

SOFTWARE ENGINEERING LEC/LAB
MIDTERM EXAM
“REGISTER, LOGIN, LOGOUT AND WHO DID IT?”

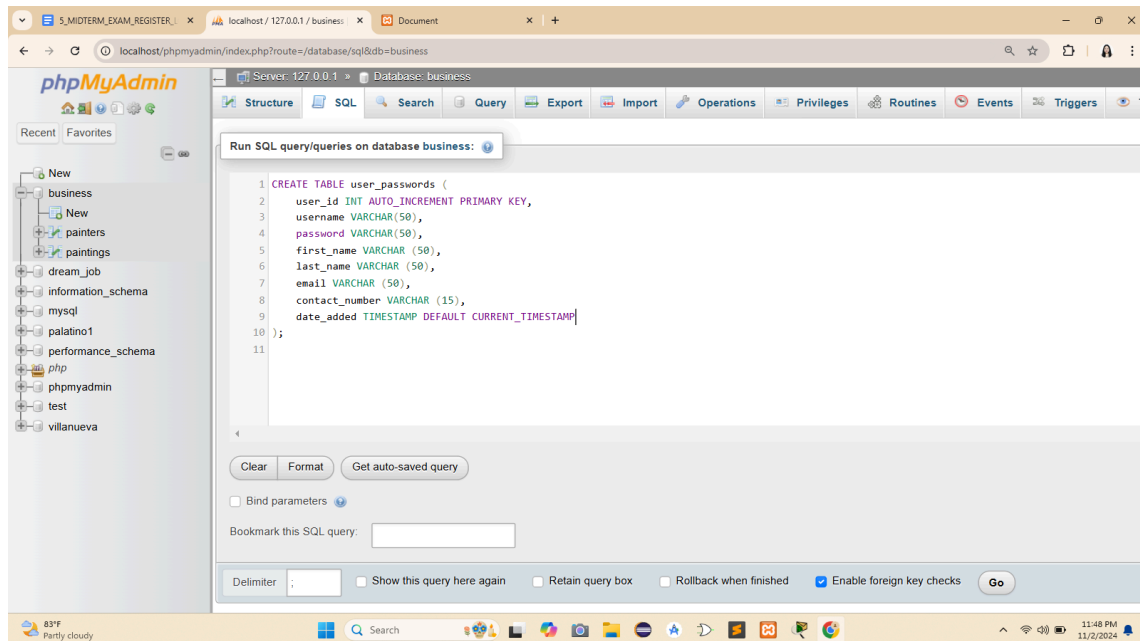
NAME: Villanueva, Marithei H.
SECTION: UCOS 3-1

INSTRUCTIONS:

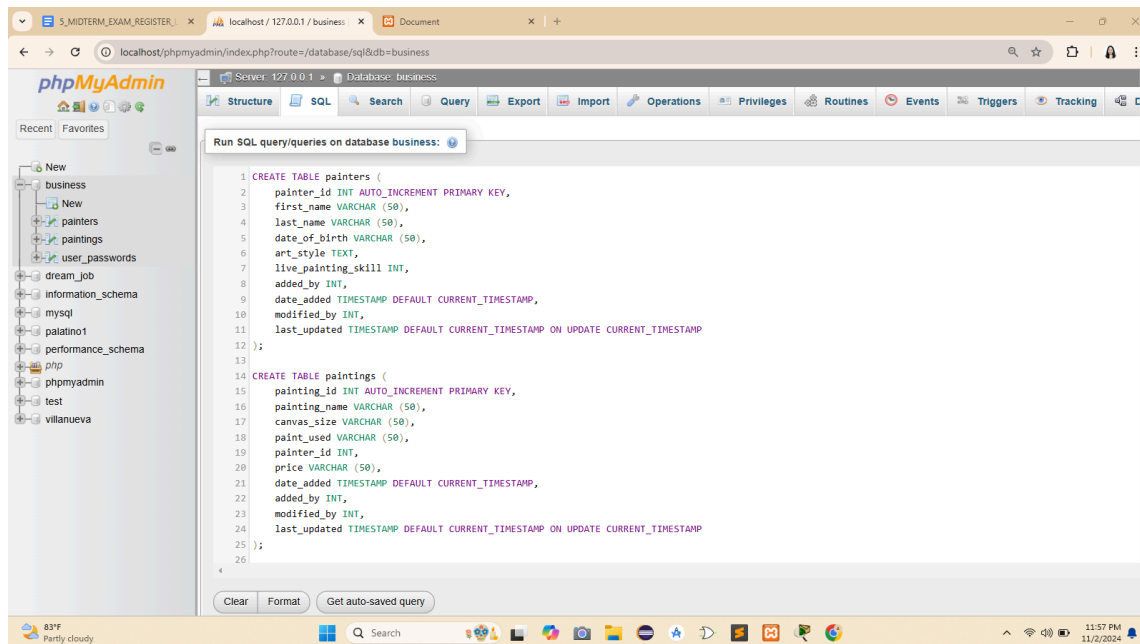
- Go back to your activity, **YOUR BUSINESS IDEAS AS SEEN WITH PDO**, what you need to do for your midterm exam is to add a **USER ENTITY** to the system you created. The user should have columns storing his personal information. For example, a user may have the first_name, last_name, address, age, etc columns for the users table. No limits on the columns that need to be filled.
- The user should be able to **LOGIN, REGISTER, AND LOGOUT** from the system and keep in mind that we should be able to keep track of that certain user who made an insertion, deletion, and updating of a record stored in the database.
- For example if your chosen business is a web development, we should be able to **IDENTIFY THAT USER** who **CREATED** a project, **UPDATED** a project, and the **LAST TIMESTAMP** when the project was updated. So for your tables, you must add additional columns, added_by and last_updated.
- Please keep in mind that a user should **NOT BE ABLE** to access the homepage without logging in. If a user goes inside index.php and he's not logged in, he should be redirected to the login page.
- Don't forget to give what's being asked from the items below.

1. How did you implement the **User entity** to the system you created? Give **the complete steps below** with some screenshots of the steps you took.

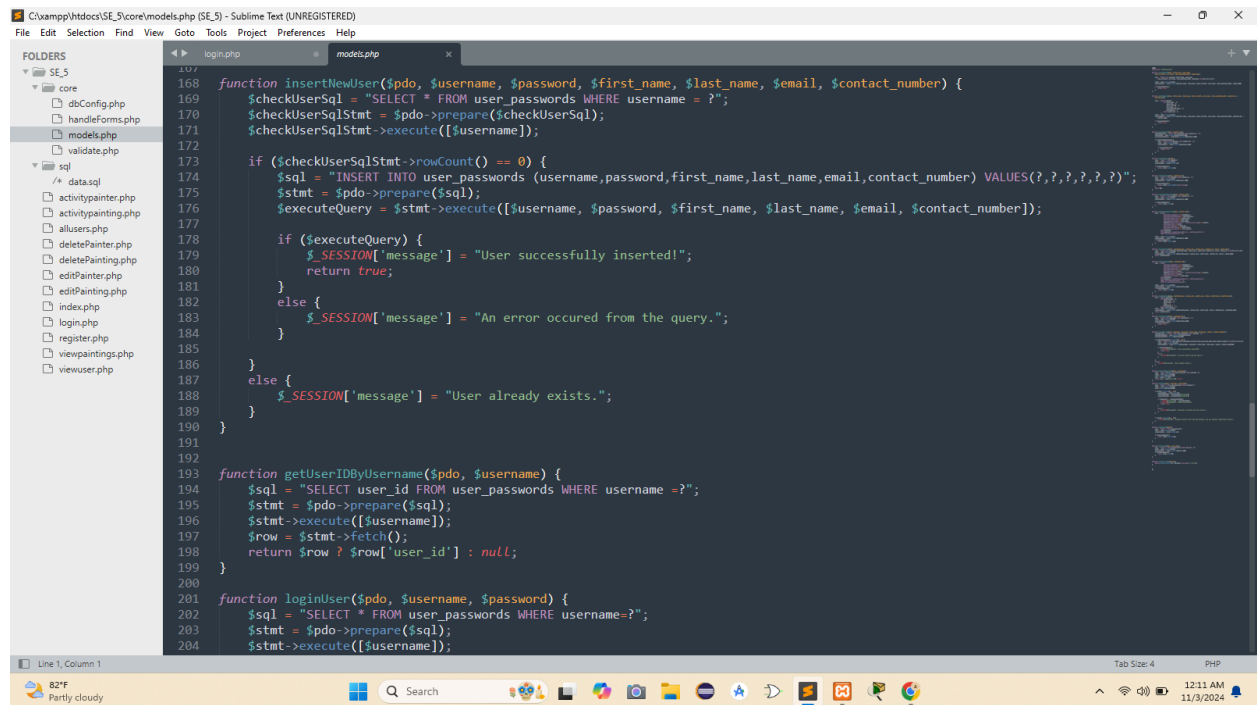
STEP 1: I created a new table called "user_passwords".



STEP 2: I updated the existing tables to include "added_by" to track the user who added the painter or painting in the system, "modified_by" to indicate who edited it, and "last_updated" to show the most recent timestamps of modifications.

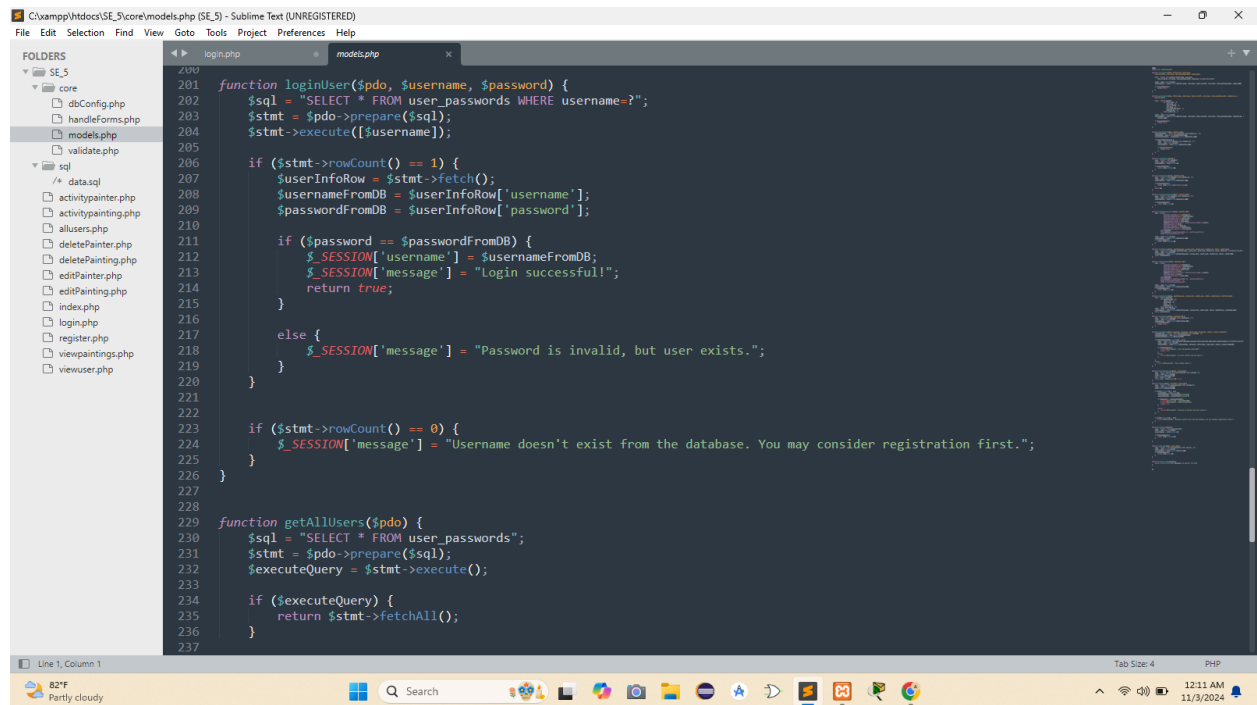


STEP 3: I declared new functions in models.php to handle user registration and login.



The screenshot shows the Sublime Text editor with the file `models.php` open. The left sidebar displays the project structure for `SE_5`, including folders like `core`, `sql`, and `data.sql`. The main editor area contains the following PHP code:

```
168 function insertNewUser($pdo, $username, $password, $first_name, $last_name, $email, $contact_number) {
169     $checkUserSql = "SELECT * FROM user_passwords WHERE username = ?";
170     $checkUserSqlStmt = $pdo->prepare($checkUserSql);
171     $checkUserSqlStmt->execute([$username]);
172
173     if ($checkUserSqlStmt->rowCount() == 0) {
174         $sql = "INSERT INTO user_passwords (username,password,first_name,last_name,email,contact_number) VALUES(?,?,?,?,?,?)";
175         $stmt = $pdo->prepare($sql);
176         $executeQuery = $stmt->execute([$username, $password, $first_name, $last_name, $email, $contact_number]);
177
178         if ($executeQuery) {
179             $_SESSION['message'] = "User successfully inserted!";
180             return true;
181         } else {
182             $_SESSION['message'] = "An error occurred from the query.";
183         }
184     } else {
185         $_SESSION['message'] = "User already exists.";
186     }
187 }
188
189 function getUserIDByUsername($pdo, $username) {
190     $sql = "SELECT user_id FROM user_passwords WHERE username =?";
191     $stmt = $pdo->prepare($sql);
192     $stmt->execute([$username]);
193     $row = $stmt->fetch();
194     return $row ? $row['user_id'] : null;
195 }
196
197 function loginUser($pdo, $username, $password) {
198     $sql = "SELECT * FROM user_passwords WHERE username=?";
199     $stmt = $pdo->prepare($sql);
200     $stmt->execute([$username]);
```



The screenshot shows the Sublime Text editor with the file `models.php` open. The left sidebar displays the project structure for `SE_5`. The main editor area contains the following PHP code:

```
201 function loginUser($pdo, $username, $password) {
202     $sql = "SELECT * FROM user_passwords WHERE username=?";
203     $stmt = $pdo->prepare($sql);
204     $stmt->execute([$username]);
205
206     if ($stmt->rowCount() == 1) {
207         $userInfoRow = $stmt->fetch();
208         $usernameFromDB = $userInfoRow['username'];
209         $passwordFromDB = $userInfoRow['password'];
210
211         if ($password == $passwordFromDB) {
212             $_SESSION['username'] = $usernameFromDB;
213             $_SESSION['message'] = "Login successful!";
214             return true;
215         } else {
216             $_SESSION['message'] = "Password is invalid, but user exists.";
217         }
218     }
219
220     if ($stmt->rowCount() == 0) {
221         $_SESSION['message'] = "Username doesn't exist from the database. You may consider registration first.";
222     }
223 }
224
225 function getAllUsers($pdo) {
226     $sql = "SELECT * FROM user_passwords";
227     $stmt = $pdo->prepare($sql);
228     $executeQuery = $stmt->execute();
229
230     if ($executeQuery) {
231         return $stmt->fetchAll();
232     }
233 }
234
235
236
237
```

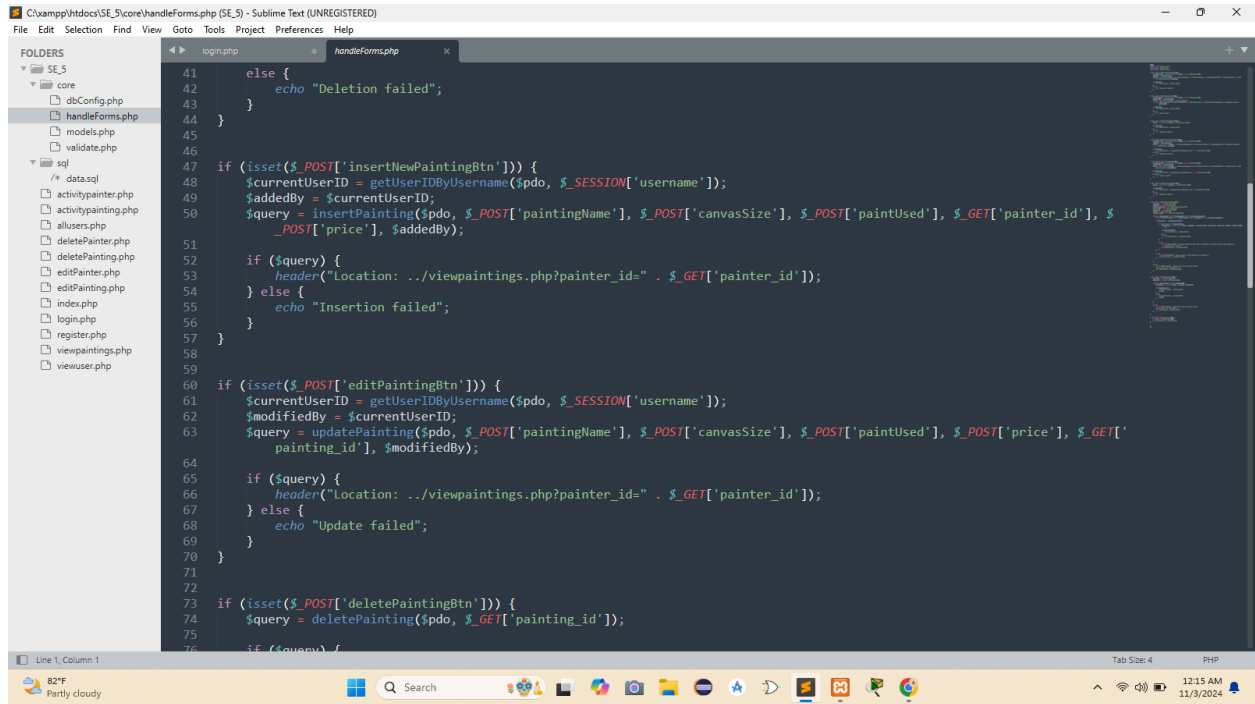
```
login.php
models.php

221
222
223 if ($stmt->rowCount() == 0) {
224     $_SESSION['message'] = "Username doesn't exist from the database. You may consider registration first.";
225 }
226 }
227
228
229 function getAllUsers($pdo) {
230     $sql = "SELECT * FROM user_passwords";
231     $stmt = $pdo->prepare($sql);
232     $executeQuery = $stmt->execute();
233
234     if ($executeQuery) {
235         return $stmt->fetchAll();
236     }
237 }
238
239
240
241 function getUserByID($pdo, $user_id) {
242     $sql = "SELECT * FROM user_passwords WHERE user_id = ?";
243     $stmt = $pdo->prepare($sql);
244     $executeQuery = $stmt->execute([$user_id]);
245     if ($executeQuery) {
246         return $stmt->fetch();
247     }
248 }
249
250
251 function sanitize_output($data) {
252     return htmlspecialchars(trim($data), ENT_QUOTES, 'UTF-8');
253 }
254
255
256 ?>
```

STEP 4: I modified my handleForms to include the “added_by” and “modified_by” attributes from my previous implementation.

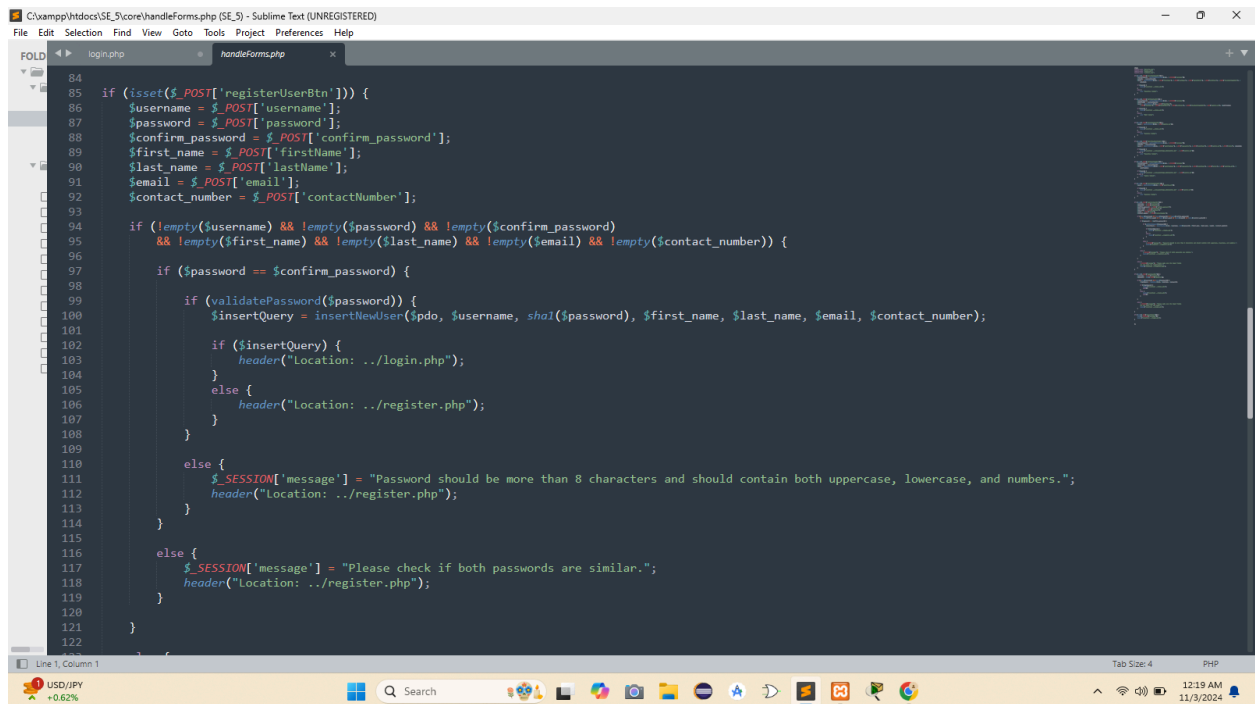
```
login.php
handleForms.php

1 <?php
2 require_once 'dbConfig.php';
3 require_once 'models.php';
4 require_once 'validate.php';
5
6 if (isset($_POST['insertPainterBtn'])) {
7     $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
8     $addedBy = $currentUserID;
9     $query = insertPainter($pdo, $_POST['firstName'], $_POST['lastName'], $_POST['dateOfBirth'], $_POST['artStyle'], $_POST['livePaintingSkill'], $addedBy);
10
11     if ($query) {
12         header("Location: ../index.php");
13     }
14     else {
15         echo "Insertion failed";
16     }
17 }
18
19
20 if (isset($_POST['editPainterBtn'])) {
21     $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
22     $modifiedBy = $currentUserID;
23     $query = updatePainter($pdo, $_POST['firstName'],
24         $_POST['lastName'], $_POST['dateOfBirth'], $_POST['artStyle'], $_POST['livePaintingSkill'], $_GET['painter_id'], $modifiedBy);
25
26     if ($query) {
27         header("Location: ../index.php");
28     }
29     else {
30         echo "Edit failed";
31     }
32 }
33
34
35 if (isset($_POST['deletePainterBtn'])) {
36     $query = deletePainter($pdo, $_GET['painter_id']);
```



```
41     else {
42         echo "Deletion failed";
43     }
44 }
45
46
47 if (isset($_POST['insertNewPaintingBtn'])) {
48     $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
49     $addedBy = $currentUserID;
50     $query = insertPainting($pdo, $_POST['paintingName'], $_POST['canvasSize'], $_POST['paintUsed'], $_GET['painter_id'], $_POST['price'], $addedBy);
51
52     if ($query) {
53         header("Location: ../viewpaintings.php?painter_id=" . $_GET['painter_id']);
54     } else {
55         echo "Insertion failed";
56     }
57 }
58
59
60 if (isset($_POST['editPaintingBtn'])) {
61     $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
62     $modifiedBy = $currentUserID;
63     $query = updatePainting($pdo, $_POST['paintingName'], $_POST['canvasSize'], $_POST['paintUsed'], $_POST['price'], $_GET['painting_id'], $modifiedBy);
64
65     if ($query) {
66         header("Location: ../viewpaintings.php?painter_id=" . $_GET['painter_id']);
67     } else {
68         echo "Update failed";
69     }
70 }
71
72
73 if (isset($_POST['deletePaintingBtn'])) {
74     $query = deletePainting($pdo, $_GET['painting_id']);
75
76     if ($query) {
```

STEP 5: I also added the registration and login buttons in my handleForms file.



```
84
85 if (isset($_POST['registerUserBtn'])) {
86     $username = $_POST['username'];
87     $password = $_POST['password'];
88     $confirm_password = $_POST['confirm_password'];
89     $first_name = $_POST['firstName'];
90     $last_name = $_POST['lastName'];
91     $email = $_POST['email'];
92     $contact_number = $_POST['contactNumber'];
93
94     if (empty($username) && empty($password) && empty($confirm_password)
95         && empty($first_name) && empty($last_name) && empty($email) && empty($contact_number)) {
96
97         if ($password == $confirm_password) {
98
99             if (validatePassword($password)) {
100                 $insertQuery = insertNewUser($pdo, $username, sha1($password), $first_name, $last_name, $email, $contact_number);
101
102                 if ($insertQuery) {
103                     header("Location: ../login.php");
104                 } else {
105                     header("Location: ../register.php");
106                 }
107             }
108         } else {
109             $_SESSION['message'] = "Password should be more than 8 characters and should contain both uppercase, lowercase, and numbers.";
110             header("Location: ../register.php");
111         }
112     } else {
113         $_SESSION['message'] = "Please check if both passwords are similar.";
114         header("Location: ../register.php");
115     }
116 }
117
118
119
120
121
122
```

```
C:\xampp\htdocs\SE_5\core\handleForms.php (SE_5) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

login.php handleForms.php
127 }
128 }
129
130
131 if (isset($_POST['loginUserBtn'])) {
132     $username = $_POST['username'];
133     $password = sha1($_POST['password']);
134
135     if (!empty($username) && !empty($password)) {
136         $loginQuery = loginUser($pdo, $username, $password);
137
138         if ($loginQuery) {
139             header("Location: ../index.php");
140             exit();
141         } else {
142             header("Location: ../login.php");
143             exit();
144         }
145     }
146 }
147
148
149 else {
150     $_SESSION['message'] = "Please make sure the input fields
151 are not empty for the login.";
152     header("Location: ../login.php");
153 }
154 }
155 }
156
157 if (isset($_GET['logoutUser'])) {
158     unset($_SESSION['username']);
159     header("Location: ../login.php");
160 }
161
162
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }
```

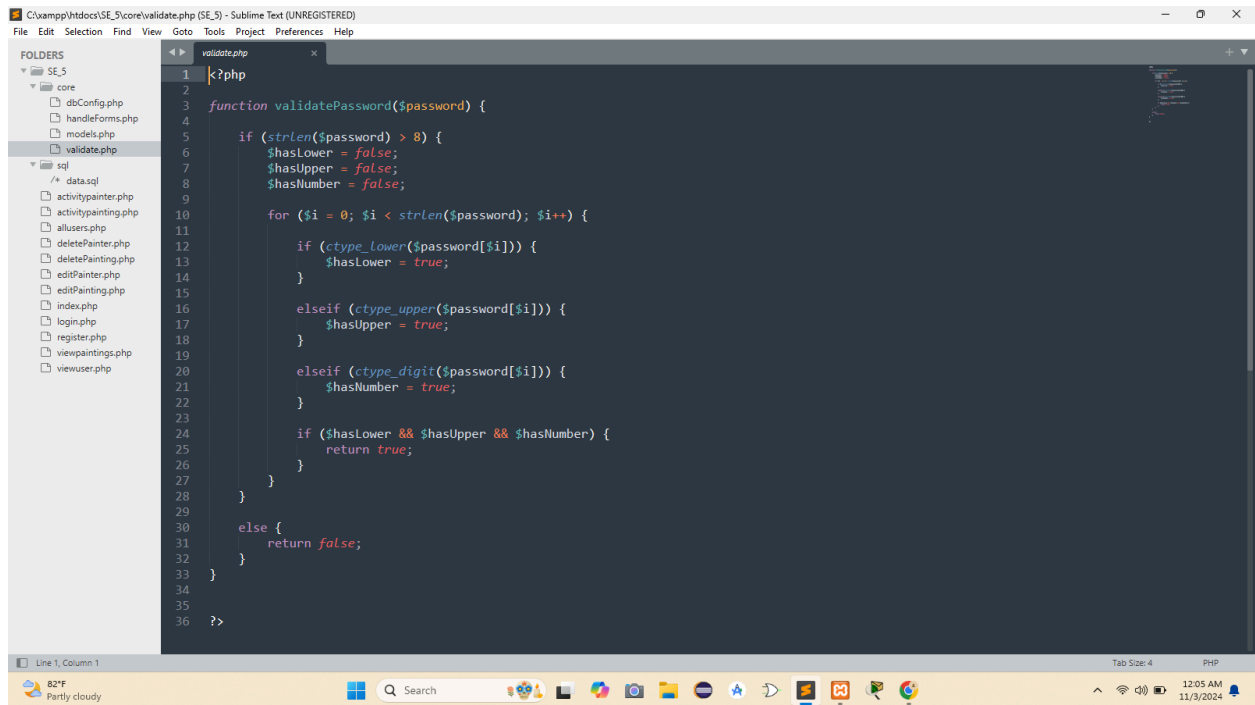
STEP 6: I created a PHP file named “register” for user registration.

```
C:\xampp\htdocs\SE_5\register.php (SE_5) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
SE_5
core
dbConfig.php
handleForms.php
models.php
validate.php
sql
data.sql
activityPainter.php
activityPainting.php
allUsers.php
deletePainter.php
deletePainting.php
editPainter.php
editPainting.php
index.php
login.php
register.php
viewPaintings.php
viewUser.php

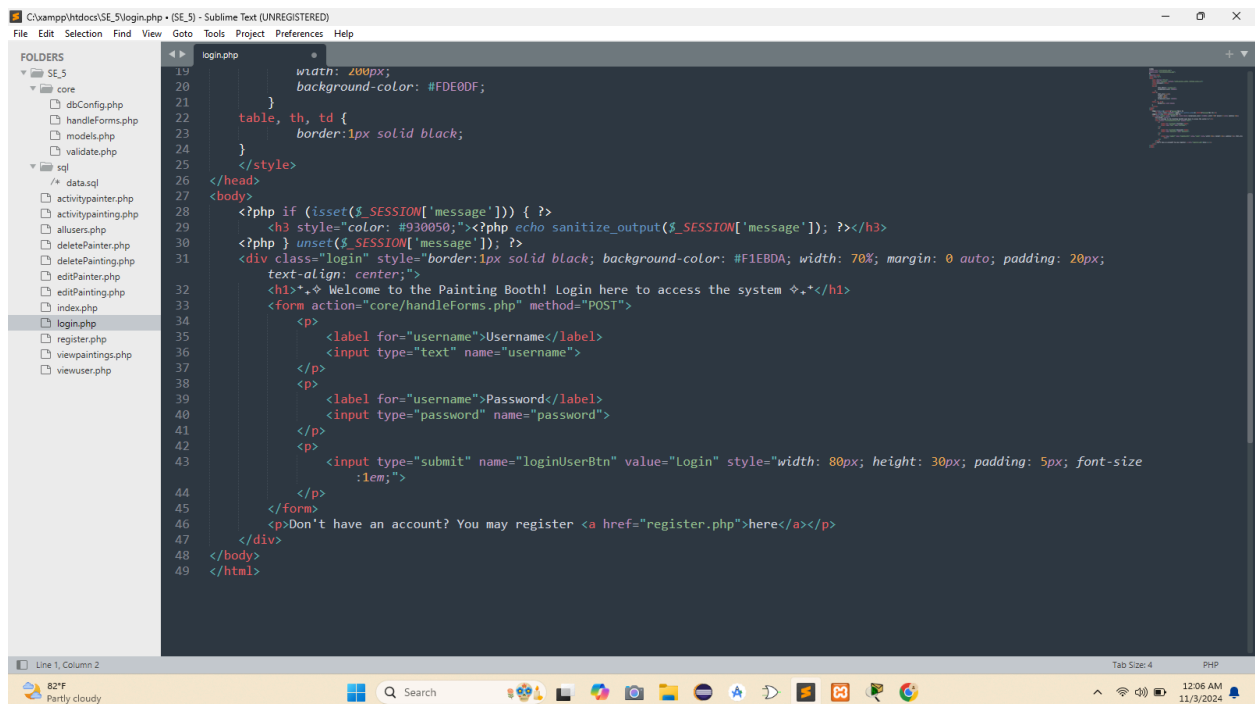
27 <body>
28
29 <a href="login.php">Login page</a>
30 <div class="register" style="border-style: solid; width: 30%; height: 570px; margin-top: 15px; padding-left: 20px; background-color: #F1E8DA
31 <h1>Kindly register a user here </h1>
32 <?php if (isset($_SESSION['message'])) { ?>
33 <h3 style="color: #930050;"><?php echo $_SESSION['message']; ?></h3>
34 <?php } unset($_SESSION['message']); ?>
35 <form action="core/handleForms.php" method="POST">
36
37 <label for="username">Username</label>
38 <input type="text" name="username" required>
39
40 <label for="password">Password</label>
41 <input type="password" name="password" required>
42
43 <label for="confirmPassword">Confirm Password</label>
44 <input type="password" name="confirm_password">
45
46 <label for="firstName">First Name</label>
47 <input type="text" name="firstName" required>
48
49 <label for="lastName">Last Name</label>
50 <input type="text" name="lastName" required>
51
52 <label for="email">Email Address</label>
53 <input type="email" name="email" required>
54
55 <label for="contactNumber">Contact Number</label>
56 <input type="number" name="contactNumber" maxlength="11" required>
57
58 <input type="submit" name="registerUserBtn" value="Register" style="width: 80px; height: 30px; padding: 5px; font-size: 1em;">
59
60 </form>
61 </div>
62 </body>
```

STEP 7: I created a PHP file named “validate” to check if the password is weak or strong.



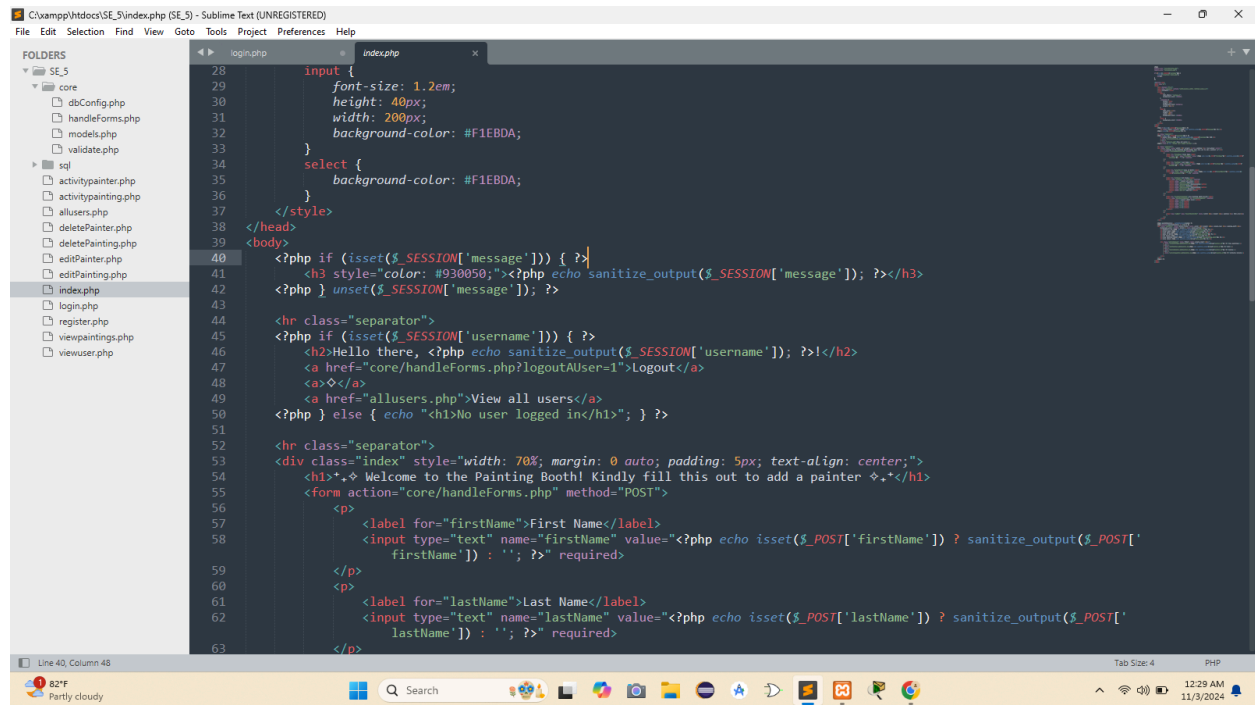
```
1 <?php
2
3 function validatePassword($password) {
4
5     if (strlen($password) > 8) {
6         $hasLower = false;
7         $hasUpper = false;
8         $hasNumber = false;
9
10        for ($i = 0; $i < strlen($password); $i++) {
11
12            if (ctype_lower($password[$i])) {
13                $hasLower = true;
14            }
15
16            elseif (ctype_upper($password[$i])) {
17                $hasUpper = true;
18            }
19
20            elseif (ctype_digit($password[$i])) {
21                $hasNumber = true;
22            }
23
24            if ($hasLower && $hasUpper && $hasNumber) {
25                return true;
26            }
27        }
28    }
29
30    else {
31        return false;
32    }
33 }
34
35 ?>
```

STEP 8: I created another PHP file named “login” for users to access the system.



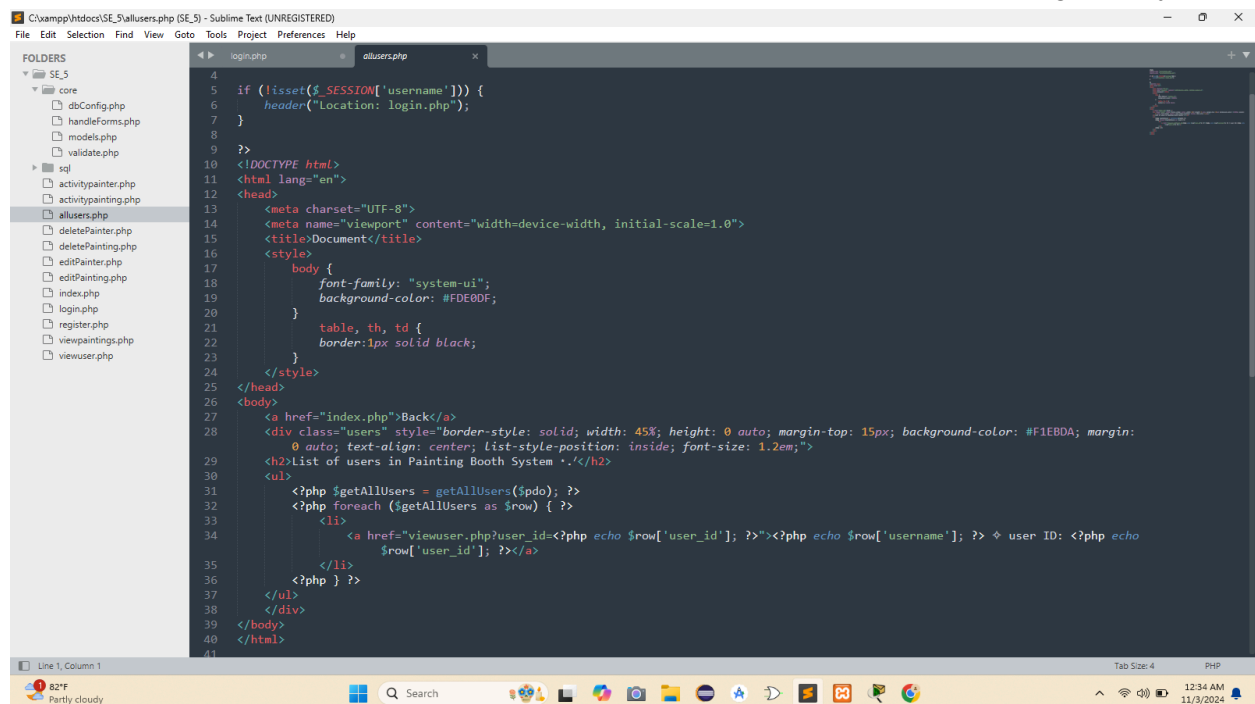
```
19 width: 200px;
20 background-color: #FDE0DF;
21 }
22 table, th, td {
23     border: 1px solid black;
24 }
25 </style>
26 </head>
27 <body>
28     <?php if (isset($_SESSION['message'])) { ?>
29         <h3 style="color: #930050;"><?php echo sanitize_output($_SESSION['message']); ?></h3>
30     <?php } unset($_SESSION['message']); ?>
31     <div class="login" style="border: 1px solid black; background-color: #F1E8DA; width: 70%; margin: 0 auto; padding: 20px; text-align: center;">
32         <h1>* Welcome to the Painting Booth! Login here to access the system *</h1>
33         <form action="core/handleForms.php" method="POST">
34             <p>
35                 <label for="username">Username</label>
36                 <input type="text" name="username">
37             </p>
38             <p>
39                 <label for="password">Password</label>
40                 <input type="password" name="password">
41             </p>
42             <p>
43                 <input type="submit" name="loginUserBtn" value="Login" style="width: 80px; height: 30px; padding: 5px; font-size: 1em;">
44             </p>
45         </form>
46         <p>Don't have an account? You may register <a href="register.php">here</a></p>
47     </div>
48 </body>
49 </html>
```

STEP 9: I inserted code in my index file to check for successful logins and display the session message.



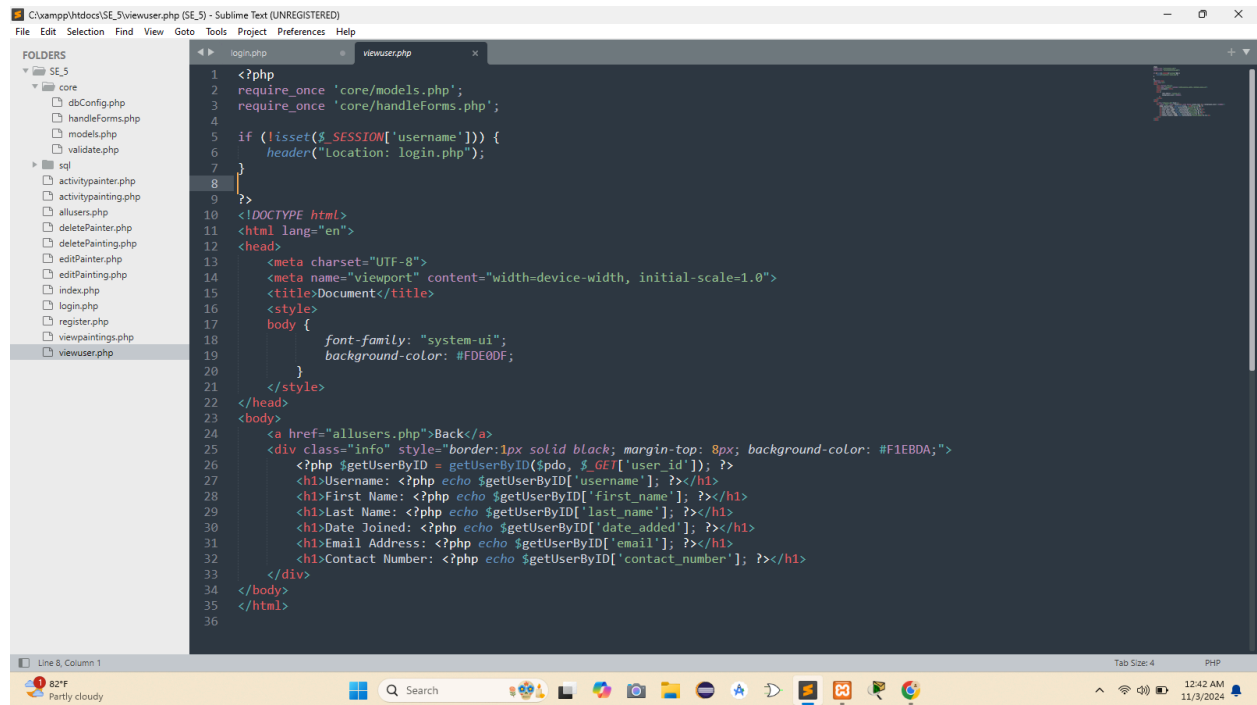
```
28 login.php
29
30     input {
31         font-size: 1.2em;
32         height: 40px;
33         width: 200px;
34         background-color: #F1EBDA;
35     }
36     select {
37         background-color: #F1EBDA;
38     }
39 </style>
40 </head>
41 <body>
42 <?php if (isset($_SESSION['message'])) { ?>
43 <h3 style="color: #930050;"><?php echo sanitize_output($_SESSION['message']); ?></h3>
44 <?php unset($_SESSION['message']); ?>
45
46 <hr class="separator">
47 <?php if (isset($_SESSION['username'])) { ?>
48 <h2>Hello there, <?php echo sanitize_output($_SESSION['username']); ?>!</h2>
49 <a href="core/handleForms.php?logoutUser=1">Logout</a>
50 <a href="allusers.php">View all users</a>
51 <?php } else { echo "<h1>No user logged in</h1>"; } ?>
52
53 <hr class="separator">
54 <div class="index" style="width: 70%; margin: 0 auto; padding: 5px; text-align: center;">
55 <h1>+ Welcome to the Painting Booth! Kindly fill this out to add a painter +</h1>
56 <form action="core/handleForms.php" method="POST">
57 <p>
58 <label for="firstName">First Name</label>
59 <input type="text" name="firstName" value="<?php echo isset($_POST['firstName']) ? sanitize_output($_POST['firstName']) : '' ; ?> required>
60
61 <p>
62 <label for="lastName">Last Name</label>
63 <input type="text" name="lastName" value="<?php echo isset($_POST['lastName']) ? sanitize_output($_POST['lastName']) : '' ; ?> required>
64
65 </p>
66 </div>
67 </body>
68 </html>
```

STEP 10: On the index page, there is an option to view users, so I created another PHP file named "all_users" that stores the list of users authorized to access and manage the system.



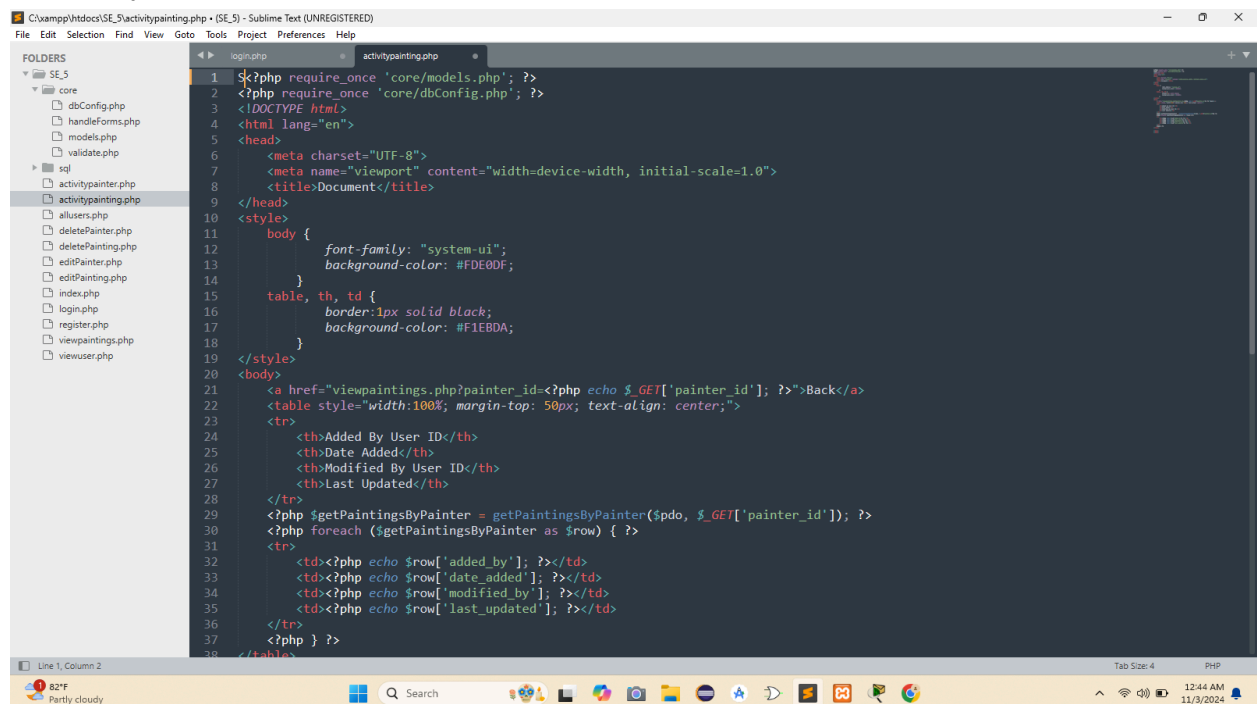
```
4
5 if (isset($_SESSION['username'])) {
6     header("Location: login.php");
7 }
8
9 ?>
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13 <meta charset="UTF-8">
14 <meta name="viewport" content="width=device-width, initial-scale=1.0">
15 <title>Document</title>
16 <style>
17     body {
18         font-family: "system-ui";
19         background-color: #FDE0DF;
20     }
21     table, th, td {
22         border: 1px solid black;
23     }
24 </style>
25 </head>
26 <body>
27 <a href="index.php">Back</a>
28 <div class="users" style="border-style: solid; width: 45%; height: 0 auto; margin-top: 15px; background-color: #F1EBDA; margin: 0 auto; text-align: center; list-style-position: inside; font-size: 1.2em;">
29 <h2>List of users in Painting Booth System </h2>
30 <ul>
31 <?php $getAllUsers = getAllUsers($pdo); ?>
32 <?php foreach ($getAllUsers as $row) { ?>
33 <li>
34 <a href="viewuser.php?user_id=<?php echo $row['user_id']; ?>"><?php echo $row['username']; ?> <span> user ID: <?php echo $row['user_id']; ?></a>
35 </li>
36 <?php } ?>
37 </ul>
38 </div>
39 </body>
40 </html>
```


STEP 11: Each user in the "allusers" file is clickable, directing to the "viewuser" file I created to show the user's information.



```
1 <?php
2 require_once 'core/models.php';
3 require_once 'core/handleForms.php';
4
5 if (!isset($_SESSION['username'])) {
6     header("Location: login.php");
7 }
8
9
10 <?php
11 <!DOCTYPE html>
12 <html lang="en">
13 <head>
14     <meta charset="UTF-8">
15     <meta name="viewport" content="width=device-width, initial-scale=1.0">
16     <title>Document</title>
17     <style>
18         body {
19             font-family: "system-ui";
20             background-color: #FDE0DF;
21         }
22     </style>
23 </head>
24 <body>
25     <a href="allusers.php">Back</a>
26     <div class="info" style="border:1px solid black; margin-top: 8px; background-color: #F1EBDA;">
27         <?php $getUserByID = getUserByID($pdo, $_GET['user_id']); ?>
28         <h1>Username: <?php echo $getUserByID['username']; ?></h1>
29         <h1>First Name: <?php echo $getUserByID['first_name']; ?></h1>
30         <h1>Last Name: <?php echo $getUserByID['last_name']; ?></h1>
31         <h1>Date Joined: <?php echo $getUserByID['date_added']; ?></h1>
32         <h1>Email Address: <?php echo $getUserByID['email']; ?></h1>
33         <h1>Contact Number: <?php echo $getUserByID['contact_number']; ?></h1>
34     </div>
35 </body>
36 </html>
```

STEP 12: I created two new files named "activitypainting" and "activitypainter" to view the activity details that occurred for each specific record.



```
1 <?php require_once 'core/models.php'; ?>
2 <?php require_once 'core/dbConfig.php'; ?>
3
4 <!DOCTYPE html>
5 <html lang="en">
6 <head>
7     <meta charset="UTF-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <title>Document</title>
10 </head>
11 <style>
12     body {
13         font-family: "system-ui";
14         background-color: #FDE0DF;
15     }
16     table, th, td {
17         border:1px solid black;
18         background-color: #F1EBDA;
19     }
20 </style>
21 <body>
22     <a href="viewpaintings.php?painter_id=<?php echo $_GET['painter_id']; ?>">Back</a>
23     <table style="width:100%; margin-top: 50px; text-align: center;">
24         <tr>
25             <th>Added By User ID</th>
26             <th>Date Added</th>
27             <th>Modified By User ID</th>
28             <th>Last Updated</th>
29         </tr>
30         <?php $getPaintingsByPainter = getPaintingsByPainter($pdo, $_GET['painter_id']); ?>
31         <?php foreach ($getPaintingsByPainter as $row) { ?>
32             <tr>
33                 <td><?php echo $row['added_by']; ?></td>
34                 <td><?php echo $row['date_added']; ?></td>
35                 <td><?php echo $row['modified_by']; ?></td>
36                 <td><?php echo $row['last_updated']; ?></td>
37             </tr>
38         </tr>
39     </table>
```

```
1 <?php require_once 'core/models.php'; ?>
2 <?php require_once 'core/dbConfig.php'; ?>
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6 <meta charset="UTF-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <title>Document</title>
9 </head>
10 <style>
11     body {
12         font-family: "system-ui";
13         background-color: #FDE0DF;
14     }
15     table, th, td {
16         border: 1px solid black;
17         background-color: #F1E8DA;
18     }
19 </style>
20 <body>
21 <a href="index.php">Back</a>
22 <table style="width:100%; margin-top: 50px; text-align: center;">
23 <tr>
24 <th>Added By User ID</th>
25 <th>Date Added</th>
26 <th>Modified By User ID</th>
27 <th>Last Updated</th>
28 </tr>
29 <?php $getPainterByID = getPainterByID($pdo, $_GET['painter_id']); ?>
30 <?php foreach ($getPainterByID as $row) { ?>
31 <tr>
32 <td><?php echo $row['added_by']; ?></td>
33 <td><?php echo $row['date_added']; ?></td>
34 <td><?php echo $row['modified_by']; ?></td>
35 <td><?php echo $row['last_updated']; ?></td>
36 </tr>
37 </?php ?>
38 </table>
```

STEP 13: Lastly, I created another function to handle empty rows, as some records have not yet been edited, resulting in the "modified_by" field being empty.

```
158 function getPainterByID($pdo, $painter_id) {
159     $sql = "SELECT * FROM painters WHERE painter_id = ?";
160     $stmt = $pdo->prepare($sql);
161     $executeQuery = $stmt->execute([$painter_id]);
162
163     if ($executeQuery) {
164         return $stmt->fetchAll(PDO::FETCH_ASSOC);
165     }
166     return [];
167 }
168
169 function getPaintingByID($pdo, $painting_id) {
170     $sql = "SELECT
171         paintings.painting_id AS painting_id,
172         paintings.painting_name AS painting_name,
173         paintings.canvas_size AS canvas_size,
174         paintings.paint_used AS paint_used,
175         CONCAT(painters.first_name, ' ', painters.last_name) AS painter,
176         paintings.price AS price,
177         paintings.date_added AS date_added,
178         paintings.added_by AS added_by,
179         paintings.modified_by AS modified_by,
180         paintings.last_updated AS last_updated
181     FROM paintings
182     JOIN painters ON paintings.painter_id = painters.painter_id
183     WHERE paintings.painting_id = ?
184     GROUP BY paintings.painting_name";
185
186     $stmt = $pdo->prepare($sql);
187     $executeQuery = $stmt->execute([$painting_id]);
188     if ($executeQuery) {
189         return $stmt->fetchAll(PDO::FETCH_ASSOC);
190     }
191     return [];
192 }
193
194 }
```

2. Show a screen-recorded demonstration of the system you created and paste its **Google Drive link** below. This should include **registration**, **logging in**, **inserting**, **updating**, **deleting** records from the system and **logging out**. Don't forget to set its visibility to **anyone with link**.

<https://drive.google.com/file/d/1khGcwrad-bKHJBVkHenZLeTpC2vS-cYP/view?usp=sharing>

3. Paste **GitHub** link of project below.

https://github.com/mystheirious/business_login/tree/main/SE_5