

1. '시작화면' UI 구현하기

2. '시작화면' 방향키 입력 시 메뉴 하이라이팅, 종료 구현
2-1) JOIN 하이라이팅

2-2) SIGN IN 하이라이팅



2-3) EXIT 하이라이팅



2-4) SIGN OUT 하이라이팅



3. '시작화면' -> '회원가입 화면' 구현



4-1. '회원 가입 화면' - 방향키, ID, PW 입력, 종료 구현

1) 방향키 구현 - ID



1) 방향키 구현 - Duplicate check



1) 방향키 구현 - PW



1) 방향키 구현 - SIGN IN



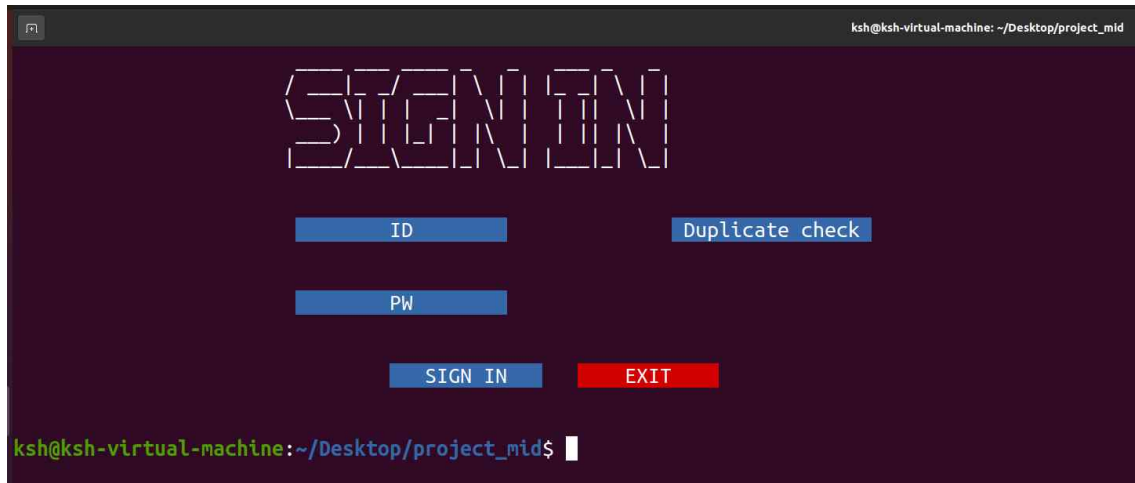
1) 방향키 구현 - EXIT



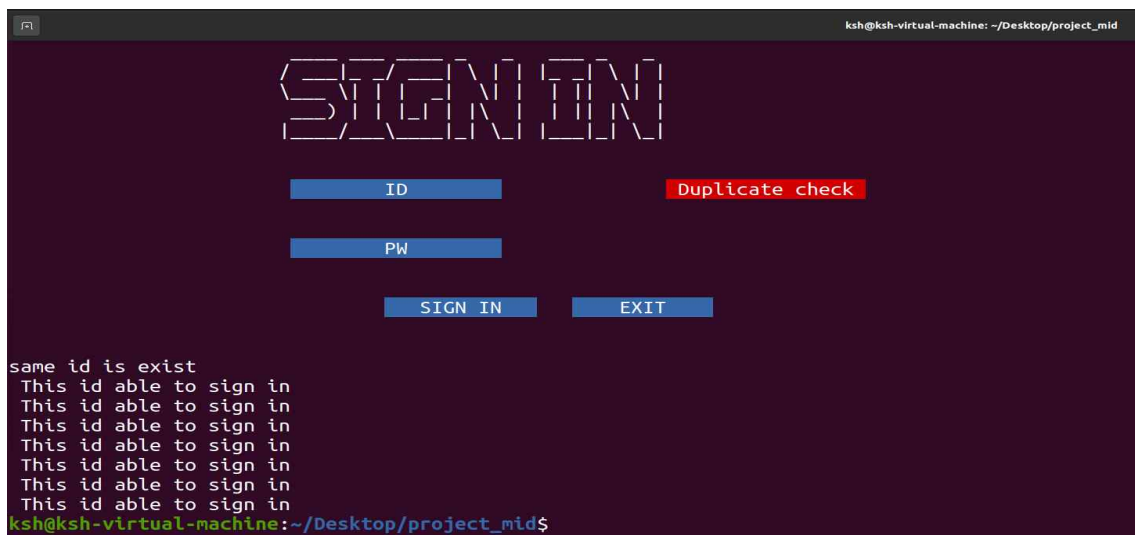
4-2. '회원 가입 화면' - 계정 생성 구현



EXIT 하이라이팅



4-3. '회원 가입 화면' - ID 중복 체크 구현



5. '시작화면' -> '회원 탈퇴 화면' UI 구현



6. '회원 탈퇴 화면' 기능 구현



7. '시작화면' -> '로그인 화면' UI 구현



ID 하이라이팅



PW 하이라이팅



LOG IN 하이라이팅



LOG IN을 ENTER 치면 2P LOGIN 화면으로 넘어간다.



ID 하이라이팅



PW 하이라이팅



LOG IN 하이라이팅



SUCCESS 화면 출력



8-1. '로그인 화면' - 1p 로그인 구현





8-2. '로그인 화면' - 2p 로그인 구현 및 성공 화면 구현



[자세한 코드]

1. '시작화면' UI 구현하기

[illegible]

명령어 echo를 사용하여 화면을 출력하였다.

```
main_opt()
{
    echo -e "\t\t\t\033[44m JOIN \033[0m" "\t\t\033[44m SIGN IN \033[0m"
    echo -e "\n"
    echo -e "\t\t\t\033[44m EXIT \033[0m" "\t\t\033[44m SIGN OUT \033[0m"
}
```

- 하이라이팅이 되지 않는 가장 default 값을 지정한 main_opt함수를 만들어 사용하였다.
- 배경색 44번으로 설정하여 default값이 파란색으로 나오게 코드를 작성했다.

2. '시작화면' 방향키 입력 시 메뉴 하이라이팅, 종료 구현

```
main_opt1()
{
    echo -e "\t\t\t\033[41m JOIN \033[0m" "\t\t\033[44m SIGN IN \033[0m"
    echo -e "\n"
    echo -e "\t\t\t\033[44m EXIT \033[0m" "\t\t\033[44m SIGN OUT \033[0m"
}

main_opt2()
{
    echo -e "\t\t\t\033[44m JOIN \033[0m" "\t\t\033[41m SIGN IN \033[0m"
    echo -e "\n"
    echo -e "\t\t\t\033[44m EXIT \033[0m" "\t\t\033[44m SIGN OUT \033[0m"
}

main_opt3()
{
    echo -e "\t\t\t\033[44m JOIN \033[0m" "\t\t\033[44m SIGN IN \033[0m"
    echo -e "\n"
    echo -e "\t\t\t\033[41m EXIT \033[0m" "\t\t\033[44m SIGN OUT \033[0m"
}

main_opt4()
{
    echo -e "\t\t\t\033[44m JOIN \033[0m" "\t\t\033[44m SIGN IN \033[0m"
    echo -e "\n"
    echo -e "\t\t\t\033[44m EXIT \033[0m" "\t\t\033[41m SIGN OUT \033[0m"
}
```

순서대로 JOIN, SIGN IN, EXIT, SIGN OUT을 하이라이팅 하는 함수 main_opt1, main_opt2, main_opt3, main_opt4를 만들어 사용하였다.

상, 하, 좌, 우를 입력받는 input_key 함수를 만든다.

```
301 input_key()
302 {
303     read -n 1 input
304     if [ $input == '^[' ]; then
305         read -n 1 input
306         if [ $input == '[' ]; then
307             read -n 1 input
308             if [ $input == 'A' ]; then
309                 re=-2
310             elif [ $input == 'B' ]; then
311                 re=2
312             elif [ $input == 'C' ]; then
313                 re=1
314             elif [ $input == 'D' ]; then
315                 re=-1
316             fi
317         fi
318     elif [ $input == `` ]; then
319         re=0
320     fi
321 }
```

```

323 main_page_exit()
324 {
325     now_dir=-2
326     count=0
327     while [ "$count" -ne 1 ]
328     do
329         input_key
330         number=$re
331         if [ "$number" -eq -2 ]; then
332             temp1=`expr $now_dir - 2`
333             now_dir=`expr $temp1 % 4`
334         elif [ "$number" -eq 2 ]; then
335             temp1=`expr $now_dir + 2`
336             now_dir=`expr $temp1 % 4`
337         elif [ "$number" -eq -1 ]; then
338             temp1=`expr $now_dir - 1`
339             now_dir=`expr $temp1 % 4`
340         elif [ "$number" -eq 1 ]; then
341             temp1=`expr $now_dir + 1`
342             now_dir=`expr $temp1 % 4`
343         fi

```

- input_key에서 입력받은 방향키에서 상: -2, 하: 2, 좌: -1, 우: 1의 값을 넣어 해당하는 if문의 수행할 문장들을 실행시킨다.

- now_dir에는 0 ~ 3까지의 숫자만 들어가게 된다.


```

345         clear
346
347         main_page
348         if [ "$now_dir" -eq 0 ]; then
349             main_opt1
350             if [ "$number" -eq 0 ]; then
351                 clear
352                 player1
353                 player1_opt
354                 player1_exit
355                 count=1
356             fi
357         elif [ "$now_dir" -eq 1 ]; then
358             main_opt2
359             if [ "$number" -eq 0 ]; then
360                 clear
361                 sign_in
362                 sign_opt
363                 sign_in_exit
364                 count=1
365             fi
366         elif [ "$now_dir" -eq 2 ]; then
367             main_opt3
368             if [ "$number" -eq 0 ]; then
369                 count=1
370             fi
371         elif [ "$now_dir" -eq 3 ]; then
372             main_opt4
373             if [ "$number" -eq 0 ]; then
374                 clear
375                 sign_out
376                 sign_out_opt
377                 sign_out_exit
378                 count=1
379             fi
380         else
381             main_opt
382         fi
383     done
384 }

```

- now_dir은 현재 커서가 위치한 자리이다. 커서의 자리가 상대적이므로, 위치값을 고정시켜 해당 위치에 커서가 있으면 if문을 수행하는 코드를 구현하였다.
- 323 - 384까지 main_page_exit 함수를 구현하였다.

3. '시작화면' -> '회원가입 화면' 구현

```

56 sign_in()
57 {
58     echo -e "
59     echo -e "
60     echo -e "
61     echo -e "
62     echo -e "
63     echo -e "\n"
64 }

```

```

SIGN IN

```

```

66 sign_opt()
67 {
68     echo -e "\t\t\t\033[44m          ID          \033[0m" "\t\t\033[44m Duplicate check \033[0m"
69     echo -e "\n"
70     echo -e '\t\t\t\033[44m          PW          \033[0m'
71     echo -e "\n"
72     echo -e "\t\t\t\t\033[44m SIGN IN  \033[0m" " \t\033[44m EXIT  \033[0m"
73     echo -e "\n"
74 }

```

- echo 명령어를 통해 화면에 출력하였다.

4-1. '회원 가입 화면' - 방향키, ID, PW 입력, 종료 구현

4-2. '회원 가입 화면' - 계정 생성 구현

```
76 sign_opt1()
77 {
78     echo -e "\t\t\t\033[41m          ID          \033[0m" "\t\t\033[44m Duplicate check \033[0m"
79     echo -e "\n"
80     echo -e '\t\t\t\033[44m          PW          \033[0m'
81     echo -e "\n"
82     echo -e "\t\t\t\t\t\033[44m SIGN IN  \033[0m" " \t\033[44m EXIT  \033[0m"
83     echo -e "\n"
84 }
85
86 sign_opt2()
87 {
88     echo -e "\t\t\t\033[44m          ID          \033[0m" "\t\t\033[41m Duplicate check \033[0m"
89     echo -e "\n"
90     echo -e '\t\t\t\033[44m          PW          \033[0m'
91     echo -e "\n"
92     echo -e "\t\t\t\t\t\033[44m SIGN IN  \033[0m" " \t\033[44m EXIT  \033[0m"
93     echo -e "\n"
94 }
95
96 sign_opt3()
97 {
98     echo -e "\t\t\t\033[44m          ID          \033[0m" "\t\t\033[44m Duplicate check \033[0m"
99     echo -e "\n"
100    echo -e '\t\t\t\033[41m          PW          \033[0m'
101    echo -e "\n"
102    echo -e "\t\t\t\t\t\033[44m SIGN IN  \033[0m" " \t\033[44m EXIT  \033[0m"
103    echo -e "\n"
104 }
105
106 sign_opt4()
107 {
108     echo -e "\t\t\t\033[44m          ID          \033[0m" "\t\t\033[44m Duplicate check \033[0m"
109     echo -e "\n"
110     echo -e '\t\t\t\033[44m          PW          \033[0m'
111     echo -e "\n"
112     echo -e "\t\t\t\t\t\033[41m SIGN IN  \033[0m" " \t\033[44m EXIT  \033[0m"
113     echo -e "\n"
114 }
115
116 sign_opt5()
117 {
118     echo -e "\t\t\t\033[44m          ID          \033[0m" "\t\t\033[44m Duplicate check \033[0m"
119     echo -e "\n"
120     echo -e '\t\t\t\033[44m          PW          \033[0m'
121     echo -e "\n"
122     echo -e "\t\t\t\t\t\033[44m SIGN IN  \033[0m" " \t\033[41m EXIT  \033[0m"
123     echo -e "\n"
124 }
```

- 명령어 echo를 사용하여 화면을 출력하였다.
- 순서대로 ID, Duplicate check, PW, SIGN IN, EXIT을 하이라이팅 하는 함수를 sign_opt1 ~ sign_opt5까지 구현하였다.


```

412 sign_in_exit()
413 {
414     now_dir=-2
415     count=0
416     while [ $count -ne 1 ]
417     do
418         input_key
419         number=$re
420         if [ "$number" -eq -2 ]; then
421             temp1=`expr $now_dir - 2`
422             if [ "$temp1" -ge 4 ]; then
423                 now_dir=`expr $temp1 % 6`
424             else
425                 now_dir=`expr $temp1 % 4`
426             fi
427         elif [ "$number" -eq 2 ]; then
428             temp1=`expr $now_dir + 2`
429             if [ "$temp1" -ge 4 ]; then
430                 now_dir=`expr $temp1 % 6`
431             else
432                 now_dir=`expr $temp1 % 4`
433             fi
434         elif [ "$number" -eq -1 ]; then
435             temp1=`expr $now_dir - 1`
436             if [ "$temp1" -ge 4 ]; then
437                 now_dir=`expr $temp1 % 6`
438             else
439                 now_dir=`expr $temp1 % 4`
440             fi
441         elif [ "$number" -eq 1 ]; then
442             temp1=`expr $now_dir + 1`
443             if [ "$temp1" -ge 4 ]; then
444                 now_dir=`expr $temp1 % 6`
445             else
446                 now_dir=`expr $temp1 % 4`
447             fi
448         fi

```

now_dir은 현재 커서가 위치한 자리이다. 커서의 자리가 상대적이므로, 위치값을 고정시켜 해당 위치에 커서가 있으면 if문을 수행하는 코드를 구현하였다.

```

451 sign_in
452
453 if [ "$now_dir" -eq 0 ]; then
454     sign_opt1
455     if [ "$number" -eq 0 ]; then
456         read_in_id
457         input_key
458         enter=$re
459         if [ "$enter" -eq 0 ]; then
460             clear
461             sign_in
462             echo -e "\t\t\t\033[41m          '$in_id'          \033[0m" "\t\t\t\033[44m Duplicate check \033[0m"
463             echo -e "\n"
464             echo -e '\t\t\t\033[44m          PW          \033[0m'
465             echo -e "\n"
466             echo -e "\t\t\t\t\033[44m SIGN IN \033[0m" " \t\033[44m EXIT \033[0m"
467             echo -e "\n"
468         fi
469     fi
470     elif [ "$now_dir" -eq 1 ]; then
471         sign_opt2
472         if [ "$number" -eq 0 ]; then
473             duplicate_check
474             count=1
475         fi
476     elif [ "$now_dir" -eq 2 ]; then
477         sign_opt3
478         if [ "$number" -eq 0 ]; then
479             read_in_pw
480             input_key
481             enter=$re
482             if [ "$enter" -eq 0 ]; then
483                 clear
484                 sign_in
485                 echo -e "\t\t\t\033[41m          '$in_id'          \033[0m" "\t\t\t\033[44m Duplicate check \033[0m"
486                 echo -e "\n"
487                 echo -e "\t\t\t\033[41m          '$in_pw'          \033[0m"
488                 echo -e "\n"
489                 echo -e "\t\t\t\t\033[44m SIGN IN \033[0m" " \t\033[44m EXIT \033[0m"
490                 echo -e "\n"
491             fi
492         fi

```

지금 현재 위치(now_dir)이 0(ID)일 때, 1(Duplicate check)일 때, 2(PW)일 때, 해당 위치에 커서가 있으면 if문을 수행하는 코드를 구현하였다.

위치가 ID에 있을 때, PW에 있을 때, in_id, in_pw값을 받아 해당 변수를 echo 명령어를 사용하여 알맞게 출력한다.

```

493         elif [ "$now_dir" -eq 3 ]; then
494             sign_opt3
495             now_dir=2
496         elif [ "$now_dir" -eq 4 ]; then
497             sign_opt4
498             if [ "$number" -eq 0 ]; then
499                 store="$in_id","$in_pw",0,0
500                 echo -e "$store" >> ./auth.txt
501                 count=1
502             fi
503         elif [ "$now_dir" -eq 5 ]; then
504             sign_opt5
505             if [ "$number" -eq 0 ]; then
506                 count=1
507             fi
508         else
509             sign_opt
510         fi
511     done
512 }

```

- 지금 현재 위치(now_dir)이 3(PW)일 때, 4(SIGN IN)일 때, 5(EXIT)일 때, 해당 위치에 커서가 있으면 if문을 수행하는 코드를 구현하였다.
- 3의 위치는 2의 위치와 같지만 방향키 입력을 맞게 하기 위해 임시로 설정한 위치이고 항상 now_dir(현재 커서 위치)를 2로 변경한다.
- 4(SIGN IN)일 때, auth.txt 파일에 해당 0번위치, 2번위치에서 적었던 in_id, in_pw 값을 저장한다.
- 5(EXIT)일 때, while문의 조건을 거짓이 되게 만들기 위해 count=1을 넣어 반복문을 그만 수행하도록 코드를 구현했다. 따라서, 프로그램 종료가 된다.

4-3. '회원 가입 화면' - ID 중복 체크 구현

```
386 duplicate_check()
387 {
388     file=`cat ./auth.txt`
389     line=`echo $file`
390     database=`echo $line | tr ",," "\n"`
391
392     time=0
393     for data in $database
394     do
395         odd_even=`expr $time % 2`
396         if [ "$in_id" == "$data" ] && [ "$odd_even" -eq 0 ]; then
397             same_id=0
398             time=`expr $time + 1`
399         else
400             same_id=1
401             time=`expr $time + 1`
402         fi
403
404         if [ "$same_id" -eq 0 ]; then
405             echo "same id is exist"
406         else
407             echo " This id able to sign in"
408         fi
409     done
410 }
```

- database 변수에 "id, pw, 승수, 패수"가 순서대로 원소로 들어갈 수 있게 만든다.
변수 time은 총 data의 개수이고 이를 %2를 사용하여 0일때는, id값만 비교할 수 있게 하고, 1일때는, pw값만 비교할 수 있게하였다.
- 그래서 같은 id값이 들어있으면 "same id is exist"를 출력, 같은 id값이 들어있지 않으면, "This id able to sign in"을 출력한다.

5. '시작화면' -> '회원 탈퇴 화면' UI 구현

```

126 sign_out()
127 {
128     echo -e "
129     echo -e "
130     echo -e "
131     echo -e "
132     echo -e "
133     echo -e "\n"
134 }
135
136 sign_out_opt()
137 {
138     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
139     echo -e "\n"
140     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
141     echo -e "\n"
142     echo -e "\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\033[44m EXIT \033[0m"
143 }
144
145 sign_out_opt1()
146 {
147     echo -e "\t\t\t\t\t\033[41m          ID          \033[0m"
148     echo -e "\n"
149     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
150     echo -e "\n"
151     echo -e "\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\033[44m EXIT \033[0m"
152 }
153
154 sign_out_opt2()
155 {
156     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
157     echo -e "\n"
158     echo -e "\t\t\t\t\t\033[41m          PW          \033[0m"
159     echo -e "\n"
160     echo -e "\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\033[44m EXIT \033[0m"
161 }
162
163 sign_out_opt3()
164 {
165     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
166     echo -e "\n"
167     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
168     echo -e "\n"
169     echo -e "\t\t\t\t \033[41m SIGN OUT \033[0m" "\t\t\033[44m EXIT \033[0m"
170 }
171
172 sign_out_opt4()
173 {
174     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
175     echo -e "\n"
176     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
177     echo -e "\n"
178     echo -e "\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\033[41m EXIT \033[0m"
179 }

```

- echo 명령어를 사용하여 sign_out, sign_out_opt 함수를 만들어 SIGN OUT UI를 구현하였다.
- ID, PW, SIGN OUT, EXIT 차례로 하이라이팅 되는 함수를 sign_out_opt1 ~ sign_out_opt4를 만들었다.

6. '회원 탈퇴 화면' 기능 구현

```
514 sign_out_exit()
515 {
516     now_dir=-2
517     count=0
518     while [ "$count" -ne 1 ]
519     do
520         input_key
521         number=$re
522         if [ "$number" -eq -2 ]; then
523             temp1=`expr $now_dir - 2`
524             if [ "$temp1" -ge 4 ]; then
525                 now_dir=`expr $temp1 % 5`
526             else
527                 now_dir=`expr $temp1 % 4`
528             fi
529         elif [ "$number" -eq 2 ]; then
530             temp1=`expr $now_dir + 2`
531             if [ "$temp1" -ge 4 ]; then
532                 now_dir=`expr $temp1 % 5`
533             else
534                 now_dir=`expr $temp1 % 4`
535             fi
536         elif [ "$number" -eq -1 ]; then
537             temp1=`expr $now_dir - 1`
538             if [ "$temp1" -ge 4 ]; then
539                 now_dir=`expr $temp1 % 5`
540             else
541                 now_dir=`expr $temp1 % 4`
542             fi
543         elif [ "$number" -eq 1 ]; then
544             temp1=`expr $now_dir + 1`
545             if [ "$temp1" -ge 5 ]; then
546                 now_dir=`expr $temp1 % 6`
547             else
548                 now_dir=`expr $temp1 % 4`
549             fi
550         fi
551     done
552     clear

554     sign_out
555
556     file=`cat ./auth.txt`
557     line=`echo $file`
558     database=`echo $line | tr "," "\n"`
559
560     if [ "$now_dir" -eq 0 ]; then
561         sign_out_opt1
562         if [ "$number" -eq 0 ]; then
563             read out_id
564             input_key
565             enter1=$re
566             if [ "$enter1" -eq 0 ]; then
567                 clear
568                 sign_out
569                 echo -e "\t\t\t\t\t\033[41m          "$out_id"          \033[0m"
570                 echo -e "\n"
571                 echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
572                 echo -e "\n"
573                 echo -e "\t\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\t\033[44m EXIT \033[0m"
574             fi
575         fi
576     elif [ "$now_dir" -eq 2 ]; then
577         sign_out_opt2
578         if [ "$number" -eq 0 ]; then
579             read out_pw
580             input_key
581             enter1=$re
582             if [ "$enter1" -eq 0 ]; then
583                 clear
584                 sign_out
585                 echo -e "\t\t\t\t\t\033[41m          "$out_id"          \033[0m"
586                 echo -e "\n"
587                 echo -e "\t\t\t\t\t\033[41m          "$out_pw"          \033[0m"
588                 echo -e "\n"
589                 echo -e "\t\t\t\t\t \033[44m SIGN OUT \033[0m" "\t\t\t\033[44m EXIT \033[0m"
590             fi
591         fi
592     fi
```



```

591     fi
592     elif [ "$snow_dir" -eq 4 ]; then
593         sign_out_opt3
594         if [ "$number" -eq 0 ]; then
595             range=0
596             remove_id=1
597             remove_pw=1
598
599             for data in $database
600             do
601                 id0=`expr $range % 4`
602                 id0=`expr $range % 2`
603                 pw1=`expr $range % 4`
604                 pw1=`expr $range % 2`
605                 if [ "$remove_id" -eq 0 ] && [ "$remove_pw" -eq 0 ]; then
606                     remove=5
607                     echo 1
608                     break
609                 else
610                     remove=99
611                 fi
612
613                 if [ "$id0" -eq 0 ] && [ "$out_id" = "$data" ]; then
614                     remove_id=0
615                     range=`expr "$range" + 1`
616                     temp=$out_id
617                     echo 2
618                 elif [ "$remove_id" -eq 0 ] && [ "$pw1" -eq 1 ] && [ "$out_pw" = "$data" ]; then
619                     remove_pw=0
620                     range=`expr "$range" + 1`
621                     echo 3
622                 else
623                     range=`expr "$range" + 1`
624                     echo 4
625                 fi
626             done
627             if [ "$remove" -eq 5 ]; then
628                 sed -i '/'$d'/d' ./auth.txt
629                 count=1
630             else
631                 count=1
632             fi
633         fi
634     elif [ "$snow_dir" -eq 5 ]; then
635         player1_opt4
636         now_dir=4
637         if [ "$number" -eq 0 ]; then
638             count=1
639         fi
640     else
641         player1_opt
642     fi
643 fi
644 done
645 }

```

- for문으로 data에 순차적으로 id, pw, 승수, 피수를 넣어가면서 해당 if문의 실행식을 실행한다.
- 해당 if문의 실행문을 따라 들어가 지우고자 하는 id, pw값을 찾는다.
- 해당하는 id, pw값을 찾으면 break를 사용하여 for문을 빠져나와 if문과 명령어 sed를 사용하여 해당 줄을 삭제한다.
- count=1을 하여 프로그램을 종료시킨다.

7. '시작화면' -> '로그인 화면' UI 구현

```

181 player1()
182 {
183     echo -e "
184     echo -e "
185     echo -e "
186     echo -e "
187     echo -e "
188     echo -e "\n"
189 }
191 player1_opt()
192 {
193     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
194     echo -e "\n"
195     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
196     echo -e "\n"
197     echo -e "\t\t\t\t\t\033[44m  LOGIN  \033[0m" "\t\t\t\t\t\033[44m  EXIT  \033[0m"
198 }
199
200 player1_opt1()
201 {
202     echo -e "\t\t\t\t\t\033[41m          ID          \033[0m"
203     echo -e "\n"
204     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
205     echo -e "\n"
206     echo -e "\t\t\t\t\t\033[44m  LOGIN  \033[0m" "\t\t\t\t\t\033[44m  EXIT  \033[0m"
207 }
208
209 player1_opt2()
210 {
211     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
212     echo -e "\n"
213     echo -e "\t\t\t\t\t\033[41m          PW          \033[0m"
214     echo -e "\n"
215     echo -e "\t\t\t\t\t\033[44m  LOGIN  \033[0m" "\t\t\t\t\t\033[44m  EXIT  \033[0m"
216 }
217
218 player1_opt3()
219 {
220     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
221     echo -e "\n"
222     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
223     echo -e "\n"
224     echo -e "\t\t\t\t\t\033[41m  LOGIN  \033[0m" "\t\t\t\t\t\033[44m  EXIT  \033[0m"
225 }
226
227 player1_opt4()
228 {
229     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
230     echo -e "\n"
231     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
232     echo -e "\n"
233     echo -e "\t\t\t\t\t\033[44m  LOGIN  \033[0m" "\t\t\t\t\t\033[41m  EXIT  \033[0m"
234 }

```

echo 명령어를 사용하여 player1 함수와 player1_opt 함수를 만들었고, 순서대로 ID, PW, LOGIN, EXIT 하이라이팅 함수를 player1_opt1 ~ player1_opt4까지 만들었다.

```

236 player2()
237 {
238     echo -e "
239     echo -e "
240     echo -e "
241     echo -e "
242     echo -e "
243     echo -e "\n"
244 }
...
246 player2_opt()
247 {
248     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
249     echo -e "\n"
250     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
251     echo -e "\n"
252     echo -e "\t\t\t\t \033[44m  LOGIN  \033[0m" "\t\t\033[44m  EXIT  \033[0m"
253 }
254
255 player2_opt1()
256 {
257     echo -e "\t\t\t\t\t\033[41m          ID          \033[0m"
258     echo -e "\n"
259     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
260     echo -e "\n"
261     echo -e "\t\t\t\t \033[44m  LOGIN  \033[0m" "\t\t\033[44m  EXIT  \033[0m"
262 }
263
264 player2_opt2()
265 {
266     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
267     echo -e "\n"
268     echo -e "\t\t\t\t\t\033[41m          PW          \033[0m"
269     echo -e "\n"
270     echo -e "\t\t\t\t \033[44m  LOGIN  \033[0m" "\t\t\033[44m  EXIT  \033[0m"
271 }
272
273 player2_opt3()
274 {
275     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
276     echo -e "\n"
277     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
278     echo -e "\n"
279     echo -e "\t\t\t\t \033[41m  LOGIN  \033[0m" "\t\t\033[44m  EXIT  \033[0m"
280 }
281
282 player2_opt4()
283 {
284     echo -e "\t\t\t\t\t\033[44m          ID          \033[0m"
285     echo -e "\n"
286     echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
287     echo -e "\n"
288     echo -e "\t\t\t\t \033[44m  LOGIN  \033[0m" "\t\t\033[41m  EXIT  \033[0m"
289 }

```

- echo 명령어를 사용하여 player1 함수와 player1_opt 함수를 만들었다.
- 순서대로 ID, PW, LOGIN, EXIT 하이라이팅 함수를 player2_opt1 ~ player2_opt4까지 만들었다.

8-1. '로그인 화면' - 1p 로그인 구현

```
640 player1_exit()
641 {
642     now_dir=-2
643     count=0
644     while [ "$count" -ne 1 ]
645     do
646         input_key
647         number=$re
648         if [ "$number" -eq -2 ]; then
649             temp1=`expr $now_dir - 2`
650             if [ "$temp1" -ge 4 ]; then
651                 now_dir=`expr $temp1 % 5`
652             else
653                 now_dir=`expr $temp1 % 4`
654             fi
655         elif [ "$number" -eq 2 ]; then
656             temp1=`expr $now_dir + 2`
657             if [ "$temp1" -ge 4 ]; then
658                 now_dir=`expr $temp1 % 5`
659             else
660                 now_dir=`expr $temp1 % 4`
661             fi
662         elif [ "$number" -eq -1 ]; then
663             temp1=`expr $now_dir - 1`
664             if [ "$temp1" -ge 4 ]; then
665                 now_dir=`expr $temp1 % 5`
666             else
667                 now_dir=`expr $temp1 % 4`
668             fi
669         elif [ "$number" -eq 1 ]; then
670             temp1=`expr $now_dir + 1`
671             if [ "$temp1" -ge 5 ]; then
672                 now_dir=`expr $temp1 % 6`
673             else
674                 now_dir=`expr $temp1 % 4`
675             fi
676         fi
677     done
678 }
679
680 player1
681
682 file=`cat ./auth.txt`
683 line=`echo $file`
684 database=`echo $line | tr "," "\n"`
685
686 if [ "$now_dir" -eq 0 ]; then
687     player1_opt1
688     if [ "$number" -eq 0 ]; then
689         read player1_id
690         input_key
691         enter=$re
692         if [ "$enter" -eq 0 ]; then
693             clear
694             player1
695             echo -e "\t\t\t\t\t\033[41m          "$player1_id          \033[0m"
696             echo -e "\n"
697             echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
698             echo -e "\n"
699             echo -e "\t\t\t\t \033[44m LOGIN \033[0m" "\t\t\033[44m EXIT \033[0m"
700         fi
701     fi
702     elif [ "$now_dir" -eq 2 ]; then
703         player1_opt2
704         if [ "$number" -eq 0 ]; then
705             read player1_pw
706             input_key
707             enter=$re
708             if [ "$enter" -eq 0 ]; then
709                 clear
710                 player1
711                 echo -e "\t\t\t\t\t\033[41m          "$player1_id          \033[0m"
712                 echo -e "\n"
713                 echo -e "\t\t\t\t\t\033[41m          "$player1_pw          \033[0m"
714                 echo -e "\n"
715                 echo -e "\t\t\t\t \033[44m LOGIN \033[0m" "\t\t\033[44m EXIT \033[0m"
716             fi
717         fi
718     fi
719 }
```

```

718     elif [ "$now_dir" -eq 4 ]; then
719         player1_opt3
720         if [ "$number" -eq 0 ]; then
721             range=0
722             isId=1
723             isPw=1
724
725             for data in $database
726             do
727                 id0=`expr "$range" % 4`
728                 id0=`expr $range % 2`
729                 pw1=`expr "$range" % 4`
730                 pw1=`expr $range % 2`
731                 if [ "$isId" -eq 0 ] && [ "$isPw" -eq 0 ]; then
732                     login=5
733                     break
734                 else
735                     login=99
736                 fi
737
738                 if [ "$id0" -eq 0 ] && [ "$player1_id" = "$data" ]; then
739                     isId=0
740                     range=`expr "$range" + 1`
741                     elif [ "$isId" -eq 0 ] && [ "$pw1" -eq 1 ] && [ "$player1_pw" = "$data" ]; then
742                         isPw=0
743                         range=`expr "$range" + 1`
744                     else
745                         range=`expr "$range" + 1`
746                     fi
747             done
748
749             if [ "$login" -eq 5 ]; then
750                 clear
751                 player2
752                 player2_opt
753                 player2_exit
754             else
755                 count=1
756             fi
757         fi
758     elif [ "$now_dir" -eq 5 ]; then
759         player1_opt4
760         now_dir=4
761         if [ "$number" -eq 0 ]; then
762             count=1
763         fi
764     else
765         player1_opt
766     fi
767 done

```

- 644 ~ 676 줄은 상, 하, 좌, 우 키를 받는 함수 input_key이다.
- 0번째, 2번째 위치에서, input_key 함수를 통해 받은 값을 number에 저장하고 number가 0일 때, 사용자가 enter를 누를 때, player2의 페이지에서 id, pw의 값이 저장된다.
- id가 auth.txt 파일에 있으면 isId값을 바꿔주고, isId값이 0이고 홀수번째에서 pw가 data값과 같으면 isPw값을 변경한다.
- 4번째 위치에서 enter를 눌렀을 때, id와 pw 값이 auth.txt파일에 알맞게 존재하면 2p LOGIN으로 넘어간다.
- EXIT를 누르면 바로 종료되게 만들었다.

8-2. '로그인 화면' - 2p 로그인 구현 및 성공 화면 구현

player2_exit 함수를 통해 하이라이팅과 데이터 저장, 비교 등을 구현했습니다.

```
770 player2_exit()
771 {
772     now_dir=-2
773     count=0
774     while [ "$count" -ne 1 ]
775     do
776         input_key
777         number=$re
778         if [ "$number" -eq -2 ]; then
779             temp1=`expr $now_dir - 2`
780             if [ "$temp1" -ge 4 ]; then
781                 now_dir=`expr $temp1 % 5`
782             else
783                 now_dir=`expr $temp1 % 4`
784             fi
785         elif [ "$number" -eq 2 ]; then
786             temp1=`expr $now_dir + 2`
787             if [ "$temp1" -ge 4 ]; then
788                 now_dir=`expr $temp1 % 5`
789             else
790                 now_dir=`expr $temp1 % 4`
791             fi
792         elif [ "$number" -eq -1 ]; then
793             temp1=`expr $now_dir - 1`
794             if [ "$temp1" -ge 4 ]; then
795                 now_dir=`expr $temp1 % 5`
796             else
797                 now_dir=`expr $temp1 % 4`
798             fi
799         elif [ "$number" -eq 1 ]; then
800             temp1=`expr $now_dir + 1`
801             if [ "$temp1" -ge 5 ]; then
802                 now_dir=`expr $temp1 % 6`
803             else
804                 now_dir=`expr $temp1 % 4`
805             fi
806         fi
807         clear
808
809         player2
810
811         file=`cat ./auth.txt`
812         line=`echo $file`
813         database=`echo $line | tr "," "\n"`
814
815         if [ "$now_dir" -eq 0 ]; then
816             player2_opt1
817             if [ "$number" -eq 0 ]; then
818                 read player2_id
819                 clear
820                 player2
821                 echo -e "\t\t\t\t\t\033[41m          \"$player2_id\"          \033[0m"
822                 echo -e "\n"
823                 echo -e "\t\t\t\t\t\033[44m          PW          \033[0m"
824                 echo -e "\n"
825                 echo -e "\t\t\t\t\t \033[44m LOGIN  \033[0m" "\t\t\t\033[44m EXIT  \033[0m"
826             fi
827         elif [ "$now_dir" -eq 2 ]; then
828             player2_opt2
829             if [ "$number" -eq 0 ]; then
830                 read player2_pw
831                 clear
832                 player2
833                 echo -e "\t\t\t\t\t\033[41m          \"$player2_id\"          \033[0m"
834                 echo -e "\n"
835                 echo -e "\t\t\t\t\t\033[41m          \"$player2_pw\"          \033[0m"
836                 echo -e "\n"
837                 echo -e "\t\t\t\t\t \033[44m LOGIN  \033[0m" "\t\t\t\033[44m EXIT  \033[0m"
838             fi
839         fi
840     done
841 }
```

now_dir이 0일 때, 2일 때, ID, PW값을 입력받는다. 또한, 이를 하이라이팅 되게 echo 명령어를 통해 출력하였다.

```

840     elif [ "$now_dir" -eq 4 ]; then
841         player2_opt3
842         if [ "$number" -eq 0 ]; then
843             range=0
844             isId=1
845             isPw=1
846
847             for data in $database
848             do
849                 id0=`expr "$range" % 4`
850                 id0=`expr $range % 2`
851                 pw1=`expr "$range" % 4`
852                 pw1=`expr $range % 2`
853                 if [ "$isId" -eq 0 ] && [ "$isPw" -eq 0 ]; then
854                     login=5
855                     break
856                 else
857                     login=99
858                 fi
859
860                 if [ "$id0" -eq 0 ] && [ "$player2_id" = "$data" ]; then
861                     isId=0
862                     range=`expr "$range" + 1`
863                 elif [ "$isId" -eq 0 ] && [ "$pw1" -eq 1 ] && [ "$player2_pw" = "$data" ]; then
864                     isPw=0
865                     range=`expr "$range" + 1`
866                 else
867                     isId=1
868                     isPw=1
869                     range=`expr "$range" + 1`
870                 fi
871             done
872             if [ "$login" -eq 5 ]; then
873                 clear
874                 success
875                 count=1
876             else
877                 count=1
878             fi
879         fi
880     elif [ "$now_dir" -eq 5 ]; then
881         player2_opt4
882         now_dir=4
883         if [ "$number" -eq 0 ]; then
884             count=1
885         fi
886     else
887         player2_opt
888     fi
889 done
890 }

```

- 위치가 4일 때, 비교를 해야한다. auth.txt 파일 안에 해당 id가 있고, id에 맞는 pw가 있으면 success 화면을 출력하고 종료한다.
- 0번째, 2번째 위치에서, input_key 함수를 통해 받은 값을 number에 저장하고 number가 0일 때, 사용자가 enter를 누를 때, player2의 페이지에서 id, pw의 값이 저장된다.
- id가 auth.txt 파일에 있으면 isId값을 바꿔주고, isId값이 0이고 홀수번째에서 pw가 data값과 같으면 isPw값을 변경한다.
- EXIT를 누르면 바로 종료되게 만들었다.