with invalid data	2. 3. 4. 5.	Go to website. Signup here. Enter user Email. Click Submit. Verification mail sent.	User Email: Rahul@gmail.com	User Email verification not done.	User Email verification not done.	Pass
TC04	Verify customer Email with invalid data	2. 3. 4. 5.	Go to website. Signup here. Enter user Email. Click Submit. Verification mail sent.	User Email: Rahul@gmail.com	User Email verification not done.	User Email verification done.	Fail
TC05	Check User login with valid data	2. 3.	Go to website. Enter Email. Enter Password. Click Submit	User Email: Rahulrao.v@gmail.com Password: *********	Login successful	Login successful	Pass
TC06	Check User login with valid data	2. 3.	Go to website. Enter Email. Enter Password. Click Submit	User Email: Rahulrao.v@gmail.com Password: *********	Login successful	Login unsuccessful	Fail
TC07	Check User login with invalid data	2. 3.	Go to website. Enter Email. Enter Password.	User Email: Rahul@gmail.com Password: ******	Login unsuccessful	Login unsuccessful	Pass

		4. Click Submit				
TC08	Check User login with invalid data	 Go to website. Enter Email. Enter Password. Click Submit 	User Email: Rahul@gmail.com Password: ********	Login unsuccessful	Login successful	Fail
TC09	Add Mnemonic	 Go to website. Login. Add Mnemonics Click submit. 	OSI Layer	Mnemonic Added	Mnemonic Added	Pass
TC10	Add Mnemonic	 Go to website. Login. Add Mnemonics Click submit. 	OSI Layer	Mnemonic added	Mnemonic not added	Fail
TC11	Add Mnemonic with swear word	 Go to website. Login. Add Mnemonics Click submit. 	S**	Mnemonic not added	Mnemonic not added	Pass
TC12	Add Mnemonic with swear word	5. Go to website.6. Login.7. Add Mnemonics8. Click submit.	S**	Mnemonic not added	Mnemonic Added	Fail

CODE

```
Login.php
<?php
```

```
session_start();
       include 'dict.inc';
       function connection($dbcon){
               if(!$dbcon){
                       die("Connection Not Established . TRY AGAIN AFTER SOME TIME " .
mysqli_connect_error());
               }
       }
function sendmail($email,$userName,$verifyhash){
               if (file_exists("phpmailer/PHPMailerAutoload.php")) {
                              require 'phpmailer/PHPMailerAutoload.php';
               }
               else {
                              echo "Please try later\n";
                              header('Location: sign.php');
               }
               $mail = new PHPMailer;
               $mail->isSMTP();
               $mail->Host = 'smtp.gmail.com';
               $mail->SMTPAuth = true;
               $mail->Username = "mnemonicscdu@gmail.com";
               $mail->Password = "simonmoss";
               $mail->SMTPSecure = 'ssl';
               $mail->Port = 465;
               $mail->SMTPDebug = 2;
               $mail->setFrom('mnemonics@test.com', 'Mnemonics');
               $mail->addAddress($email,$userName);
               $mail->Subject = 'Mnemonics Signup verification';
```

```
$mail->Body = '
               Hello '.$userName.',
               Your account has been created, you can login post verification.
               Please click the link below to activate your account:
               http://localhost/project-oct-
4/mnemonics/userverify.php?email='.$email.'&hash='.$verifyhash.'
               if ($mail->send()) {
                       $_SESSION['username']= $username;
                       $_SESSION['success'] = "Signed Up successfully. Verify EmailID to login.";
                       header('Location: Mnemonics/sign.php');
               } else {
                       $ SESSION['success'] = "Something went wrong! Please try again later.";
                       header('Location: Mnemonics/sign.php');
               }
       }
       function login($dbcon){
               $name = mysqli real escape string($dbcon, $ POST['username']);
               $checkPassword = mysqli_real_escape_string($dbcon, $_POST['password']);
                       $validate = "SELECT * from users WHERE UserName = '$name' limit 1";
                       $exec = mysqli_query($dbcon,$validate);
                       $array = mysqli_fetch_array($exec);
                       $currentPassword = $array['Password'];
```

```
$checkverified = $array['Email verified'];
               if(crypt($checkPassword, $currentPassword) === $currentPassword &&
$checkverified=='1'){
                       $ SESSION["username"]= "$name";
                       header('Location: /project/Mnemonics/sign.php');
               }else if(crypt($checkPassword, $currentPassword) === $currentPassword &&
$array['Email_verified']=='0'){
                       $email = $array['Email'];
                       $verifyhash= $array['VerificationHash'];
                       sendmail($email,$name,$verifyhash);
                       $_SESSION['success'] = "Email not verified! Please verify to login.";
                       header('Location: Mnemonics/sign.php');
               }else if(crypt($checkPassword, $currentPassword) !== $currentPassword ){
                       $_SESSION['success'] = "Invalid Username or Password";
                       header('Location: Mnemonics/sign.php');
               }else {
                       header('Location: sign.php');
               }
       }
       function signUp($dbcon){
               $userName = mysqli_real_escape_string($dbcon, $_POST['username']);
               $password = mysqli_real_escape_string($dbcon, $_POST['password']);
               $studentid = mysqli real escape string($dbcon, $ POST['studentid']);
               $email = mysqli_real_escape_string($dbcon, $_POST['email']);
               $course = mysqli_real_escape_string($dbcon, $_POST['course']);
               $verifyhash = md5($userName.rand(0,1000));
               $salt = '$2a$1$Ku2hb.'.rand(100000,999999).'.$';
```

```
$pass = crypt($password, $salt);
                $sign = "INSERT INTO `users` (`StudentID`, `UserName`, `Password`, `Email`,
`Course`, 'VerificationHash`, 'Email_verified`, 'LinkExpired`) VALUES ('$studentid', '$userName', '$pass',
'$email', '$course', '$verifyhash','0','0')";
                $update = mysqli_query($dbcon,$sign);
                        if($update){
                                sendmail($email,$userName,$verifyhash);
                        }
                        else{
                                $ SESSION['signup'] .= "EmailID or StudentID already exists .";
                                header('Location: sign.php');
                        }
       }
        function new_pass($dbcon){
                $password = mysqli real escape string($dbcon, $ POST['new password']);
                $studentid = $_SESSION['recovery_studentid'];
                $salt = '$2a$1$Ku2hb.'.rand(100000,999999).'.$';
                $pass = crypt($password, $salt);
                $update password = mysqli query($dbcon, "UPDATE `users` SET `Password` =
'$password' WHERE `StudentID` = '$studentid'");
               if($update_password){
                        $_SESSION['success'] .= "Password Reset Successfull!";
                        header('Location: sign.php');
               }else {
                        $_SESSION['success'] .= "Something went wrong";
                        header('Location: sign.php');
               }
        }
```

```
function sendemailrecovery($email_id,$verifyhash,$studentid){
               if (file_exists("phpmailer/PHPMailerAutoload.php")) {
                              require 'phpmailer/PHPMailerAutoload.php';
               }
               else {
                              echo "Please try later\n";
                              header('Location: sign.php');
               $mail = new PHPMailer;
               $mail->isSMTP();
               $mail->Host = 'smtp.gmail.com';
               $mail->SMTPAuth = true;
               $mail->Username = "mnemonicscdu@gmail.com";
               $mail->Password = "simonmoss";
               $mail->SMTPSecure = 'ssl';
               $mail->Port = 465;
               $mail->SMTPDebug = 2;
               $mail->setFrom('mnemonics@test.com', 'Mnemonics');
               $mail->addAddress($email_id,");
               $mail->Subject = 'Mnemonics Password Recovery';
               $mail->Body = '
               Hello,
               You can reset your password by clicking the link below.
               -----
               Reset Password:
               http://localhost/project-oct-
4/mnemonics/passwdrecovery.php?studentid='.$studentid.'&hash='.$verifyhash.'
```

```
١;
               if ($mail->send()){
                       $_SESSION['username']= $username;
                       $_SESSION['success'] = "Recovery Link has been sent.";
                       header('Location: Mnemonics/sign.php');
               } else {
                       $_SESSION['success'] = "Something went wrong! Please try again later.";
                       header('Location: Mnemonics/sign.php');
               }
       }
       function sendrecoverymail($dbcon,$email id){
               $id = mysqli_query($dbcon,"SELECT StudentID,UserName from users where Email =
'$email id' limit 1");
               if (mysqli_num_rows($id)>0) {
                       $data = mysqli fetch array($id);
                       $studentid = $data[0];
                       $username = $data[1];
                       $verifyhash = md5($studentid.rand(0,1000));
                       $check_from_recovery = mysqli_query($dbcon,"SELECT RecoveryHash AND
RecoveryTimeStamp from recovery where StudentID = '$studentid' limit 1");//or
die(mysqli_error($dbcon));
                       if (mysqli_num_rows($check_from_recovery)>0) {
                              $data_arr = mysqli_fetch_array($check_from_recovery);
                              $r hash = $data arr[0];
                              r = \frac{1}{r}
                              $overwrite previous recovery hashes =
mysgli query($dbcon,"UPDATE recovery SET PreviousRecoveryHash='$r hash',
PreviousRecoveryTimeStamp= '$r hash ts' where StudentID = '$studentid' ") ;//or
die(mysqli_error($dbcon));
```

```
$update_new_recovery = mysqli_query($dbcon,"UPDATE recovery SET
RecoveryHash = '$verifyhash' and RecoveryTimeStamp =TIMESTAMP where StudentID = '$studentid' ")
;//or die(mysqli error($dbcon));
                               if($update new recovery){
                                       sendemailrecovery($email_id,$verifyhash,$studentid);
                               }
                       }else {
                               $fresh insert = mysqli query($dbcon,"INSERT INTO
recovery(StudentID,RecoveryHash)VALUES('$studentid','$verifyhash') ");//or die(mysqli_error($dbcon));
                               if($fresh_insert){
                                       sendemailrecovery($email_id,$verifyhash,$studentid);
                               }
                       }
               }
       }
       function forgot($dbcon){
               $email = mysqli_real_escape_string($dbcon, $_POST['recover_email']);
               $verify = mysqli query($dbcon, "SELECT `Email` FROM `users` WHERE `Email` = '$email'
limit 1");// or die(mysqli_error($dbcon));
               if (!$verify) {
                       die('Error: '.mysqli error($dbcon));
               }
               if(mysqli num rows($verify) >0){
                       $email_tosend = mysqli_fetch_array($verify);
                       $email_id = $email_tosend[0];
                       sendrecoverymail($dbcon,$email_id);
               }else{
                       die('Error');
               }
       }
```

```
$count = count($_POST);
        if (isset($_POST['username'])) {
               if($count==2){
                       login($dbcon);
               }
               else{
                       signUp($dbcon);
               }
        }
        if(isset($_POST['recover_email'])){
               forgot($dbcon);
       }
        if(isset($_POST['new_password'])){
               new_pass($dbcon);
        }else {
               header('Location: sign.php');
}
?>
Sign up.php
<?php
session_start();
include 'dict.inc';
if(isset($_SESSION["username"])){
   header('Location: Mnemonics');
}
else{
 ?>
<!DOCTYPE html>
```

```
<html>
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, height=device-height,initial-scale=1.0">
  <meta name="author" content="Aman">
  <meta name="">
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
  <link rel="stylesheet" type="text/css" href="css/loginpage.css">
  k rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  <link rel="stylesheet" type="text/css" href="">
  <title>Login to continue</title>
  <script type="text/javascript">
  $(document).ready(function() {
       $("#userName").keyup(function(){
               $("#username_small").html('checking...');
                       if(this.value.length > 3)
                       {
                               $("#username_small").html('checking...');
                               $.ajax({
                               type: 'POST',
                               url: 'username_check.php',
                               data: {username:this.value},
                               success: function(data)
```

```
{
                                             $("#username_small").html(data);
                                           }
                               });
                                return false;
               }else{
                       $("#result").html(");
               }
       });
$("#StudentID").keyup(function(){
 $("#studentid_unique").html('checking...');
  if(this.value.length > 3)
  {
   $("#studentid_unique").html('checking...');
   $.ajax({
   type: 'POST',
   url: 'studentid_check.php',
   data: {studentid:this.value},
   success : function(data)
       {
         $("#studentid_unique").html(data);
       }
   });
   return false;
 }else{
  $("#result").html(");
 }
});
```

```
$("#userEmail").keyup(function(){
```

 $\label{thm:policy:continuous:co$

```
var emailAddress = $(this).value;
 if(!pattern.test(emailAddress)){
 $("#email unique").html('checking...');
  if(this.value.length > 3)
  {
   $("#email unique").html('checking...');
   $.ajax({
   type: 'POST',
   url: 'email_check.php',
   data: {email:this.value},
   success: function(data)
       {
          $("#email_unique").html(data);
        }
   });
   return false;
 }else{
  $("#result").html(");
 }
}
});
 $('#signup4').prop('disabled', true);
```

```
$('#inputPassword4, #inputConfPassword4').on('keyup', function () {
       if (this.value.length< 3) {</pre>
              $('#message').html(");
              }
else{
if ($('#inputPassword4').val() == $('#inputConfPassword4').val()) {
  $('#message').html('Matching').css('color', 'green');
  $('#signup4').prop('disabled', false);
} else {
  $('#message').html('Not Matching').css('color', 'red');
  $('#signup4').prop('disabled', true);
}
}
});
});
  </script>
 </head>
 <body>
  <div class="container">
   <div class="row">
    <div class="form">
     <a href="index.php">Home</a>
     <a data-toggle="pill" href="#login">Login</a>
```

```
class="col-md-2 tab"><a data-toggle="pill" href="#SignUp">SignUp</a>
      cli class="col-md-3 tab"><a data-toggle="pill" href="#forgot">Forgot Password</a>
     <div class="tab-content">
      <div id="login" class="tab-pane fade in active ">
       <h3>Login</h3>
       <form method="POST" action="login.php" id="login_">
        <div class="form-group">
         <label for="UserName">Username</label>
         <input type="text" class="form-control" id="UserName" name="username" aria-</pre>
describedby="emailHelp" placeholder="Enter Username">
         <small id="emailHelp" class="form-text text-muted"></small>
        </div>
        <div class="form-group">
         <label for="password">Password</label>
         <input type="password" class="form-control" id="password" name="password"
placeholder="Password">
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
       </form>
      </div>
      <div id="SignUp" class="tab-pane fade">
       <h3>SignUp</h3>
       <form action="login.php" method="post" id="signup_" enctype="multipart/form-data">
        <div class="form-row">
         <div class="form-group col-md-6">
          <label for="email" class="col-form-label">Email</label>
          <input type="email" name="email" class="form-control" id="userEmail" placeholder="Email"</pre>
required>
          <small id="email_unique"></small>
         </div>
```

```
<div class="form-group col-md-6">
          <label for="userName" class="col-form-label">Username</label>
          <input type="text" pattern=".{6,20}" required name="username" class="form-control"</pre>
id="userName">
          <small id="username small">Enter a preferrend username min 6 characters</small>
         </div>
        </div>
        <div class="form-row">
         <div class="form-group col-md-6">
          <label for="password" class="col-form-label">Password</label>
          <input pattern=".{8,20}" type="password" name="password" class="form-control"
id="inputPassword4" placeholder="Password (Min 8 Characters)" required>
         </div>
          <div class="form-group col-md-6">
          <label for="password" class="col-form-label">Password</label>
          <input pattern=".{8,20}" type="password" name="confirmpassword" class="form-control"
id="inputConfPassword4" placeholder="Confirm Password (Min 8 Characters)" required>
          <small id="message"></small>
         </div>
        </div>
        <div class="form-group">
         <label for="StudentID">StudentID</label>
         <input type="text" name="studentid" class="form-control" id="StudentID">
         <small id="studentid_unique"></small>
        </div>
        <div class="form-group">
         <label for="course" class="col-form-label">Course</label>
         <input type="text" pattern="{3,20}" id="course" name="course" class="form-control"</pre>
required>
        </div>
        <div class="form-group">
```

```
</div>
        <button type="submit" class="btn btn-primary" id="signup4">Sign Up</button>
       </form>
      </div>
      <div id="forgot" class="tab-pane fade">
       <h3>Enter EmailID to recover password</h3>
       <form method="POST" action="login.php" id="forgot_">
        <div class="form-group">
         <label for="EmailID">EmailID</label>
         <input type="email" class="form-control" id="email" name="recover email" aria-
describedby="emailHelp" placeholder="Enter EmailID">
         <small id="emailHelp" class="form-text text-muted"></small>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
       </form>
      </div>
     </div>
     <?php
       if(isset($_SESSION['signup'])){
       echo $_SESSION['signup'];
       echo "Sign Up failed try again";
       if(isset($ SESSION['success'])){
       echo $_SESSION['success'];
       // echo "Sign Up failed try again";
       }
       if(isset($_SESSION['verified'])){
```

```
echo "Email Successfully verified. Login to continue.";
       }
       ?>
    </div>
   </div>
  </div>
 </body>
</html>
<?php
$_SESSION= [];
session_destroy();
}
 ?>
Index.php
<?php
session_start();
if(!isset($_SESSION['username'])){
   header('Location: ../');
}
 else{
   require 'dict.inc';
   $username = $_SESSION['username'];
   $sql_for_user1 = "SELECT `StudentID` FROM `users` WHERE `UserName` = '$username'";
   $sql_for_user_query1 = mysqli_query($dbcon, $sql_for_user1);
   $userID1 = mysqli_fetch_row($sql_for_user_query1);
```

```
$user1 = $userID1[0];
   $sql get favourites list = mysqli query($dbcon, "SELECT * FROM `userfavourites` WHERE
`StudentID` = '$user1' AND `IsFavourite` = '1'");
   $my_mnemonics = mysqli_query($dbcon, "SELECT * FROM `keywords` WHERE `AddedBy` =
'$username'");
}
if(isset($_GET['id'])){
   $keyid = $_GET['id'];
   $sql = "SELECT * FROM `keywords` WHERE `KeyID` = '$keyid'";
   $sql query = mysqli query($dbcon, $sql);
   $row = mysqli_fetch_row($sql_query);
   $category id = $row[11];
   $brand_logo = $row[12];
   4 = \frac{1}{2}
   $category_name_sql = mysqli_query($dbcon, "SELECT `CategoryName` FROM `categories` WHERE
`CategoryID` = '$category id'");
   $category name = mysqli fetch row($category name sql);
   $category name = $category name[0];
   $username = $_SESSION['username'];
   $sql_for_user = "SELECT `StudentID` FROM `users` WHERE `UserName` = '$username'";
   $sql_for_user_query = mysqli_query($dbcon, $sql_for_user);
   $userID = mysqli_fetch_row($sql_for_user_query);
   $user = $userID[0];
   $get rating = mysqli query($dbcon, "SELECT * FROM `userkeysrelation` WHERE `StudentID` = '$user'
AND `KeyID` = '$keyid'");
   $get_rating_value = mysqli_fetch_row($get_rating);
   $get_fav_or_not = mysqli_query($dbcon, "SELECT * FROM `userfavourites` WHERE `StudentID` =
'$user' AND `KeyID` = '$keyid'");
   $get fav or not value = mysqli fetch row($get fav or not);
}
$category = mysqli_query($dbcon, "SELECT * FROM `categories`");
```

```
$category1 = mysqli query($dbcon, "SELECT * FROM `categories`");
$category list query = mysqli query($dbcon, "SELECT * FROM `categories`");
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <meta content="search here" name="description">
  <meta content="Aman " name="author">
  <meta name="">
  <meta content="width=device-width, height=device-height,initial-scale=1.0" name="viewport">
  <title>Project</title>
  k href="semantic/dist/semantic.min.css" rel="stylesheet">
  <link href="semantic/dist/components/reset.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/site.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/grid.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/card.min.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/icon.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/button.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/card.css" rel="stylesheet" type="text/css">
  k href="semantic/dist/components/dropdown.min.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/label.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/image.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/reveal.css" rel="stylesheet" type="text/css">
  k href="semantic/dist/components/dimmer.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/rating.css" rel="stylesheet" type="text/css">
  k href="semantic/dist/components/transition.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/popup.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/modal.css" rel="stylesheet" type="text/css">
  <link href="semantic/dist/components/accordion.css" rel="stylesheet" type="text/css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="semantic/dist/components/popup.js" type="text/javascript"></script>
<script src="semantic/dist/components/dropdown.js" type="text/javascript"></script>
<script src="semantic/dist/components/dimmer.js" type="text/javascript"></script>
<script src="semantic/dist/components/rating.js" type="text/javascript"></script>
<script src="semantic/dist/components/transition.js" type="text/javascript"></script>
<script src="semantic/dist/components/modal.js" type="text/javascript"></script>
<script src="semantic/dist/components/accordion.js" type="text/javascript"></script>
<style type="text/css">
html {
position: relative;
min-height: 100%;
}
body {
margin: 0 0 150px;
}
footer {
position: absolute;
left: 0;
bottom: 0;
}
</style>
<script type="text/javascript">
$(document).ready(function() {
   $(".favourites").on('click', function(){
     $('#fav modal').modal('show');
   });
   $(".my mnemonics").on('click', function(){
     $('#my_mnemonics').modal('show');
   });
```

```
$(".add_Mnemonic").on('click', function(){
    $('#add_mne_modal').modal('show');
  });
  $(".change_password").on('click', function(){
    $('#change_pass_modal').modal('show');
  });
  $(".categories_list").on('click', function(){
    $('#category_list_modal').modal('show');
    $('.ui.accordion').accordion();
  });
 $('.ui.dropdown').dropdown();
$(".ui.message").hide();
  <?php
 if(isset($_GET['id'])){
 ?>
  $('.ui.star.rating')
   <?php
 if($get_rating_value != NULL){
   ?>
       .rating('disable');
       <?php
 }
 else{
   ?>
        .rating();
       <?php
 }
 ?>
      $('.ui.heart.rating').rating();
      $("#rate").on('click',function(){
```

```
getRating();
   function getRating() {
     var rating = $('#rate').rating("get rating");
     // alert(rating);
     var res = "
      $.ajax({
        type: "POST",
        url: 'rate.php',
        data: {rate:rating, word:'<?php echo $_GET['id'] ?>' },
        success: function (data) {
          //alert(data);
          $('#rate')
          .rating('disable');
        },
        error: function(jqXHR, text, error){
        // Displaying if there are any errors
         $('#result').html(error);
        }
     });
   }
 });
$("#fav").on('click',function(){
  getfav();
  function getfav() {
    var fav = $('#fav').rating("get rating");
    // alert(fav);
    var res = "
    $.ajax({
       type: "POST",
```

```
url: 'rate.php',
           data: {fav:fav, word:'<?php echo $_GET['id'] ?>' },
           success: function (data) {
           },
           error: function(jqXHR, text, error){
           // Displaying if there are any errors
            $('#result').html(error);
           }
        });
      }
    });
 <?php
}
?>
 $( "p" ).removeClass( "myClass yourClass" )
 $(".todelete").remove();
 $("#formID").submit(function(event) {
   $(".card").remove();
   $( ".top" ).remove();
   //to prevent normal form submition and reloading page
   event.preventDefault();
   var formDetails = $('#SearchInput').val();
   if (formDetails.length > 0) {
     $(".todelete").remove();
     $.ajax({
       type: "POST",
       url: 'aj.php',
       data: {word:formDetails},
       success: function (data) {
        var jsonData = JSON.parse(data);
```

```
objData = jsonData[0];
             catData = jsonData[1];
             catColor = jsonData[2];
              if (objData['None'] == 0) {
              var res = "<div class=\"todelete\"><div class=\"card\"> 
menu\">";
              for (var link in objData){
               if (link != 'None')
                res = res + "<a class='ui "+catColor[catData[link]]+" ribbon
label'>"+catData[link]+"</a><a class=\"item\" href = \""+objData[link]+"\">"+link+"<\/a>";
               }
                res = res + "<\/div>";
                $(".toappend").append(res);
              }
              else{
               var res = "<div class=\"todelete\"><div class=\"card\"> <ul class=\"ui fluid vertical
menu\">";
                res = res + "<a class=\"item\">No Results<\/a>";
                res = res + "<\/div>";
                $(".toappend").append(res);
              }
            },
            error: function(jqXHR, text, error){
           // Displaying if there are any errors
             $('#result').html(error);
            }
         });
       }
       else {
         $(".attach_top").append( "<div class=\" top\"><\/div>" );
          $(".todelete").remove();
```

```
}
  return false;
});
$("#category_id").change(function(){
  var is_other = $(this).val();
  if(is\_other == 1){
   $('#new_category').show();
   $('#new_category').prop('required', true);
   $('#new_category').attr('name', 'new_category');
  }
  else{
   $('#new_category').hide();
   $('#new_category').prop('required', false);
   $('#new_category').removeAttr('name');
  }
});
$("#category_id_edit").change(function(){
  var is_other = $(this).val();
  if(is_other == 1){
   $('#new_category_edit').show();
   $('#new_category_edit').prop('required', true);
   $('#new_category_edit').attr('name', 'new_category_edit');
  }
  else{
   $('#new_category_edit').hide();
   $('#new_category_edit').prop('required', false);
   $('#new_category_edit').removeAttr('name');
  }
});
```

```
$("#add_Mnemonic").submit(function(event) {
       event.preventDefault();
  //to prevent normal form submition and reloading page
  var input = document.getElementById('file');
  var file = input.files[0];
  if (input.files[0]) {
   if(file.size< 200000){
   var formDetails = new FormData(this);
     $.ajax({
       type: "POST",
       url: 'add_mnemonic.php',
       data: formDetails,
       success: function (data) {
         if (data.indexOf("Image") < 0){</pre>
          $("#uimessage").html(data);
          $("#uimessage").show();
          $('#add_mne_modal').modal('hide');
         }
         else{
          $('#error message add mne modal').html(data);
        }
       },
       error: function(jqXHR, text, error){
       // Displaying if there are any errors
        $('#result').html(error);
       },
       cache: false,
```

```
contentType: false,
      processData: false
    });
  return false;
 }
 else{
  alert("UPload limit exceeded")
 }
}
 else{
  var formDetails = new FormData(this);
    $.ajax({
      type: "POST",
      url: 'add_mnemonic.php',
      data: formDetails,
      success: function (data) {
       if (data.indexOf("Image") < 0){</pre>
        $("#uimessage").html(data);
        $("#uimessage").show();
        $('#add_mne_modal').modal('hide');
       }
       else{
        $('#error_message_add_mne_modal').html(data);
       }
      },
      error: function(jqXHR, text, error){
      // Displaying if there are any errors
       $('#result').html(error);
      },
```

```
cache: false,
        contentType: false,
        processData: false
     });
   return false;
  }
});
$(".formedit").submit(function(event) {
 event.preventDefault();
// alert("kjhsd");
  // var input = document.getElementById('file_edit');
  // var file = input.files[0];
  // alert(file);
  // if (input.files[0]) {
  // if(file.size< 200000){
    var edit_form_details = $(this).serializeArray();
    len = edit_form_details.length;
    dataObj = {};
    for (i=0; i<len; i++) {
     dataObj[edit_form_details[i].name] = edit_form_details[i].value;
    }
    // alert(dataObj['edit_mnemonic_value']);
    var keyid = dataObj['edit_keyid_value'];
    $('#keyid_edit').val(keyid);
    var opt = dataObj["edit_mnemonic_category"];
```

```
$('#category_id_edit').dropdown('set selected', opt);
  $('#edit modal chapter').val(dataObj['edit chapter no value']);
  $('#add_concept_text_edit').val(dataObj['edit_conecpt_value']);
  $('#add_Mnemonic_text_edit').html(dataObj['edit_mnemonic_value']);
  $('#edit_mne_modal').modal('show');
  $('#edit_mne').submit(function(event){
   event.preventDefault();
   var input = document.getElementById('file_edit');
var file = input.files[0];
if (input.files[0]) {
 if(file.size< 200000){
 var formDetails = new FormData(this);
   $.ajax({
     type: "POST",
     url: 'edit_mnemonic.php',
     data: formDetails,
     success: function (data) {
      if (data.indexOf("Image") < 0){</pre>
       $("#uimessage").html(data);
       $("#uimessage").show();
       $('#edit mne modal').modal('hide');
      }
      else{
       $('#error_message_add_mne_modal').html(data);
      }
     },
     error: function(jqXHR, text, error){
     // Displaying if there are any errors
```

```
$('#result').html(error);
      },
      cache: false,
      contentType: false,
      processData: false
    });
  return false;
 }
 else{
  alert("UPload limit exceeded")
 }
}
 else{
  var formDetails = new FormData(this);
    $.ajax({
      type: "POST",
      url: 'edit_mnemonic.php',
      data: formDetails,
      success: function (data) {
       if (data.indexOf("Image") < 0){
        $("#uimessage").html(data);
        $("#uimessage").show();
        $('#edit_mne_modal').modal('hide');
       }
       else{
        $('#error_message_add_mne_modal').html(data);
       }
      },
      error: function(jqXHR, text, error){
```

```
// Displaying if there are any errors
        $('#result').html(error);
       },
       cache: false,
       contentType: false,
       processData: false
     });
   return false;
  }
     var edit_form_details_submit = new FormData(this);
    });
});
$("#add_Mnemonic_text").keyup(function(){
 var mne = $(this).val();
 $.ajax({
  type: "POST",
  url: 'ajax_swear.php',
  data: {mnemonic:mne},
  success: function (data) {
    if (data.trim() == ") {
     $('#swear_message').hide();
     $('#add mnemonic submit').prop('disabled', false);
    }else{
     $('#swear_message').show();
     $('#swear_message').html(data);
     $('#add_mnemonic_submit').prop('disabled', true);
    }
  },
  error: function(jqXHR, text, error){
```

```
// Displaying if there are any errors
   $('#result').html(error);
  }
 });
});
$("#add_concept_text").keyup(function(){
 var mne = $(this).val();
 $.ajax({
  type: "POST",
  url: 'ajax_swear.php',
  data: {mnemonic:mne},
  success: function (data) {
    if (data.trim() == ") {
     $('#swear message concept').hide();
     $('#add_mnemonic_submit').prop('disabled', false);
    }else{
     $('#swear_message_concept').show();
     $('#swear_message_concept').html(data);
      $('#add_mnemonic_submit').prop('disabled', true);
    }
  },
  error: function(jqXHR, text, error){
  // Displaying if there are any errors
   $('#result').html(error);
  }
 });
});
```

```
$("#add_Mnemonic_text_edit").keyup(function(){
 var mne = $(this).val();
 $.ajax({
  type: "POST",
  url: 'ajax_swear.php',
  data: {mnemonic:mne},
  success: function (data) {
    if (data.trim() == ") {
     $('#swear_message_edit').hide();
     $('#add_mnemonic_submit_edit').prop('disabled', false);
    }else{
     $('#swear_message_edit').show();
     $('#swear_message_edit').html(data);
      $('#add mnemonic submit edit').prop('disabled', true);
    }
  },
  error: function(jqXHR, text, error){
  // Displaying if there are any errors
   $('#result').html(error);
  }
 });
});
$("#add concept text edit").keyup(function(){
 var mne = $(this).val();
 $.ajax({
  type: "POST",
  url: 'ajax_swear.php',
  data: {mnemonic:mne},
  success: function (data) {
    if (data.trim() == ") {
```

```
$('#swear_message_concept_edit').hide();
     $('#add mnemonic submit edit').prop('disabled', false);
    }else{
     $('#swear_message_concept_edit').show();
     $('#swear_message_concept_edit').html(data);
      $('#add_mnemonic_submit_edit').prop('disabled', true);
    }
  },
  error: function(jqXHR, text, error){
  // Displaying if there are any errors
   $('#result').html(error);
  }
 });
});
$('.del_button, .undo_delete').click(function(event){
 event.preventDefault();
 var buton_id = $(this).attr('id');
 buton_id = buton_id.slice(3);
 var action = $(this).attr('class');
 $.ajax({
      type: "POST",
      url: 'del_mne.php',
      data: {buton id:buton id, delete: action},
      success: function (data) {
        if (data.indexOf("deleted") >= 0){
          $('#del'+buton_id).removeClass('del_button').addClass('undo_delete');
          $("#row"+buton_id).addClass('negative');
          $("#del icon"+buton id).removeClass('delete').addClass('undo');
        }
        else{
```

```
$("#del_icon"+buton_id).addClass('delete').removeClass('undo');
          $("#row"+buton id).removeClass('negative');
        }
      },
      error: function(jqXHR, text, error){
      // Displaying if there are any errors
       $('#result').html(error);
      }
    });
});
$("#change_pass_modal_form").submit(function(event) {
  //to prevent normal form submition and reloading page
  event.preventDefault();
  var formDetails = $('#change_pass_modal_form').serializeArray();
    $.ajax({
      type: "POST",
      url: 'change_pass.php',
      data: formDetails,
      success: function (data) {
        $("#uimessage_password").html(data);
        $("#uimessage password").show();
        $('#change_pass_modal').modal('hide');
      },
      error: function(jqXHR, text, error){
      // Displaying if there are any errors
       $('#result').html(error);
      }
    });
```

```
return false:
     });
  });
  </script>
 </head>
 <body>
  <!-- <div class="ui stackable center aligned grid container"> -->
  <div class="ui stackable eight column center aligned pointing menu grid container">
   <a class="column active item" href=".">Home</a>
   <a class="column item add_Mnemonic">Add a Mnemonic</a>
   <a class="column item favourites">Favourites</a>
   <a class="column item my_mnemonics">My Mnemonics</a>
   <a class="column item categories_list">Categories</a>
   <a class="column item change_password">Change Password</a>
   <b>&nbsp;<?php echo " ".$_SESSION["username"];?></b>
   <a class="column item " href="../logout.php">Sign out</a>
  </div>
  <div class="ui message" id="uimessage">
  </div>
  <div class="ui message pass" id="uimessage_password">
  </div>
  <div class="ui one column centered grid container">
   <div class="column">
   </div>
   <div class="column">
    <div class="ui segment">
     <form id="formID" method="post" name="formID">
      <div class="ui transparent icon input ui fluid category search">
       <input class="form-control searchbox" id="SearchInput" placeholder="Search for..."
type="text"> <button class="btn btn-default searchbox search link icon" hidden="True"
type="submit"></button>
```

```
</div>
  </form>
 </div>
</div>
<div class="column">
</div>
<div class="column">
 <div class="toappend">
  <!--
   This div is used to append cards when a successsfull ajax request is made
   and cards are returned
  <div class="todelete">
   <!--
    This div will get deleted when an empty ajax request is made
    -->
  </div>
 </div>
</div>
<?php if (isset($_GET['id'])) {</pre>
 # code...
 ?>
<div class="column">
 <div class="ui container center aligned card" style="width: 100%; text-align: center;">
  <div class="extra content">
   <div class="ui equal width grid">
    <span class="left floated column">Category: <?php echo $category_name; ?></span>
    <span class="column"></span>
    <span class="right floated like column">
     Favorite
```

```
<div class="ui heart rating fav" id="fav" data-rating="<?php echo $get_fav_or_not_value[3];</pre>
?>" data-max-rating="1"></div>
       </span>
      </div>
     </div>
     <div class="content">
      <div class="header">
       <h2> Chapter Title: <?php echo $row[1]; ?></h2>
      </div>
      <div class="description">
       <span>
        <h4>Concept To Remember: </h4>
       </span>
       <?php echo $row[2]; ?>
       <span>
        <h4>Mnemonic: </h4>
       </span>
       <?php echo $row[3]; ?>
      </div>
     </div>
     <div class="extra content">
      <div class="ui equal width grid">
       <span class="left floated column">
       Average Rating: <?php echo round($row[7], 2); ?>
       </span>
       <span class=" floated column">Added by: <?php echo $row[8]; ?> &nbsp<img src="<?php echo</pre>
$brand_logo; ?>" style="height: 25px;"> </span>
       <span class="right floated column">
        Rate:
        <div class="ui star rating" id="rate" data-rating="<?php echo $get_rating_value[3]; ?>" data-
max-rating="5"></div>
```

```
</span>
      </div>
     </div>
     <?php
     }
       ?>
    </div>
  </div>
  </div>
  <div class="ui modal mnemonic" id="add_mne_modal">
   <div class="header">Add a New Mnemonic</div>
   <div class="content">
    <form class="ui form" id="add_Mnemonic" enctype="multipart/form-data">
     <div class="field">
      <label>Chapter No</label>
      <input type="number" step="0.01" name="chapter_no" placeholder="0.0" required>
     </div>
     <div class="field">
      <div class="field">
       <label>Concept to remember</label>
       <input type="text" name="concept_remember" placeholder="Concept"id="add_concept_text"
required>
       <small id="swear_message_concept"></small>
      </div>
      <div class="field">
       <label>Mnemonic</label>
       <textarea type="text" name="mnemonic" placeholder="Mnemonic" id="add_Mnemonic_text"
required></textarea>
       <small id="swear_message"></small>
      </div>
     </div>
```

```
<div class="two fields">
      <div class="field">
       <select class="ui search dropdown fluid" id="category_id" name="category_id" style="float:</pre>
left;">
        <option value="">Category</option>
        <?php
         while ($row = mysqli fetch row($category)) {
          echo "<option value='$row[1]'>$row[0]</option>";
         }
         ?>
        <option value="1">Others</option>
       </select>
      </div>
      <div class="field">
       <input type="text" placeholder="Enter a new category name" id="new category"
style="display: none; float: right;">
      </div>
     </div>
     <div class="field">
      <div>
       <label>Brand Logo:&nbsp&nbsp</label>
       <small id="swear message">Max size 200kb</small>
       <label for="file" class="ui icon button">
       <i class="file icon"></i>
       Open File</label>
       <input type="file" name="brand_image" id="file" style="display:none">
      </div>
     </div>
     <div class="actions">
      <button class="ui button" id="add mnemonic submit" type="submit">Submit</button>
      <div class="ui cancel button">Cancel</div>
```

```
</div>
     <small id="error message add mne modal"></small>
    </form>
   </div>
  </div>
  <div class="ui modal mnemonic" id="edit_mne_modal">
   <div class="header">Edit Mnemonic</div>
   <div class="content">
    <form class="ui form" id="edit mne" enctype="multipart/form-data">
     <div class="field">
      <label>Chapter No</label>
      <input type="number" step="0.01" id="edit modal chapter" name="chapter no edit"</pre>
placeholder="0.0" required>
     </div>
     <div class="field">
      <div class="field">
       <label>Concept to remember</label>
       <input type="text" name="concept remember edit" placeholder="Concept"
id="add_concept_text_edit" required>
       <small id="swear_message_concept_edit"></small>
      </div>
      <div class="field">
       <label>Mnemonic</label>
       <textarea type="text" name="mnemonic_edit" placeholder="Mnemonic"
id="add_Mnemonic_text_edit" required></textarea>
       <small id="swear message edit"></small>
      </div>
     </div>
     <div class="two fields">
      <div class="field">
       <select class="ui search dropdown fluid" id="category_id_edit" name="category_id_edit"
style="float: left;">
```

```
<option value="">Category</option>
        <?php
         while ($row1 = mysqli_fetch_row($category1)) {
          echo "<option value='$row1[1]'>$row1[0]</option>";
         }
         ?>
        <option value="1">Others</option>
       </select>
      </div>
      <div class="field">
       <input type="text" placeholder="Enter a new category name" id="new category edit"</pre>
style="display: none; float: right;">
      </div>
     </div>
     <div class="field">
      <div>
       <label>Brand Logo:&nbsp&nbsp</label>
       <label for="file edit" class="ui icon button">
       <i class="file icon"></i>
       Open File</label>
       <input type="file" name="brand_image_edit" id="file_edit" style="display:none">
      </div>
     </div>
     <input type="text" name="keyid edit" id="keyid edit" style="display: none;">
     <div class="actions">
      <button class="ui button" id="add mnemonic submit edit" type="submit">Submit</button>
      <div class="ui cancel button">Cancel</div>
     </div>
     <small id="error_message_edit_mne_modal"></small>
    </form>
```

```
</div>
</div>
<div class="ui modal" id="my_mnemonics">
<div class="header">
 Mnemonics added by you
</div>
<div class="scrolling content">
 <thead>
  Chapter No
   Concept to Remember
   Mnemonic
   Category
  </thead>
  <?php
   while($my_mnemonics_list = mysqli_fetch_row($my_mnemonics)){
    if($my_mnemonics_list == NULL){
     echo "Add Mnemonics to view.";
    }
    if (\text{$my mnemonics list}[10] == 0) {
     echo "";
   }else{
   echo "";
   }
    echo "<form method='POST' id='$my_mnemonics_list[4]' class='formedit'>";
```

```
echo " <input type='text' style='display: none;' value='$my mnemonics list[4]'
name='edit keyid value'> <input type='text' style='display: none;' value='$my mnemonics list[1]'
name='edit chapter no value'> <input type='text' style='display: none;'
value='$my mnemonics list[2]' name='edit conecpt value'> <input type='text' style='display: none;'
value='$my mnemonics list[3]' name='edit mnemonic value'> <input type='text' style='display:
none; 'value='$my mnemonics list[11]' name='edit mnemonic category'>";
        echo "".$my mnemonics list[1]."";
       echo "".$my mnemonics list[2]."";
        echo "".$my mnemonics list[3]."";
        $category name sql my mne = mysqli query($dbcon, "SELECT `CategoryName` FROM
`categories` WHERE `CategoryID` = '$my mnemonics list[11]'");
       $category name my mne = mysqli fetch row($category name sql my mne);
       $category_name_my_mne = $category_name_my_mne[0];
        echo "".$category name my mne."";
        if (\text{$my mnemonics list}[10] == 0) {
         echo "<button class='circular ui icon button' type='submit'><i class='edit
icon'></i></button><button id='del$my mnemonics list[4]' class='circular ui icon button del button'
><i id='del_icon$my_mnemonics_list[4]' class='delete icon'></i></button>";
        }else{
        echo "<button class='circular ui icon button' type='submit'><i class='edit
icon'></i></button><button id='del$my mnemonics list[4]' class='circular ui icon button undo delete'
><i id='del icon$my mnemonics list[4]' class='undo icon'></i></button>";
       }
        echo "</form>";
       echo "";
       }
       ?>
     </div>
  </div>
  <div class="ui modal" id="fav_modal">
   <div class="header">
   Your Favourites
```

```
</div>
  <div class="scrolling content">
   <thead>
     Concept to Remember
      Mnemonic
     </thead>
    <?php
      while($sql_get_favourites_list_values = mysqli_fetch_row($sql_get_favourites_list)){
       if($sql_get_favourites_list_values == NULL){
        echo "Add favorites to view.";
       }
       echo "";
       $sql_get_word_list = mysqli_query($dbcon, "SELECT `ConceptToRemember`, `Mnemonics`
FROM `keywords` WHERE `KeyID` = '$sql_get_favourites_list_values[2]'");
       $sql_get_word_list_values = mysqli_fetch_row($sql_get_word_list);
       echo "".$sql_get_word_list_values[0]."";
       echo "".$sql_get_word_list_values[1]."";
       echo "";
      }
      ?>
    </div>
 </div>
 <div class="ui modal" id="category_list_modal">
  <div class="header">Categories</div>
```

```
<div class="content">
    <div class="ui accordion">
     <?php
      while($row = mysqli_fetch_row($category_list_query)){
      ?>
     <div class="title " id="<?php echo $row[1]; ?>">
      <i class="dropdown icon "></i>
      <?php
       echo $row[0];
       $sql_cat_list = "SELECT * FROM `keywords` WHERE (`CategoryID` = '$row[1]') AND (`Deleted` =
'0')";
       $sql_query_cat_list = mysqli_query($dbcon,$sql_cat_list);
       ?>
     </div>
     <div class="content">
      <?php while($row_cat_list = mysqli_fetch_row($sql_query_cat_list))</pre>
echo "<a class=\"ui big label\" href='?id=$row_cat_list[4]'>$row_cat_list[2]</a>"; ?>
     </div>
     <?php
     }
      ?>
    </div>
  </div>
  </div>
  <div class="ui modal change_pass" id="change_pass_modal">
   <div class="header">Change Password</div>
   <div class="content">
    <form class="ui form" id="change_pass_modal_form">
     <div class="field">
```

```
<div class="field">
       <label>Present Password</label>
       <input type="password" name="pre pass" placeholder="present password" required>
      </div>
      <div class="field">
       <label>New Password</label>
       <input type="password" name="new pass" placeholder="min 8 characters" pattern=".{8,20}"
required>
       <small id="new_password_small_text">Enter a preferrend password min 8 characters</small>
      </div>
     </div>
     <div class="actions">
      <button class="ui button" type="submit">Submit</button>
      <div class="ui cancel button">Cancel</div>
     </div>
    </form>
   </div>
  </div>
  <style type="text/css">
   .body{
   overflow: hidden;
   }
  </style>
  <!-- <script ref="js/bootstrap.js"></script> -->
 </body</html>
Swearwords.php
<?php
$swear words =
array("anal", "anus", "arse", "ass", "ballsack", "balls", "bastard", "bitch", "biatch", "bloody", "blowjob", "blow
job","bollock","bollok","boner","boob","bugger","bum","butt","buttplug","clitoris","cock","coon","crap"
"cunt","damn","dick","dildo","dyke","fag","feck","fellate","fellatio","felching","fuck","f u c
k","fudgepacker","fudge packer","flange","Goddamn","God
```

```
damn","hell","homo","jerk","jizz","knobend","knob
end","labia","lmao","lmfao","muff","nigger","nigga","omg","penis","piss","poop","prick","pube","pussy"
,"queer","scrotum","sex","shit","s
hit", "sh1t", "slut", "smegma", "spunk", "tit", "tosser", "turd", "twat", "vagina", "wank", "whore");
$string = $_POST['mnemonic'];
$flag swear = FALSE;
$matchfound = preg match all(
        "/\b(".implode($swear words,"|").")\b/i",
        $string,
        $matches
       );
if ($matchfound) {
$words = array_unique($matches[0]);
foreach($words as $word) {
 echo "You used $word, which is offensive please remove to submit your Mnemonic <br/> ";
}
echo "";
}
?>
Passwordrecovery.php
<?php
session_start();
include 'dict.inc';
if(isset($ GET['studentid']) && !empty($ GET['studentid']) AND isset($ GET['hash']) &&
!empty($ GET['hash'])){
  $studentid = mysqli_real_escape_string($dbcon,$_GET['studentid']);
  $hash = mysqli_real_escape_string($dbcon,$_GET['hash']);
  $search = mysqli_query($dbcon,"SELECT * FROM recovery WHERE StudentID='$studentid' AND
RecoveryHash='$hash' AND LinkExpired='0' ") or die(mysqli error());
  $match = mysqli_num_rows($search);
  if(\$match > 0){
```

```
$expire_link = mysqli_query($dbcon,"UPDATE recovery SET LinkExpired='1' WHERE
StudentID='$studentid' AND RecoveryHash='$hash' ");
   if($expire_link){
    $_SESSION['recovery_studentid']= $studentid;
    $_SESSION['recovery_hash']= $hash;
    header('Location: new_pass.php');
   }else {
    echo "Something went worng! Try again later";
    header('Location: index.php');
   }
  }else{
  echo "Invalid URL/LINK EXPIRED";
  }
}else {
echo "Invalid URL/LINK EXPIREDD";
}
?>
Usernamecheck.php
<?php
session_start();
require 'dict.inc';
if(isset($_POST['username'])){
       $name = $_POST['username'];
       $result_set = mysqli_query($dbcon, "SELECT `Sno` FROM `users` WHERE `UserName` =
'$name'");
       $num_rows = mysqli_num_rows($result_set);
       if ($num_rows>0){
               echo "username is not available";
       }else{
               echo "username is available";
       }
```

```
}
else{
       header('Location: index.php');
}
<?php
       session_start();
       include 'dict.inc';
       function connection($dbcon){
               if(!$dbcon){
                       die("Connection Not Established . TRY AGAIN AFTER SOME TIME " .
mysqli_connect_error());
               }
       }
       function sendmail($email,$userName,$verifyhash){
               if (file_exists("phpmailer/PHPMailerAutoload.php")) {
                               require 'phpmailer/PHPMailerAutoload.php';
               }
               else {
                               echo "Please try later\n";
                               header('Location: sign.php');
               }
               $mail = new PHPMailer;
               $mail->isSMTP();
               $mail->Host = 'smtp.gmail.com';
               $mail->SMTPAuth = true;
               $mail->Username = "mnemonicscdu@gmail.com";
               $mail->Password = "simonmoss";
               $mail->SMTPSecure = 'ssl';
               $mail->Port = 465;
```

```
$mail->SMTPDebug = 2;
               $mail->setFrom('mnemonics@test.com', 'Mnemonics');
               $mail->addAddress($email,$userName);
               $mail->Subject = 'Mnemonics Signup verificaion';
               $mail->Body = '
               Hello '.$userName.',
               Your account has been created, you can login post verification.
               Please click the link below to activate your account:
       http://localhost/mnemonics_v2/finalmne/project/userverify.php?email='.$email.'&hash='.$veri
fyhash.'
               if ($mail->send()) {
                       $_SESSION['username']= $username;
                       $_SESSION['success'] = "Signed Up successfully. Verify EmailID to login.";
                       header('Location: Mnemonics/sign.php');
               } else {
                       $ SESSION['success'] = "Something went wrong! Please try again later.";
                       header('Location: Mnemonics/sign.php');
               }
       }
       function login($dbcon){
               $name = mysqli_real_escape_string($dbcon, $_POST['username']);
               $checkPassword = mysqli_real_escape_string($dbcon, $_POST['password']);
```

```
$validate = "SELECT * from users WHERE UserName = '$name' limit 1";
                       $exec = mysqli query($dbcon,$validate);
                       $array = mysqli fetch array($exec);
                       $currentPassword = $array['Password'];
                       $checkverified = $array['Email_verified'];
               if(crypt($checkPassword, $currentPassword) === $currentPassword &&
$checkverified=='1'){
                       $_SESSION["username"]= "$name";
                       header('Location: /project/Mnemonics/sign.php');
               }else if(crypt($checkPassword, $currentPassword) === $currentPassword &&
$array['Email_verified']=='0'){
                       $email = $array['Email'];
                       $verifyhash= $array['VerificationHash'];
                       sendmail($email,$name,$verifyhash);
                       $ SESSION['success'] = "Email not verified! Please verify to login.";
                       header('Location: Mnemonics/sign.php');
               }else if(crypt($checkPassword, $currentPassword) !== $currentPassword ){
                       $_SESSION['success'] = "Invalid Username or Password";
                       header('Location: Mnemonics/sign.php');
               }else {
                       header('Location: sign.php');
               }
       }
       function signUp($dbcon){
               $userName = mysqli real escape string($dbcon, $ POST['username']);
               $password = mysqli real escape string($dbcon, $ POST['password']);
               $studentid = mysqli_real_escape_string($dbcon, $_POST['studentid']);
```

```
$email = mysqli_real_escape_string($dbcon, $_POST['email']);
               $course = mysqli real escape string($dbcon, $ POST['course']);
               $verifyhash = md5($userName.rand(0,1000));
               $salt = '$2a$1$Ku2hb.'.rand(100000,999999).'.$';
               $pass = crypt($password, $salt);
               $sign = "INSERT INTO `users` (`StudentID`, `UserName`, `Password`, `Email`,
`Course`,`VerificationHash`,`Email verified`,`LinkExpired`) VALUES ('$studentid', '$userName', '$pass',
'$email', '$course', '$verifyhash','0','0')";
               $update = mysqli_query($dbcon,$sign);
                       if($update){
                               sendmail($email,$userName,$verifyhash);
                       }
                       else{
                               $_SESSION['signup'] .= "EmailID or StudentID already exists .";
                               header('Location: sign.php');
                       }
       }
       function new_pass($dbcon){
               $password = mysqli real escape string($dbcon, $ POST['new password']);
               $studentid = $_SESSION['recovery_studentid'];
               $update_password = mysqli_query($dbcon, "UPDATE `users` SET `Password` =
'$password' WHERE `StudentID` = '$studentid'");
               if($update password){
                       $_SESSION['success'] .= "Password Reset Successfull!";
                       header('Location: sign.php');
               }else {
                       $_SESSION['success'] .= "Something went wrong";
                       header('Location: sign.php');
```

```
}
}
function sendemailrecovery($email_id,$verifyhash,$studentid){
       if (file_exists("phpmailer/PHPMailerAutoload.php")) {
                       require 'phpmailer/PHPMailerAutoload.php';
       }
       else {
                       echo "Please try later\n";
                       header('Location: sign.php');
       $mail = new PHPMailer;
       $mail->isSMTP();
       $mail->Host = 'smtp.gmail.com';
       $mail->SMTPAuth = true;
       $mail->Username = "mnemonicscdu@gmail.com";
       $mail->Password = "simonmoss";
       $mail->SMTPSecure = 'ssl';
       $mail->Port = 465;
       $mail->SMTPDebug = 2;
       $mail->setFrom('mnemonics@test.com', 'Mnemonics');
       $mail->addAddress($email_id,");
       $mail->Subject = 'Mnemonics Password Recovery';
       $mail->Body = '
       Hello,
       You can reset your password by clicking the link below.
       Reset Password:
```

```
http://localhost/mnemonics_v2/finalmne/project/passwdrecovery.php?studentid='.$studentid.'
&hash='.$verifyhash.'
               if ($mail->send()){
                      $_SESSION['username']= $username;
                      $ SESSION['success'] = "Recovery Link has been sent.";
                      header('Location: Mnemonics/sign.php');
               } else {
                      $ SESSION['success'] = "Something went wrong! Please try again later.";
                      header('Location: Mnemonics/sign.php');
               }
       }
       function sendrecoverymail($dbcon,$email_id){
               $id = mysqli_query($dbcon,"SELECT StudentID,UserName from users where Email =
'$email id' limit 1");
               if (mysqli num rows($id)>0) {
                      $data = mysqli_fetch_array($id);
                      $studentid = $data[0];
                      $username = $data[1];
                      $verifyhash = md5($studentid.rand(0,1000));
                      $check_from_recovery = mysqli_query($dbcon,"SELECT RecoveryHash AND
RecoveryTimeStamp from recovery where StudentID = '$studentid' limit 1");//or
die(mysgli error($dbcon));
                      if (mysgli num rows($check from recovery)>0) {
                              $data_arr = mysqli_fetch_array($check_from_recovery);
                              $r hash = $data arr[0];
                              r = \frac{1}{3}
                              $overwrite_previous_recovery_hashes =
mysgli query($dbcon,"UPDATE recovery SET PreviousRecoveryHash='$r hash',
```

```
PreviousRecoveryTimeStamp= '$r hash ts' where StudentID = '$studentid' ") ;//or
die(mysqli_error($dbcon));
                               $update_new_recovery = mysqli_query($dbcon,"UPDATE recovery SET
RecoveryHash = '$verifyhash' and RecoveryTimeStamp =TIMESTAMP where StudentID = '$studentid' ")
;//or die(mysqli_error($dbcon));
                               if($update_new_recovery){
                                      sendemailrecovery($email id,$verifyhash,$studentid);
                               }
                       }else {
                               $fresh_insert = mysqli_query($dbcon,"INSERT INTO
recovery(StudentID,RecoveryHash)VALUES('$studentid','$verifyhash') ");//or die(mysqli_error($dbcon));
                               if($fresh insert){
                                      sendemailrecovery($email id,$verifyhash,$studentid);
                               }
                       }
               }
       }
       function forgot($dbcon){
               $email = mysqli_real_escape_string($dbcon, $_POST['recover_email']);
               $verify = mysqli_query($dbcon, "SELECT `Email` FROM `users` WHERE `Email` = '$email'
limit 1");// or die(mysqli_error($dbcon));
               if (!$verify) {
                       die('Error: '.mysqli_error($dbcon));
               }
               if(mysgli num rows($verify) >0){
                       $email tosend = mysqli fetch array($verify);
                       $email_id = $email_tosend[0];
                       sendrecoverymail($dbcon,$email id);
               }else{
                       die('Error');
               }
```

```
}
$count = count($_POST);
if (isset($_POST['username'])) {
       if($count==2){
                login($dbcon);
       }
       else{
                signUp($dbcon);
       }
}
if(isset($ POST['recover email'])){
       forgot($dbcon);
}
if(isset($_POST['new_password'])){
       new_pass($dbcon);
}else {
       header('Location: sign.php');
}
```

?>

References

- COLLIER, K. W. 2011. Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. *CIO Insight*, 30.
- Agile Alliance. (2017). What is Agile Software Development?. [online] Available at: https://www.agilealliance.org/agile101/ [Accessed 29 Sep. 2017].
- LARMAN, C. 2004. *Agile and iterative development : a manager's guide,* Boston ; London, Boston ; London : Addison-Wesley.
- Cockburn, A., 2002. Agile software development (Vol. 177). Boston: Addison-Wesley.
- Sommerville, I. 2011, *Software engineering*, 9th, International edn, Pearson, Boston.

- Brighthub Project Management. 2016. What do You do with Change Requests in Software Project Development: the Handling Process. [ONLINE] Available at: http://www.brighthubpm.com/change-management/5737-change-management-in-software-development-projects/.
- Agile Process Image 1 reference: Ghosh, A. (2017). Agile Application and Cloud Computing. [online]
 The Customize Windows. Available at: https://thecustomizewindows.com/2014/11/agile-application-cloud-computing/ [Accessed 1 Oct. 2017].
- Agile Process Image 2 reference: 2.Pivotaltracker.com. (2017). *Agile Project Management*. [online] Available at: https://www.pivotaltracker.com/agile/what-is-agile-project-management [Accessed 1 Oct. 2017].