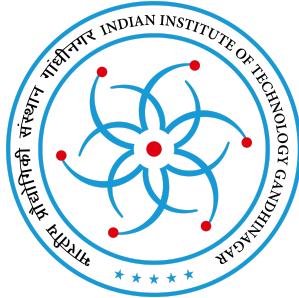


Indian Institute of Technology Gandhinagar



CS432: Databases

Semester II, AY 2023-24

ASSIGNMENT 3

GROUP - DataWeavers

TOPIC: oCEO Activities Portal Management

Group Members

Aditya Gupte	21110012
Balgopal Moharana	21110040
Darshi Doshi	21110050
Shah Faisal Khan	21110156
Pranjal	21110160
Rohit Raj	21110179
Tanish Phopalkar	21110221
Animesh Tumne	21110227

Under the guidance of
Prof. Mayank Singh

Tasks of G1:

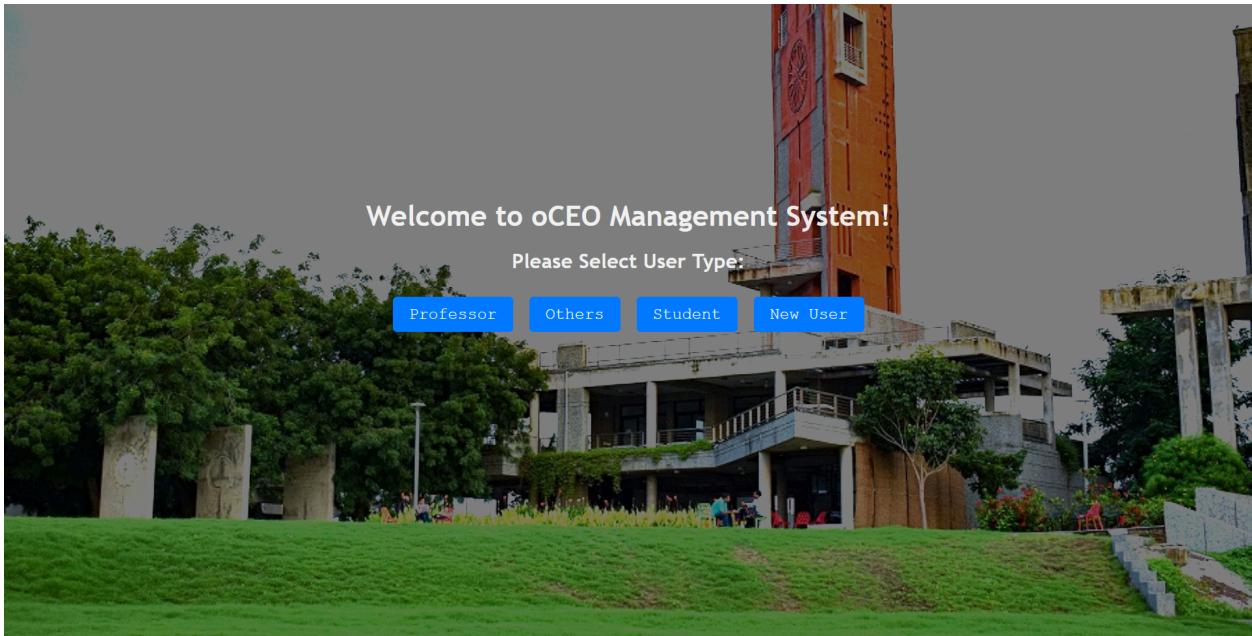


Fig. HomePage

A screenshot of a "New User Registration" form. The form has a white background with black text labels and input fields. It includes fields for Roll Number/Employee ID, First Name, Middle Name, Last Name, Email ID, Password, and Confirm Password. Below these is a "User Type:" section with three radio button options: "Other", "Professor", and "Student". At the bottom right is a green "Register" button.

Fig. After Clicking New User on HomePage

A screenshot of a web browser window titled "Student X". The address bar shows the URL "127.0.0.1:5000/login/student". The main content area displays a "Student Login" form. It includes fields for "Email ID" and "Password", both represented by empty input boxes. Below these fields is a green "Submit" button.

Fig. After Clicking Student on the Homepage

A screenshot of a web browser window titled "Others X". The address bar shows the URL "127.0.0.1:5000/login/others". The main content area displays an "Enter Login Details" form. It includes fields for "Email ID" and "Password", both represented by empty input boxes. Below these fields is a green "Submit" button. At the bottom, there is a "User Type:" label followed by four radio buttons: "Admin", "Dean", "SA JS", and "oCEO Coordinator".

Fig. After Clicking Others on the Homepage

A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5000/login/professor`. The main content area displays a login form titled "Professor Login". It contains two input fields: "Email ID:" and "Password:", both represented by white input boxes with black outlines. Below the password field is a green rectangular button labeled "Submit". The background of the form is light gray.

Fig. After Clicking Professor on HomePage

STUDENT LOGIN

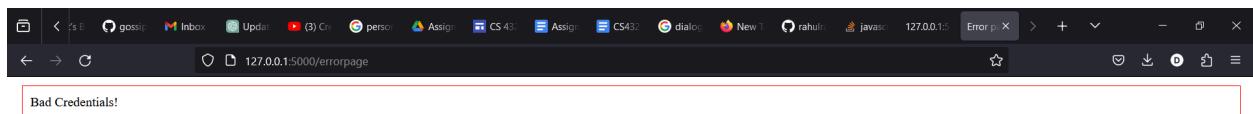


Fig. Unsuccessful Login

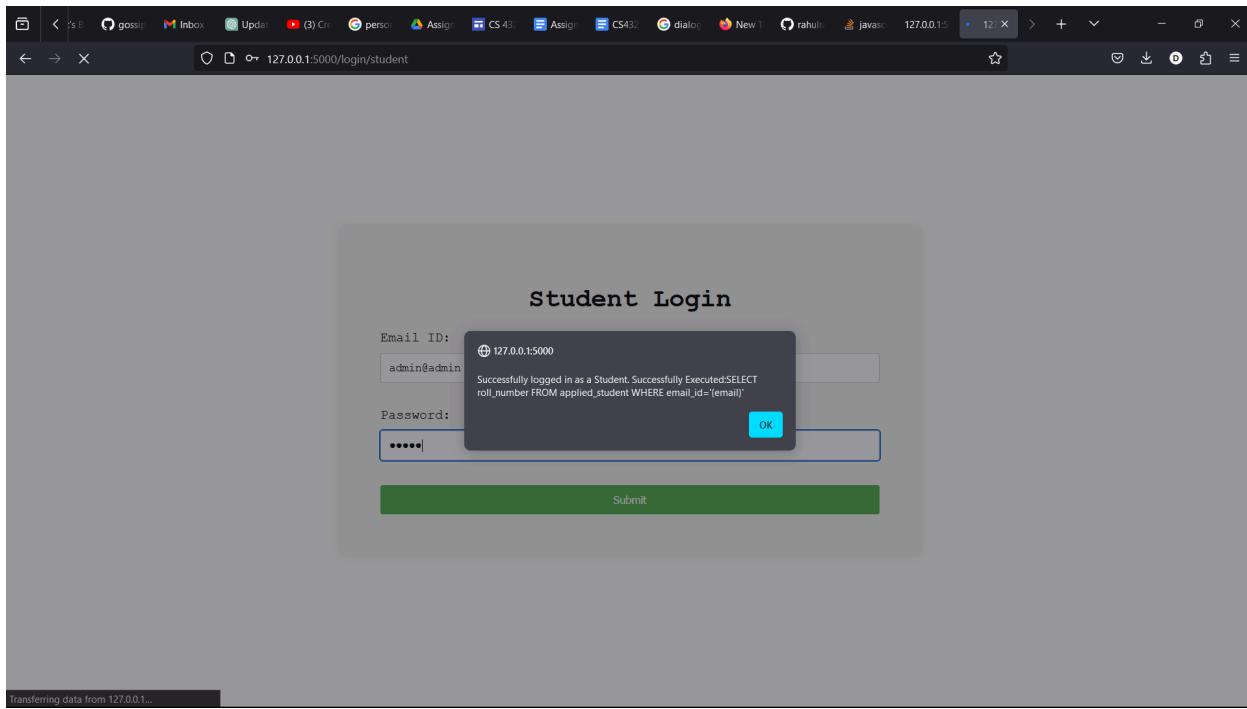


Fig. Successful Login

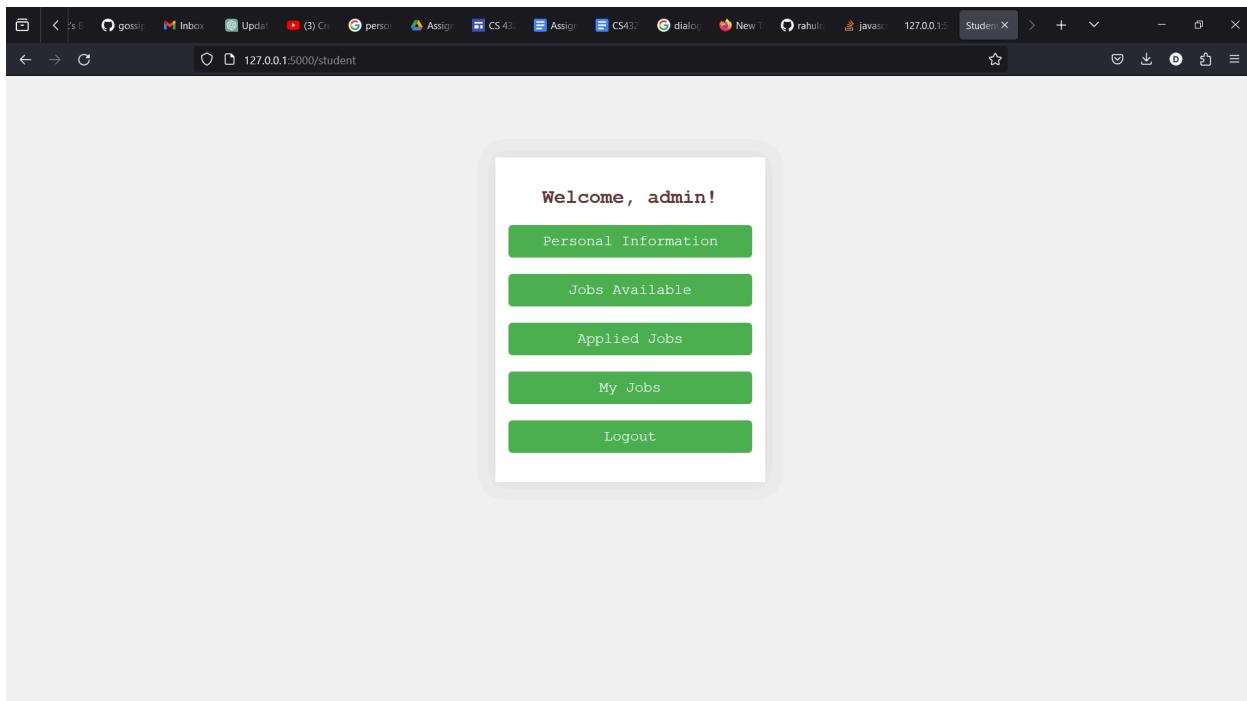


Fig. After Successful Login

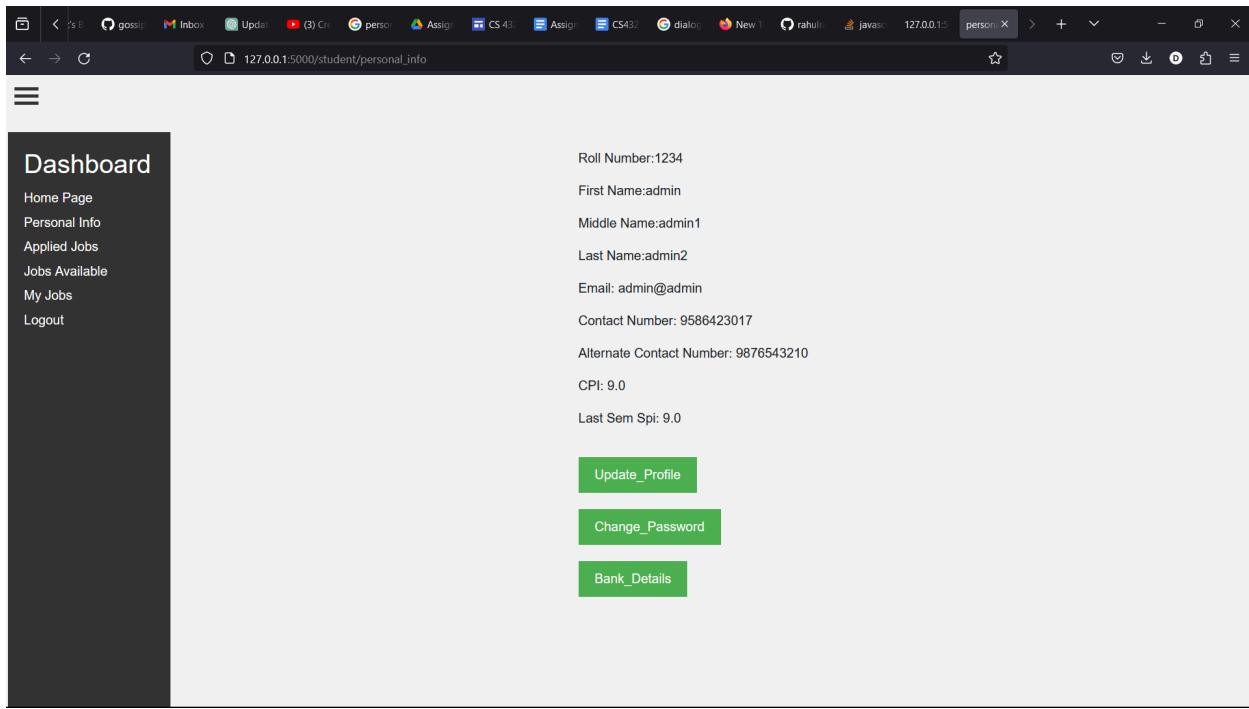


Fig. Personal Information Page (along with sidebar-Dashboard)

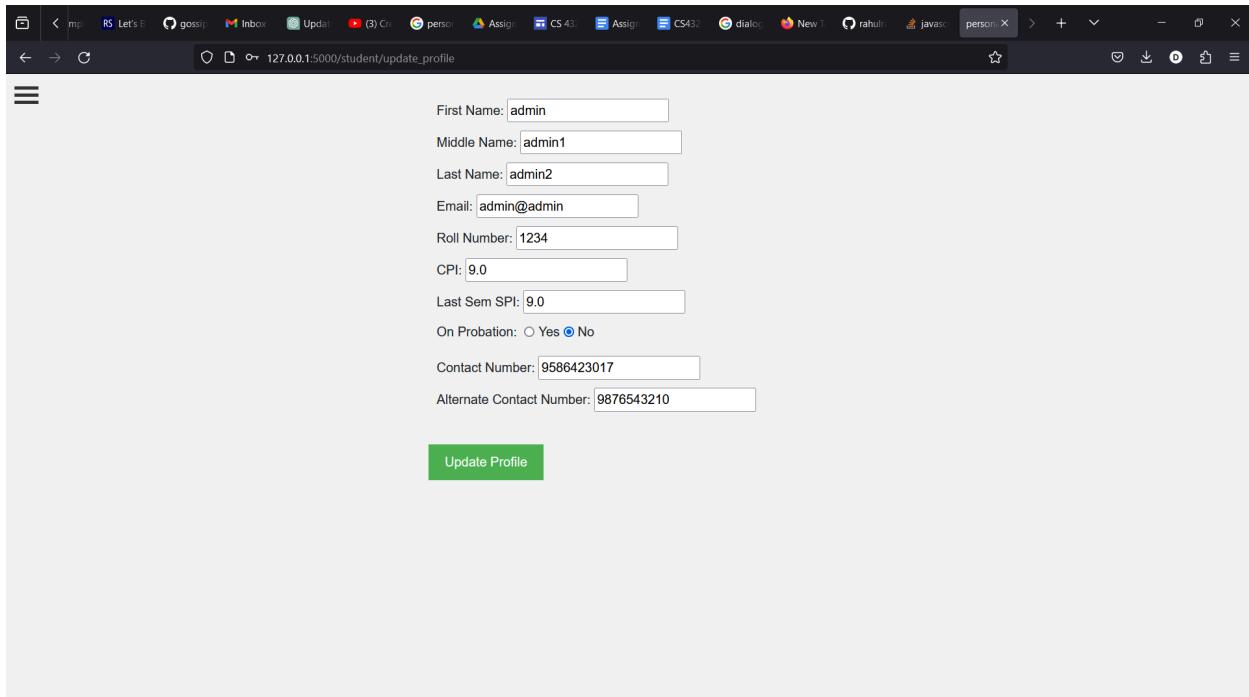


Fig. Update_Profile Page

Change Password

Old Password:

New Password:

Confirm Password:

Change Password

Fig. Change Password Page

Bank Details

Bank Name	Account Number	IFSC Code
HDFC	890123456798	CDEF0123480

Edit Bank Details

Fig. Bank Details Page

Available Jobs

job_id	job_type	job_description	min_qualifications	job_criteria	prerequisites	additional_info	pay_per_hour	no_of_positions	start_date	end_date	tenure	faculty_id	is_position_open	is_available	application_deadline
104	other	Public Speaking Coach for Debating Club. The coach will train students in public speaking techniques, help them prepare for debates, and guide them in improving communication skills.	7.6	Experience in public speaking	Interest in public speaking and debate	Certification in public speaking or debate coaching is a plus.	185	2	2024-01-25	2024-04-25	91	1013	1	yes	2023-12-31 Apply
106	other	Digital Marketing Mentor for Business School. The mentor will guide students in understanding digital marketing strategies, conducting marketing	7.7	Experience in digital marketing	Interest in marketing, Basic knowledge of digital marketing concepts	Certification in digital marketing is a plus.	195	2	2024-02-10	2024-05-10	90	1027	1	yes	2024-01-31 Apply

Fig. List of All Available Jobs

Job Application

Job:

Current CPI:

Last Semester SPI:

Statement of Motivation:

[Submit Application](#)

Fig. On clicking Applying Button

Application Id	Job Id	Job Type	Job Description	Approval Status

Fig. List of Applied Jobs by the Students

Application Id	Job Id	Job Type	Job Description	Approval Status

Fig. List of the Jobs of the Student

PROFESSOR LOGIN

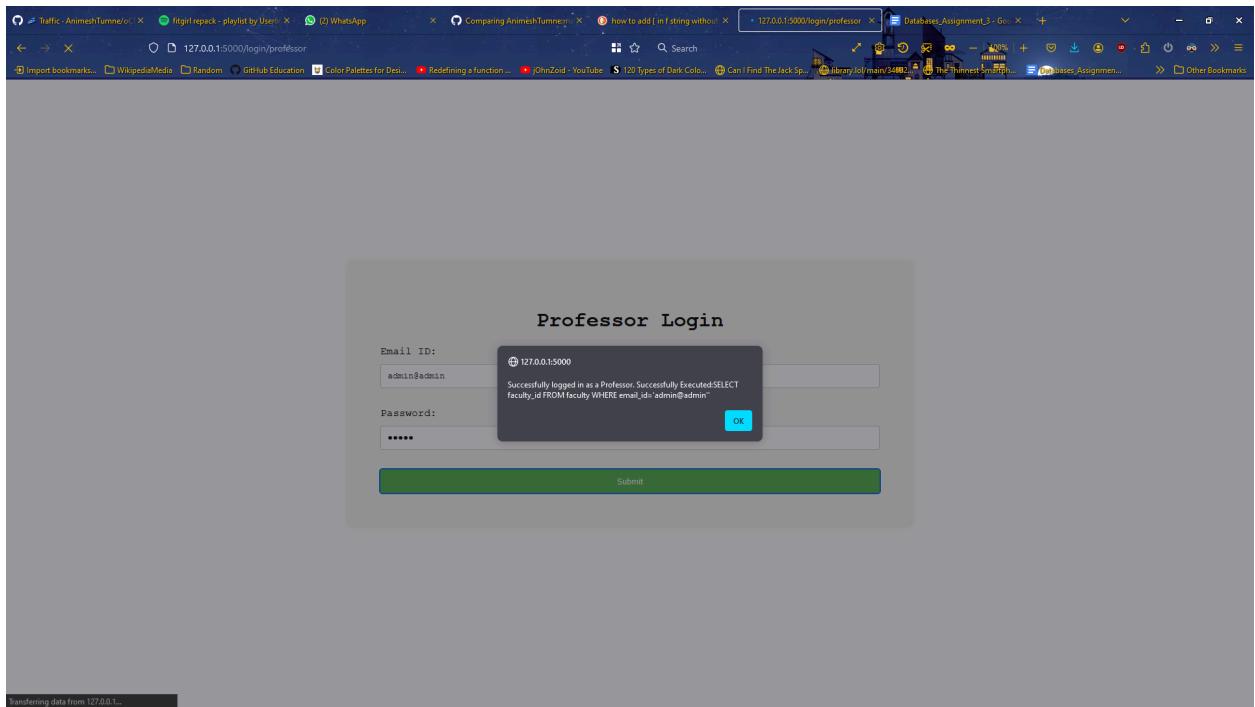


Fig. Successful Login

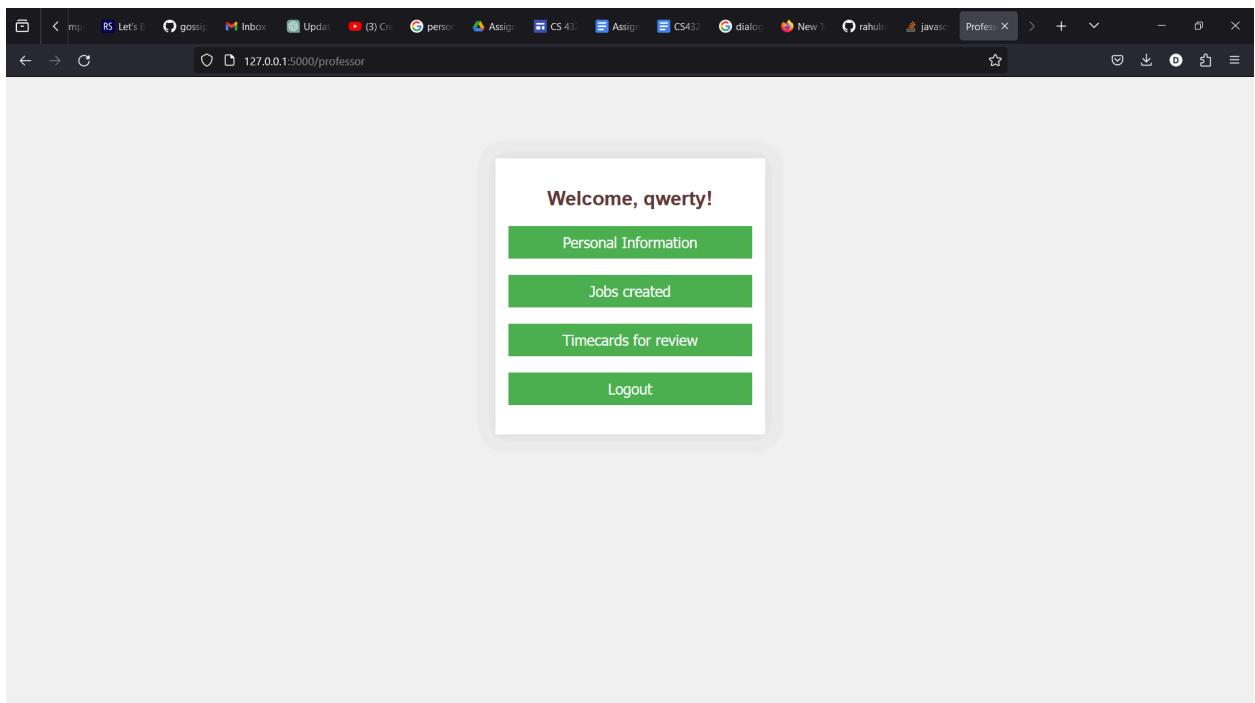


Fig. After Professor Login

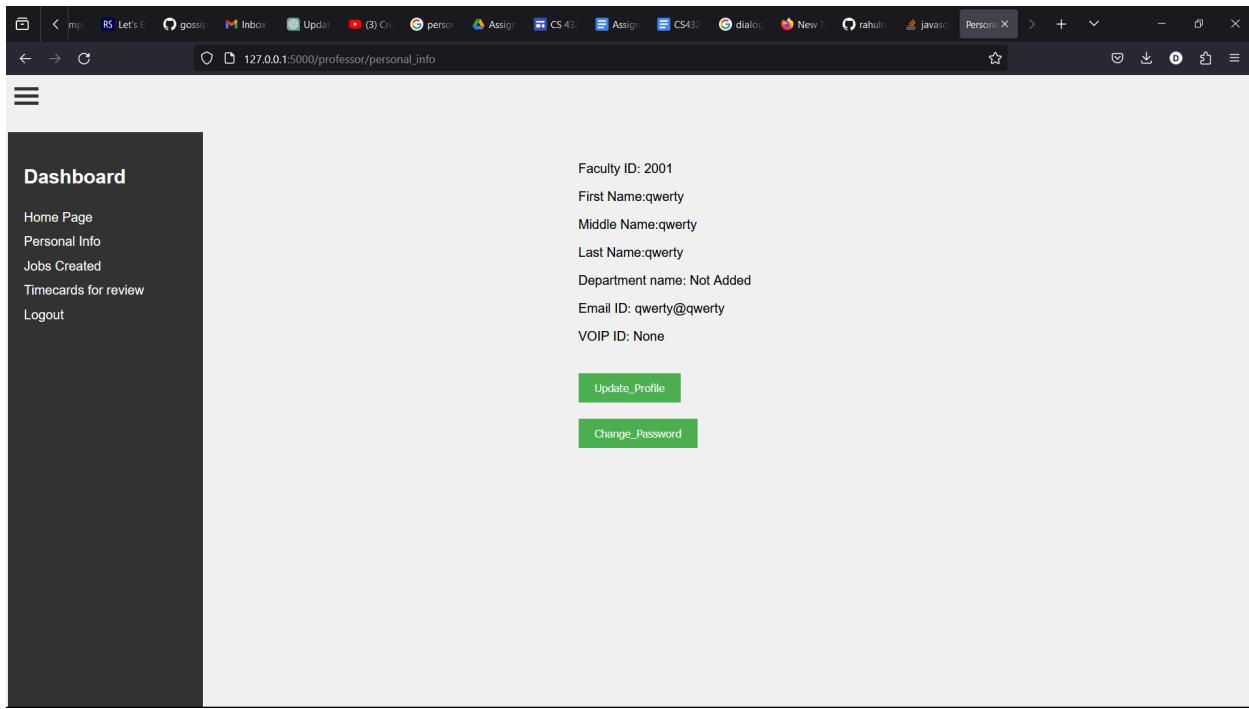


Fig. Personal Information Page

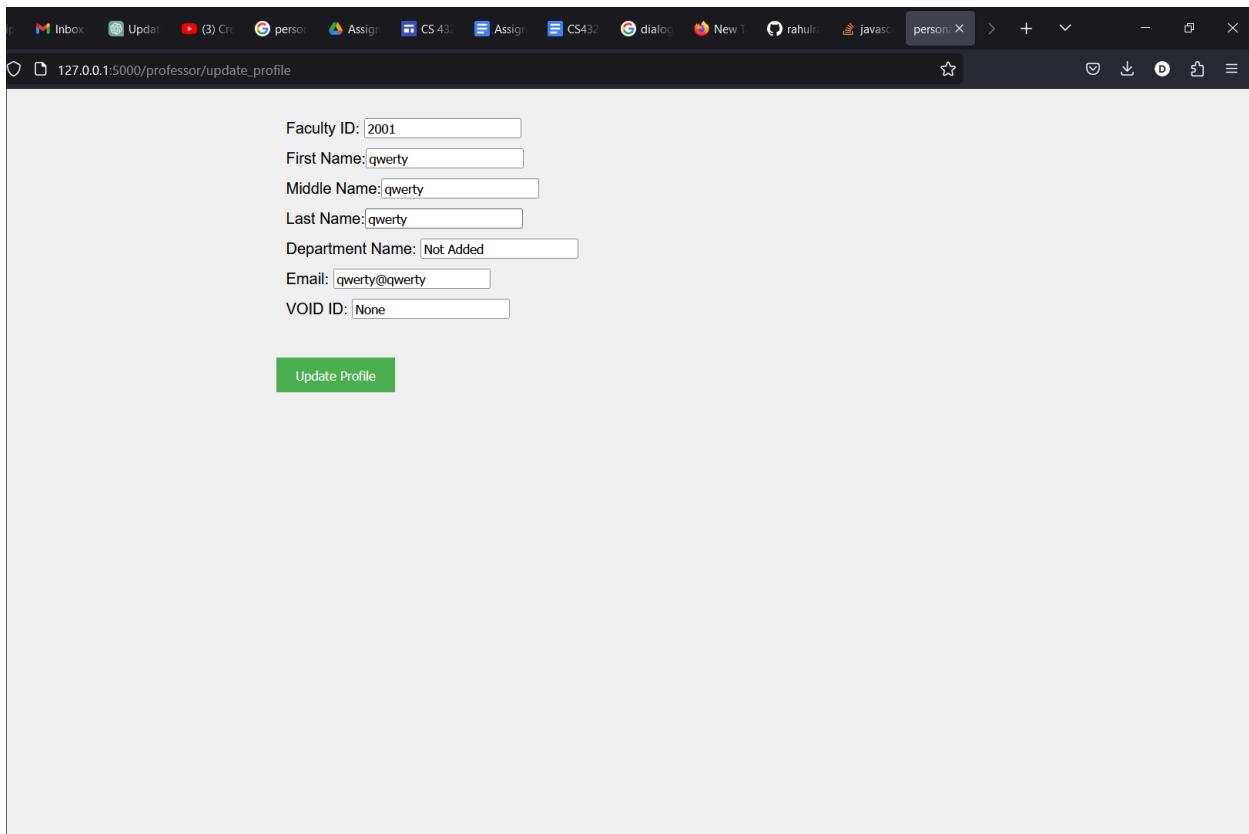


Fig. Update Profile Page

Change Password

Old Password:

New Password:

Confirm Password:

change_password

Fig. Change Password Page

Jobs Created

Create a new job

job_id	job_type	job_description	min_qualifications	job_criteria	prerequisites	additional_info	pay_per_hour	no_of_positions	start_date	end_date	tenure	is_available	application_deadline

Fig. Jobs Created Page

Job Application

Job type:

ADH
 PAL
 Others

Job Description:

Enter Job Description

Minimum qualification:

Enter Minimum Qualification

Job Criteria:

Enter Job Criteria

Prerequisites:

Enter Prerequisites

Pay per hour:

Enter Pay per Hour

Number of Positions:

Enter Number of Positions

Start Date:

dd/mm/yyyy

End Date:

dd / mm / yyyy

Tenure:

Enter Tenure

Application Deadline:

dd / mm / yyyy

Submit Application

Fig. Creating New Job

job_id	roll_number	month	year	hours_worked	work_description	is_approved

Fig. List of all Timecards

Tasks of G2:

```

main_project> main_app.py > student.personal.info.change
1  from flask import Flask, render_template, request, redirect, url_for, session
2  from flask_mysqldb import MySQL
3  from flask_bcrypt import Bcrypt
4  import random
5  from flask import session, redirect, url_for, render_template_string
6
7  app = Flask(__name__)
8  bcrypt = Bcrypt(app)
9  app.secret_key = 'mySecretKey'
10
11 app.config['MYSQL_HOST'] = "localhost"
12 app.config['MYSQL_USER'] = "oceoAdmin"
13 app.config['MYSQL_PASSWORD'] = "oceoAdmin"
14 app.config['MYSQL_DB'] = "oceo_management"
15
16 db = MySQL(app)
17 #
18 #
19 # Home Page - Select User Type
20 @app.route("/")
21 def index():
22     return render_template("index.html")
23
24 # NEW USER REGISTRATION
25 @app.route("/login/new_user", methods=["GET", "POST"])
26 > def register():
27     #
28     # -----
29     # check the credentials of the user
30     or <--> https://github.com/rohit-pranjal/CORRECT
31
32     # -----
33     # check if user exists
34     # -----
35     # insert user into database
36     # -----
37     # return success message
38
39     # -----
40     # check if user exists
41     # -----
42     # insert user into database
43     # -----
44     # return success message
45
46     # -----
47     # check if user exists
48     # -----
49     # insert user into database
50     # -----
51     # return success message
52
53     # -----
54     # check if user exists
55     # -----
56     # insert user into database
57     # -----
58     # return success message
59
60     # -----
61     # check if user exists
62     # -----
63     # insert user into database
64     # -----
65     # return success message
66
67     # -----
68     # check if user exists
69     # -----
70     # insert user into database
71     # -----
72     # return success message
73
74     # -----
75     # check if user exists
76     # -----
77     # insert user into database
78     # -----
79     # return success message
80
81     # -----
82     # check if user exists
83     # -----
84     # insert user into database
85     # -----
86     # return success message
87
88     # -----
89     # check if user exists
90     # -----
91     # insert user into database
92     # -----
93     # return success message
94
95     # -----
96     # check if user exists
97     # -----
98     # insert user into database
99     # -----
100    # return success message
101
102    # -----
103    # check if user exists
104    # -----
105    # insert user into database
106    # -----
107    # return success message
108
109    # -----
110    # check if user exists
111    # -----
112    # insert user into database
113    # -----
114    # return success message
115
116    # -----
117    # check if user exists
118    # -----
119    # insert user into database
120    # -----
121    # return success message
122
123    # -----
124    # check if user exists
125    # -----
126    # insert user into database
127    # -----
128    # return success message
129
130    # -----
131    # check if user exists
132    # -----
133    # insert user into database
134    # -----
135    # return success message
136
137    # -----
138    # check if user exists
139    # -----
140    # insert user into database
141    # -----
142    # return success message
143
144    # -----
145    # check if user exists
146    # -----
147    # insert user into database
148    # -----
149    # return success message
150
151    # -----
152    # check if user exists
153    # -----
154    # insert user into database
155    # -----
156    # return success message
157
158    # -----
159    # check if user exists
160    # -----
161    # insert user into database
162    # -----
163    # return success message
164
165    # -----
166    # check if user exists
167    # -----
168    # insert user into database
169    # -----
170    # return success message
171
172    # -----
173    # check if user exists
174    # -----
175    # insert user into database
176    # -----
177    # return success message
178
179    # -----
180    # check if user exists
181    # -----
182    # insert user into database
183    # -----
184    # return success message
185
186    # -----
187    # check if user exists
188    # -----
189    # insert user into database
190    # -----
191    # return success message
192
193    # -----
194    # check if user exists
195    # -----
196    # insert user into database
197    # -----
198    # return success message
199
200    # -----
201    # check if user exists
202    # -----
203    # insert user into database
204    # -----
205    # return success message
206
207    # -----
208    # check if user exists
209    # -----
210    # insert user into database
211    # -----
212    # return success message
213
214    # -----
215    # check if user exists
216    # -----
217    # insert user into database
218    # -----
219    # return success message
220
221    # -----
222    # check if user exists
223    # -----
224    # insert user into database
225    # -----
226    # return success message
227
228    # -----
229    # check if user exists
230    # -----
231    # insert user into database
232    # -----
233    # return success message
234
235    # -----
236    # check if user exists
237    # -----
238    # insert user into database
239    # -----
240    # return success message
241
242    # -----
243    # check if user exists
244    # -----
245    # insert user into database
246    # -----
247    # return success message
248
249    # -----
250    # check if user exists
251    # -----
252    # insert user into database
253    # -----
254    # return success message
255
256    # -----
257    # check if user exists
258    # -----
259    # insert user into database
260    # -----
261    # return success message
262
263    # -----
264    # check if user exists
265    # -----
266    # insert user into database
267    # -----
268    # return success message
269
270    # -----
271    # check if user exists
272    # -----
273    # insert user into database
274    # -----
275    # return success message
276
277    # -----
278    # check if user exists
279    # -----
280    # insert user into database
281    # -----
282    # return success message
283
284    # -----
285    # check if user exists
286    # -----
287    # insert user into database
288    # -----
289    # return success message
290
291    # -----
292    # check if user exists
293    # -----
294    # insert user into database
295    # -----
296    # return success message
297
298    # -----
299    # check if user exists
300    # -----
301    # insert user into database
302    # -----
303    # return success message
304
305    # -----
306    # check if user exists
307    # -----
308    # insert user into database
309    # -----
310    # return success message
311
312    # -----
313    # check if user exists
314    # -----
315    # insert user into database
316    # -----
317    # return success message
318
319    # -----
320    # check if user exists
321    # -----
322    # insert user into database
323    # -----
324    # return success message
325
326    # -----
327    # check if user exists
328    # -----
329    # insert user into database
330    # -----
331    # return success message
332
333    # -----
334    # check if user exists
335    # -----
336    # insert user into database
337    # -----
338    # return success message
339
340    # -----
341    # check if user exists
342    # -----
343    # insert user into database
344    # -----
345    # return success message
346
347    # -----
348    # check if user exists
349    # -----
350    # insert user into database
351    # -----
352    # return success message
353
354    # -----
355    # check if user exists
356    # -----
357    # insert user into database
358    # -----
359    # return success message
360
361    # -----
362    # check if user exists
363    # -----
364    # insert user into database
365    # -----
366    # return success message
367
368    # -----
369    # check if user exists
370    # -----
371    # insert user into database
372    # -----
373    # return success message
374
375    # -----
376    # check if user exists
377    # -----
378    # insert user into database
379    # -----
380    # return success message
381
382    # -----
383    # check if user exists
384    # -----
385    # insert user into database
386    # -----
387    # return success message
388
389    # -----
390    # check if user exists
391    # -----
392    # insert user into database
393    # -----
394    # return success message
395
396    # -----
397    # check if user exists
398    # -----
399    # insert user into database
400    # -----
401    # return success message
402
403    # -----
404    # check if user exists
405    # -----
406    # insert user into database
407    # -----
408    # return success message
409
410    # -----
411    # check if user exists
412    # -----
413    # insert user into database
414    # -----
415    # return success message
416
417    # -----
418    # check if user exists
419    # -----
420    # insert user into database
421    # -----
422    # return success message
423
424    # -----
425    # check if user exists
426    # -----
427    # insert user into database
428    # -----
429    # return success message
430
431    # -----
432    # check if user exists
433    # -----
434    # insert user into database
435    # -----
436    # return success message
437
438    # -----
439    # check if user exists
440    # -----
441    # insert user into database
442    # -----
443    # return success message
444
445    # -----
446    # check if user exists
447    # -----
448    # insert user into database
449    # -----
450    # return success message
451
452    # -----
453    # check if user exists
454    # -----
455    # insert user into database
456    # -----
457    # return success message
458
459    # -----
460    # check if user exists
461    # -----
462    # insert user into database
463    # -----
464    # return success message
465
466    # -----
467    # check if user exists
468    # -----
469    # insert user into database
470    # -----
471    # return success message
472
473    # -----
474    # check if user exists
475    # -----
476    # insert user into database
477    # -----
478    # return success message
479
480    # -----
481    # check if user exists
482    # -----
483    # insert user into database
484    # -----
485    # return success message
486
487    # -----
488    # check if user exists
489    # -----
490    # insert user into database
491    # -----
492    # return success message
493
494    # -----
495    # check if user exists
496    # -----
497    # insert user into database
498    # -----
499    # return success message
500
501    # -----
502    # check if user exists
503    # -----
504    # insert user into database
505    # -----
506    # return success message
507
508    # -----
509    # check if user exists
510    # -----
511    # insert user into database
512    # -----
513    # return success message
514
515    # -----
516    # check if user exists
517    # -----
518    # insert user into database
519    # -----
520    # return success message
521
522    # -----
523    # check if user exists
524    # -----
525    # insert user into database
526    # -----
527    # return success message
528
529    # -----
530    # check if user exists
531    # -----
532    # insert user into database
533    # -----
534    # return success message
535
536    # -----
537    # check if user exists
538    # -----
539    # insert user into database
540    # -----
541    # return success message
542
543    # -----
544    # check if user exists
545    # -----
546    # insert user into database
547    # -----
548    # return success message
549
550    # -----
551    # check if user exists
552    # -----
553    # insert user into database
554    # -----
555    # return success message
556
557    # -----
558    # check if user exists
559    # -----
560    # insert user into database
561    # -----
562    # return success message
563
564    # -----
565    # check if user exists
566    # -----
567    # insert user into database
568    # -----
569    # return success message
570
571    # -----
572    # check if user exists
573    # -----
574    # insert user into database
575    # -----
576    # return success message
577
578    # -----
579    # check if user exists
580    # -----
581    # insert user into database
582    # -----
583    # return success message
584
585    # -----
586    # check if user exists
587    # -----
588    # insert user into database
589    # -----
590    # return success message
591
592    # -----
593    # check if user exists
594    # -----
595    # insert user into database
596    # -----
597    # return success message
598
599    # -----
600    # check if user exists
601    # -----
602    # insert user into database
603    # -----
604    # return success message
605
606    # -----
607    # check if user exists
608    # -----
609    # insert user into database
610    # -----
611    # return success message
612
613    # -----
614    # check if user exists
615    # -----
616    # insert user into database
617    # -----
618    # return success message
619
620    # -----
621    # check if user exists
622    # -----
623    # insert user into database
624    # -----
625    # return success message
626
627    # -----
628    # check if user exists
629    # -----
630    # insert user into database
631    # -----
632    # return success message
633
634    # -----
635    # check if user exists
636    # -----
637    # insert user into database
638    # -----
639    # return success message
640
641    # -----
642    # check if user exists
643    # -----
644    # insert user into database
645    # -----
646    # return success message
647
648    # -----
649    # check if user exists
650    # -----
651    # insert user into database
652    # -----
653    # return success message
654
655    # -----
656    # check if user exists
657    # -----
658    # insert user into database
659    # -----
660    # return success message
661
662    # -----
663    # check if user exists
664    # -----
665    # insert user into database
666    # -----
667    # return success message
668
669    # -----
670    # check if user exists
671    # -----
672    # insert user into database
673    # -----
674    # return success message
675
676    # -----
677    # check if user exists
678    # -----
679    # insert user into database
680    # -----
681    # return success message
682
683    # -----
684    # check if user exists
685    # -----
686    # insert user into database
687    # -----
688    # return success message
689
690    # -----
691    # check if user exists
692    # -----
693    # insert user into database
694    # -----
695    # return success message
696
697    # -----
698    # check if user exists
699    # -----
700    # insert user into database
701    # -----
702    # return success message
703
704    # -----
705    # check if user exists
706    # -----
707    # insert user into database
708    # -----
709    # return success message
710
711    # -----
712    # check if user exists
713    # -----
714    # insert user into database
715    # -----
716    # return success message
717
718    # -----
719    # check if user exists
720    # -----
721    # insert user into database
722    # -----
723    # return success message
724
725    # -----
726    # check if user exists
727    # -----
728    # insert user into database
729    # -----
730    # return success message
731
732    # -----
733    # check if user exists
734    # -----
735    # insert user into database
736    # -----
737    # return success message
738
739    # -----
740    # check if user exists
741    # -----
742    # insert user into database
743    # -----
744    # return success message
745
746    # -----
747    # check if user exists
748    # -----
749    # insert user into database
750    # -----
751    # return success message
752
753    # -----
754    # check if user exists
755    # -----
756    # insert user into database
757    # -----
758    # return success message
759
760    # -----
761    # check if user exists
762    # -----
763    # insert user into database
764    # -----
765    # return success message
766
767    # -----
768    # check if user exists
769    # -----
770    # insert user into database
771    # -----
772    # return success message
773
774    # -----
775    # check if user exists
776    # -----
777    # insert user into database
778    # -----
779    # return success message
780
781    # -----
782    # check if user exists
783    # -----
784    # insert user into database
785    # -----
786    # return success message
787
788    # -----
789    # check if user exists
790    # -----
791    # insert user into database
792    # -----
793    # return success message
794
795    # -----
796    # check if user exists
797    # -----
798    # insert user into database
799    # -----
800    # return success message
801
802    # -----
803    # check if user exists
804    # -----
805    # insert user into database
806    # -----
807    # return success message
808
809    # -----
810    # check if user exists
811    # -----
812    # insert user into database
813    # -----
814    # return success message
815
816    # -----
817    # check if user exists
818    # -----
819    # insert user into database
820    # -----
821    # return success message
822
823    # -----
824    # check if user exists
825    # -----
826    # insert user into database
827    # -----
828    # return success message
829
830    # -----
831    # check if user exists
832    # -----
833    # insert user into database
834    # -----
835    # return success message
836
837    # -----
838    # check if user exists
839    # -----
840    # insert user into database
841    # -----
842    # return success message
843
844    # -----
845    # check if user exists
846    # -----
847    # insert user into database
848    # -----
849    # return success message
850
851    # -----
852    # check if user exists
853    # -----
854    # insert user into database
855    # -----
856    # return success message
857
858    # -----
859    # check if user exists
860    # -----
861    # insert user into database
862    # -----
863    # return success message
864
865    # -----
866    # check if user exists
867    # -----
868    # insert user into database
869    # -----
870    # return success message
871
872    # -----
873    # check if user exists
874    # -----
875    # insert user into database
876    # -----
877    # return success message
878
879    # -----
880    # check if user exists
881    # -----
882    # insert user into database
883    # -----
884    # return success message
885
886    # -----
887    # check if user exists
888    # -----
889    # insert user into database
890    # -----
891    # return success message
892
893    # -----
894    # check if user exists
895    # -----
896    # insert user into database
897    # -----
898    # return success message
899
900    # -----
901    # check if user exists
902    # -----
903    # insert user into database
904    # -----
905    # return success message
906
907    # -----
908    # check if user exists
909    # -----
910    # insert user into database
911    # -----
912    # return success message
913
914    # -----
915    # check if user exists
916    # -----
917    # insert user into database
918    # -----
919    # return success message
920
921    # -----
922    # check if user exists
923    # -----
924    # insert user into database
925    # -----
926    # return success message
927
928    # -----
929    # check if user exists
930    # -----
931    # insert user into database
932    # -----
933    # return success message
934
935    # -----
936    # check if user exists
937    # -----
938    # insert user into database
939    # -----
940    # return success message
941
942    # -----
943    # check if user exists
944    # -----
945    # insert user into database
946    # -----
947    # return success message
948
949    # -----
950    # check if user exists
951    # -----
952    # insert user into database
953    # -----
954    # return success message
955
956    # -----
957    # check if user exists
958    # -----
959    # insert user into database
960    # -----
961    # return success message
962
963    # -----
964    # check if user exists
965    # -----
966    # insert user into database
967    # -----
968    # return success message
969
970    # -----
971    # check if user exists
972    # -----
973    # insert user into database
974    # -----
975    # return success message
976
977    # -----
978    # check if user exists
979    # -----
980    # insert user into database
981    # -----
982    # return success message
983
984    # -----
985    # check if user exists
986    # -----
987    # insert user into database
988    # -----
989    # return success message
990
991    # -----
992    # check if user exists
993    # -----
994    # insert user into database
995    # -----
996    # return success message
997
998    # -----
999    # check if user exists
1000   # -----
1001   # insert user into database
1002   # -----
1003   # return success message
1004
1005   # -----
1006   # check if user exists
1007   # -----
1008   # insert user into database
1009   # -----
1010   # return success message
1011
1012   # -----
1013   # check if user exists
1014   # -----
1015   # insert user into database
1016   # -----
1017   # return success message
1018
1019   # -----
1020   # check if user exists
1021   # -----
1022   # insert user into database
1023   # -----
1024   # return success message
1025
1026   # -----
1027   # check if user exists
1028   # -----
1029   # insert user into database
1030   # -----
1031   # return success message
1032
1033   # -----
1034   # check if user exists
1035   # -----
1036   # insert user into database
1037   # -----
1038   # return success message
1039
1040   # -----
1041   # check if user exists
1042   # -----
1043   # insert user into database
1044   # -----
1045   # return success message
1046
1047   # -----
1048   # check if user exists
1049   # -----
1050   # insert user into database
1051   # -----
1052   # return success message
1053
1054   # -----
1055   # check if user exists
1056   # -----
1057   # insert user into database
1058   # -----
1059   # return success message
1060
1061   # -----
1062   # check if user exists
1063   # -----
1064   # insert user into database
1065   # -----
1066   # return success message
1067
1068   # -----
1069   # check if user exists
1070   # -----
1071   # insert user into database
1072   # -----
1073   # return success message
1074
1075   # -----
1076   # check if user exists
1077   # -----
1078   # insert user into database
1079   # -----
1080   # return success message
1081
1082   # -----
1083   # check if user exists
1084   # -----
1085   # insert user into database
1086   # -----
1087   # return success message
1088
1089   # -----
1090   # check if user exists
1091   # -----
1092   # insert user into database
1093   # -----
1094   # return success message
1095
1096   # -----
1097   # check if user exists
1098   # -----
1099   # insert user into database
1100   # -----
1101   # return success message
1102
1103   # -----
1104   # check if user exists
1105   # -----
1106   # insert user into database
1107   # -----
1108   # return success message
1109
1110   # -----
1111   # check if user exists
1112   # -----
1113   # insert user into database
1114   # -----
1115   # return success message
1116
1117   # -----
1118   # check if user exists
1119   # -----
1120   # insert user into database
1121   # -----
1122   # return success message
1123
1124   # -----
1125   # check if user exists
1126   # -----
1127   # insert user into database
1128   # -----
1129   # return success message
1130
1131   # -----
1132   # check if user exists
1133   # -----
1134   # insert user into database
1135   # -----
1136   # return success message
1137
1138   # -----
1139   # check if user exists
1140   # -----
1141   # insert user into database
1142   # -----
1143   # return success message
1144
1145   # -----
1146   # check if user exists
1147   # -----
1148   # insert user into database
1149   # -----
1150   # return success message
1151
1152   # -----
1153   # check if user exists
1154   # -----
1155   # insert user into database
1156   # -----
1157   # return success message
1158
1159   # -----
1160   # check if user exists
1161   # -----
1162   # insert user into database
1163   # -----
1164   # return success message
1165
1166   # -----
1167   # check if user exists
1168   # -----
1169   # insert user into database
1170   # -----
1171   # return success message
1172
1173   # -----
1174   # check if user exists
1175   # -----
1176   # insert user into database
1177   # -----
1178   # return success message
1179
1180   # -----
1181   # check if user exists
1182   # -----
1183   # insert user into database
1184   # -----
1185   # return success message
1186
1187   # -----
1188   # check if user exists
1189   # -----
1190   # insert user into database
1191   # -----
1192   # return success message
1193
1194   # -----
1195   # check if user exists
1196   # -----
1197   # insert user into database
1198   # -----
1199   # return success message
1200
1201   # -----
1202   # check if user exists
1203   # -----
1204   # insert user into database
1205   # -----
1206   # return success message
1207
1208   # -----
1209   # check if user exists
1210   # -----
1211   # insert user into database
1212   # -----
1213   # return success message
1214
1215   # -----
1216   # check if user exists
1217   # -----
1218   # insert user into database
1219   # -----
1220   # return success message
1221
1222   # -----
1223   # check if user exists
1224   # -----
1225   # insert user into database
1226   # -----
1227   # return success message
1228
1229   # -----
1230   # check if user exists
1231   # -----
1232   # insert user into database
1233   # -----
1234   # return success message
1235
1236   # -----
1237   # check if user exists
1238   # -----
1239   # insert user into database
1240   # -----
1241   # return success message
1242
1243   # -----
1244   # check if user exists
1245   # -----
1246   # insert user into database
1247   # -----
1248   # return success message
1249
1250   # -----
1251   # check if user exists
1252   # -----
1253   # insert user into database
1254   # -----
1255   # return success message
1256
1257   # -----
1258   # check if user exists
1259   # -----
1260   # insert user into database
1261   # -----
1262   # return success message
1263
1264   # -----
1265   # check if user exists
1266   # -----
1267   # insert user into database
1268   # -----
1269   # return success message
1270
1271   # -----
1272   # check if user exists
1273   # -----
1274   # insert user into database
1275   # -----
1276   # return success message
1277
1278   # -----
1279   # check if user exists
1280   # -----
1281   # insert user into database
1282   # -----
1283   # return success message
1284
1285   # -----
1286   # check if user exists
1287   # -----
1288   # insert user into database
1289   # -----
1290   # return success message
1291
1292   # -----
1293   # check if user exists
1294   # -----
1295   # insert user into database
1296   # -----
1297   # return success message
1298
1299   # -----
1300   # check if user exists
1301   # -----
1302   # insert user into database
1303   # -----
1304   # return success message
1305
1306   # -----
1307   # check if user exists
1308   # -----
1309   # insert user into database
1310   # -----
1311   # return success message
1312
1313   # -----
1314   # check if user exists
1315   # -----
1316   # insert user into database
1317   # -----
1318   # return success message
1319
1320   # -----
1321   # check if user exists
1322   # -----
1323   # insert user into database
1324   # -----
1325   # return success message
1326
1327   # -----
1328   # check if user exists
1329   # -----
1330   # insert user into database
1331   # -----
1332   # return success message
1333
1334   # -----
1335   # check if user exists
1336   # -----
1337   # insert user into database
1338   # -----
1339   # return success message
1340
1341   # -----
1342   # check if user exists
1343   # -----
1344   # insert user into database
1345   # -----
1346   # return success message
1347
1348   # -----
1349   # check if user exists
1350   # -----
1351   # insert user into database
1352   # -----
1353   # return success message
1354
1355   # -----
1356   # check if user exists
1357   # -----
1358   # insert user into database
1359   # -----
1360   # return success message
1361
1362   # -----
1363   # check if user exists
1364   # -----
1365   # insert user into database
1366   # -----
1367   # return success message
1368
1369   # -----
1370   # check if user exists
1371   # -----
1372   # insert user into database
1373   # -----
1374   # return success message
1375
1376   # -----
1377   # check if user exists
1378   # -----
1379   # insert user into database
1380   # -----
1381   # return success message
1382
1383   # -----
1384   # check if user exists
1385   # -----
1386   # insert user into database
1387   # -----
1388   # return success message
1389
1390   # -----
1391   # check if user exists
1392   # -----
1393   # insert user into database
1394   # -----
1395   # return success message
1396
1397   # -----
1398   # check if user exists
1399   # -----
1400   # insert user into database
1401   # -----
1402   # return success message
1403
1404   # -----
1405   # check if user exists
1406   # -----
1407   # insert user into database
1408   # -----
1409   # return success message
1410
1411   # -----
1412   # check if user exists
1413   # -----
1414   # insert user into database
1415   # -----
1416   # return success message
1417
1418   # -----
1419   # check if user exists
1420   # -----
1421   # insert user into database
1422   # -----
1423   # return success message
1424
1425   # -----
1426   # check if user exists
1427   # -----
1428   # insert user into database
1429   # -----
1430   # return success message
1431

```

Integrated the database with the backend logic using Flask. Authentication is implemented for every user type and every user page. All the Routing and respective required SQL queries are implemented, Password column for every user is added to the applied_student table. We are using Bcrypt to encrypt the provided password, and the encrypted password is stored in the database to ensure security. We try to handle most of the edge cases that can arise.

The backend logic is designed in such a way that every time a new user comes, he/she will register by specifying the user type. For student users, there are options to view the available jobs, where there is a button corresponding to each job through which the student can apply for the job. After that, students can view the status of the applications in another section, and there is also a section for all the approved jobs (i.e. My Jobs), where the student can add time cards corresponding to each job and also view the timecard status. Students are also provided with an option to cancel the applications.

Similarly, for Professors, different web pages are designed for different functionalities. Every Professor can create jobs and provide relevant job details; The professor has provided the option to stop accepting applications. When the Professor stop accepting the applications, the job will not be available for students to view, and therefore they cannot apply for the job.

Each application is reviewed by the Admin, Dean, Student Affairs Junior Superintendent (sa_js), and oCEO_coordinator and after all of them approve the application, then the status of the application will be set to "approved". After faculty approval, the application will go to the Admin and then to the oCEO coordinator and the Dean; after the approval of the Dean, the status will be set to "approved". If any one of them rejects the application, it will be set to "rejected".

Overall we tried to implement all the functionalities and create a full-fledged Web application; we are ready to add other required functionalities based on the feedback from the Stakeholders. Based on the feedback from Stakeholders, we will implement the grant to student's bank details to the desired stakeholders for the payment.

Tasks of G1 and G2 combined:

Steps on how to run our GitHub repository have been written down in the README.md.

https://github.com/AnimeshTumne/oCEO_management_DataWeavers/blob/main/README.md

Implementation of Dynamic Execution:

INSERT

```
others.html main_app.py X after_login.html student.html personal_info_cha D <
main_project > main_app.py > professor_add_job
750 ssor_add_job():
751     cursor = db.connection.cursor()
752     cursor.execute("SELECT count(*) FROM job")
753     total_count = cursor.fetchone()[0]
754     if total_count == 0:
755         job_id = random.randint(12345678, 29999999)
756     else:
757         cursor.execute("SELECT MAX(job_id) FROM job")
758         max_id = cursor.fetchone()[0]
759         job_id = max_id + 1
760
761     cursor.execute(f"INSERT INTO job (job_id, job_type, job_description, min_qualifications, job_c
762 db.connection.commit()
763 if job_type == 'ADH':
764     course_id = request.form.get('course_id')
765     course_name = request.form.get('course_name')
766     cursor.execute(f"INSERT INTO adh (job_id, course_id, course_name) VALUES ({job_id}, '{cour
767 db.connection.commit()
768 elif job_type == 'PAL':
769     subjects_to_teach = request.form.get('subjects_to_teach')
770     cursor.execute(f"INSERT INTO subjects_under_pal (job_id, subjects_to_teach) VALUES ({job_i
771 db.connection.commit()
772 else:
773     role_name = request.form.get('role_name')
774     cursor.execute(f"INSERT INTO other (job_id, role_name) VALUES ({job_id}, '{role_name}');")
775     db.connection.commit()
776
777 cursor.close()
```

Fig. Example of INSERT Usage

New User Registration

Roll Number/Employee ID:	<input type="text" value="21100500"/>
First Name:	<input type="text" value="Deshi"/>
Middle Name:	<input type="text"/>
Last Name:	<input type="text" value="Doshi"/>
Email ID:	<input type="text" value="deshidoshi@tgt.ac.in"/>
Password:	<input type="password" value="*****"/>
Confirm Password:	<input type="password" value="*****"/>
User Type:	<input type="radio"/> Other <input type="radio"/> Professor <input checked="" type="radio"/> Student

127.0.0.1:5000/register

127.0.0.1:5000/login

User registered successfully! Query Executed: INSERT INTO applied_student (roll_number, first_name, middle_name, last_name, email_id, password) VALUES ('21100500', 'Deshi', 'Doshi', 'Doshi', 'deshidoshi@tgt.ac.in', '\$2b\$10\$PZDNe5eG58qfRjAyg4LcLgkHt6yfzCwvycQWu')

OK

Fig. Successful Execution of INSERT

UPDATE

```

main_project > main.app.py > change_password
590     @app.route('/student/change_password', methods=['GET', 'POST'])
591     def change_password():
592         if "roll_number" in session:
593             if request.method == 'POST':
594                 roll_number = session['roll_number']
595                 cursor = db.connection.cursor()
596                 old_password = request.form.get('old_password')
597                 new_password = request.form.get('new_password')
598                 confirm_password = request.form.get('confirm_password')
599                 cursor.execute(f"SELECT password FROM applied_student WHERE roll_number = {roll_number}")
600                 hashed_password = cursor.fetchone()[0]
601                 cursor.close()
602                 if bcrypt.check_password_hash(hashed_password, old_password):
603                     if new_password == confirm_password:
604                         new_hashed_password = bcrypt.generate_password_hash(new_password).decode("utf-8")
605                         cursor = db.connection.cursor()
606                         cursor.execute(f"UPDATE applied_student SET password = '{new_hashed_password}'")
607                         db.connection.commit()
608                         cursor.close()
609                         return redirect(url_for('student_personal_info'))
610                     else:
611                         return redirect(url_for('errorpage'))
612                 else:
613                     return redirect(url_for('errorpage'))
614             return render_template('student/change_password.html')
615         else:
616             return redirect(url_for('errorpage'))
617

```

Fig. Example of UPDATE in change_password in Student

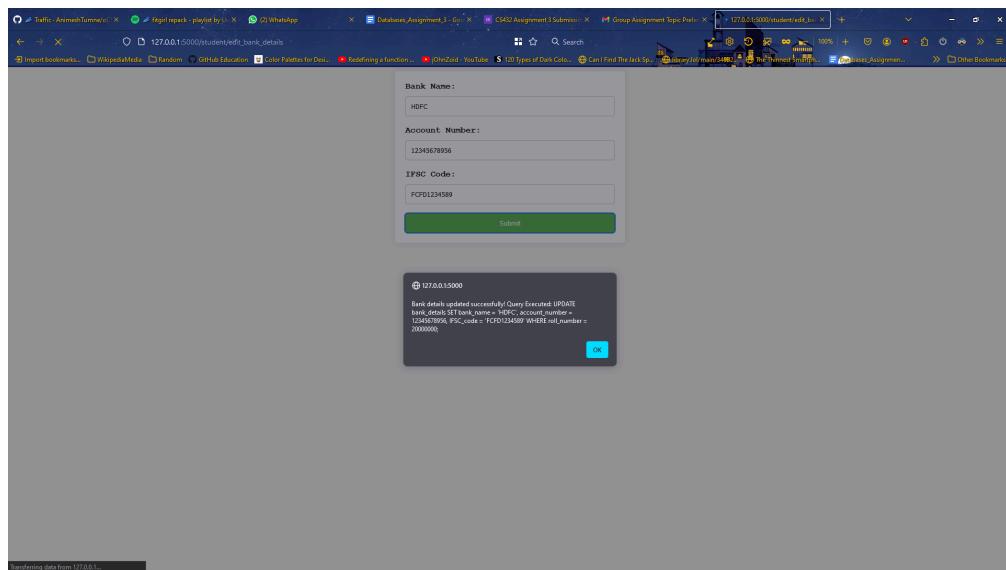


Fig.: Successful Execution of UPDATE

DELETE

```

90 def student_personal_info_change():
91     cursor.execute(f"UPDATE applied_student SET first_name = '{first_name}', middle_name = '{middle_name}',")
92     db.connection.commit()
93
94     if 'image' in request.files:
95         image = request.files['image']
96         image_data = image.read()
97         cursor.execute(f"UPDATE applied_student SET image = {image_data} WHERE roll_number = {roll_number};")
98         db.connection.commit()
99
100    # check if phone number exists
101    if phone_num_exist(roll_number):
102        # update existing phone numbers
103        delete_query = f"DELETE FROM applied_student_phone WHERE roll_number={roll_number};"
104        cursor.execute(delete_query)
105        db.connection.commit()
106
107    # add new phone numbers
108    cursor.execute(f"INSERT INTO applied_student_phone (roll_number, phone_number) VALUES ({roll_number}, {phone_number});")
109    db.connection.commit()
110
111    if phone_2!="":
112        cursor.execute(f"INSERT INTO applied_student_phone (roll_number, phone_number) VALUES ({roll_number}, {phone_2});")
113        db.connection.commit()
114
115
116    db.connection.commit()
117    cursor.close()
118
119
120    return redirect(url_for('student_personal_info'))
121
122
123    # fetch all data
124    cursor = db.connection.cursor()
125    roll_number = session['roll_number']
126    cursor.execute(f"SELECT * FROM applied_student WHERE roll_number = {roll_number};")

```

Fig. Use of DELETE in student_personal_info_change

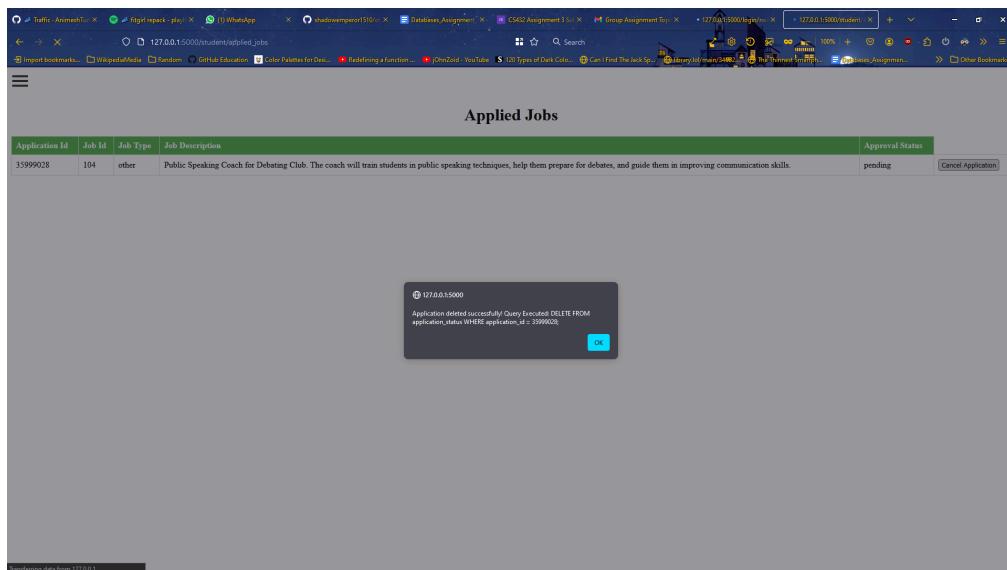


Fig. Successful Execution of DELETE on cancel application

RENAME

```

def index():

    passwordRegex = r'^[!@#$%^&*][a-zA-Z0-9!@#$%^&*]{8,}$'

    passwords = []
    for _ in range(70):
        password = generate_random_password(passwordRegex)
        hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
        passwords.append("'" + hashed_password + "'")

    cur = db.connection.cursor()

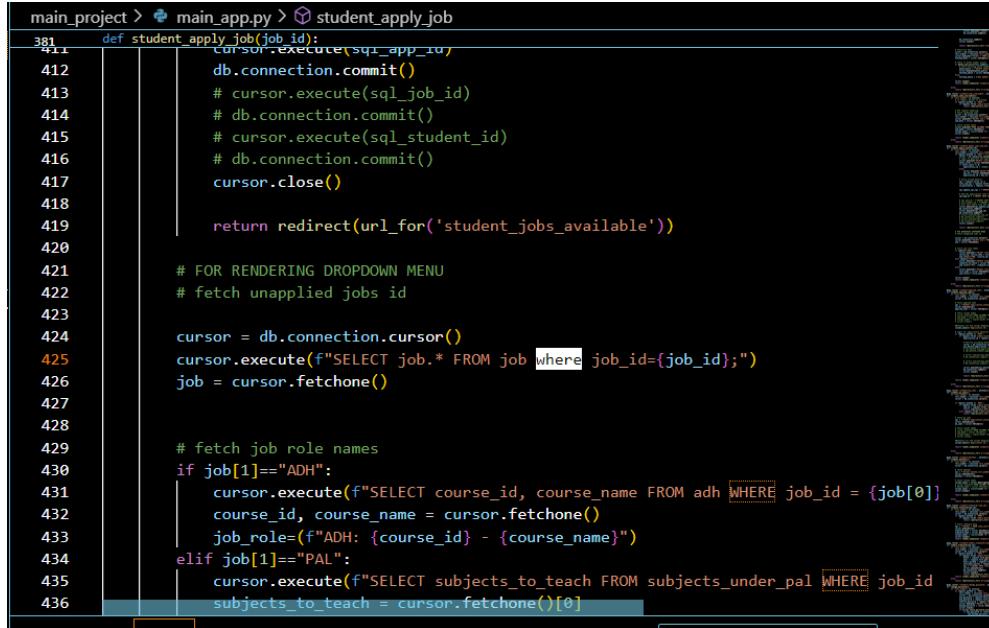
    # added query to add password column in applied_student table
    query_add_column_password = "ALTER TABLE faculty ADD COLUMN password VARCHAR(255);"
    cur.execute(query_add_column_password)
    # rename column Approved to approval
    query_rename_approved_column = "ALTER TABLE application_status RENAME COLUMN Approved TO approval;"
    cur.execute(query_rename_approved_column)

    query_fetch_roll_numbers = "SELECT faculty_id FROM faculty;"
    cur.execute(query_fetch_roll_numbers)
    roll_numbers = [int(row[0]) for row in cur.fetchall()]

```

Fig Use of Rename during Inserting the encrypted password

□ WHERE clause



```

main_project > main_app.py > student_apply.job
381     def student_apply_job(job_id):
382         cursor.execute(sql_app_id)
383         db.connection.commit()
384         # cursor.execute(sql_job_id)
385         # db.connection.commit()
386         # cursor.execute(sql_student_id)
387         # db.connection.commit()
388         cursor.close()
389
390         return redirect(url_for('student_jobs_available'))
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

```

The code shows a Python function `student_apply_job` that performs a database query. It uses a `WHERE` clause in a `SELECT` statement to filter results based on the `job_id` parameter.

Fig. Example of use of WHERE in student_apply_job

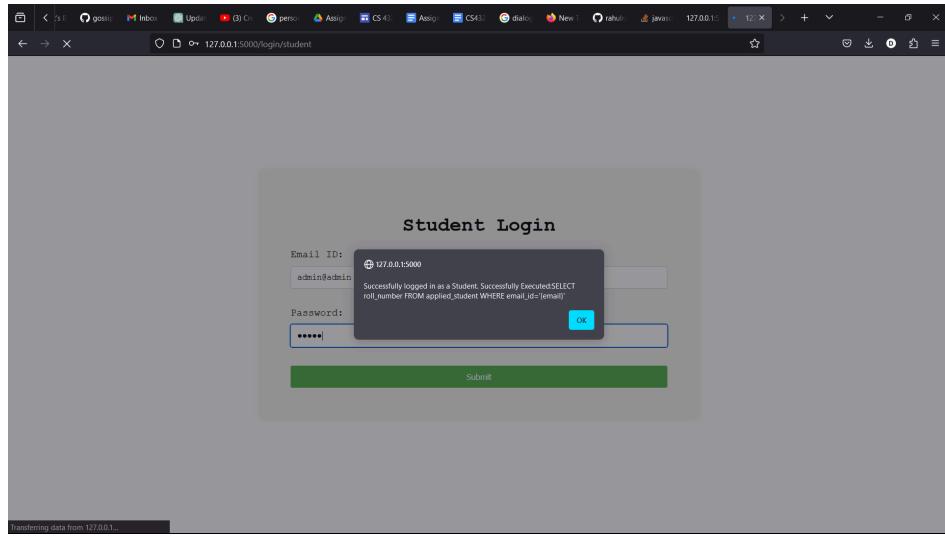


Fig. Successful execution of WHERE clause

Contributions:

Aditya Gupte:

- Worked in frontend of the Other users part.
- Corresponding relevant functions in the backend.

Balgopal Moharana:

- Worked in Frontend of the "Others" page

Darshi Doshi:

- Worked in the Frontend of the Student page (mainly beautification)
- Worked in creating the pop up messages for successful execution of certain queries.

Shah Faisal Khan:

- Worked in ideation and Backend tasks sequence.
- Experimented with different designs for Frontend.

Pranjal:

- Worked in Backend development of Student and Professor User Type.
- Designed basic HTML templates of Student and Professor Web pages to test the designed Backend Logic.
- Also involved in the beautification of some HTML pages.
- Designed the Home Page of the Web Application.

Rohit Raj:

- Worked in Backend
- Worked with Pranjal in creation of backend for the "Student" user type

Tanish Phopalkar:

- Worked in Backend of the "Others" page

Animesh Tumne:

- Worked in Frontend and Backend
- Wrote initial code for routing and login pages - testing code
- Worked with Pranjal to create routing for the "Professor" user type, and relevant frontend for it
- Brainstormed and added various pages/buttons that reflect onto "Professor" pages, for easy and relevant access to the database