

Source Code

```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <time.h>
6 void unused_data2(char data160[])
7 {
8     FILE *original_file160 = fopen("data_statistics_available.txt", "r");
9     FILE *temp_file160 = fopen("data_statistics_temp.txt", "w");
10    if (original_file160 == NULL)
11    {
12        original_file160 = fopen("data_statistics_available.txt", "w");
13        fclose(original_file160);
14        original_file160 = fopen("data_statistics_available.txt", "r");
15    }
16    char line160[10000];
17    int taka160 = +1;
18    while (fgets(line160, sizeof(line160), original_file160))
19    {
20        char money160[100];
21        char no160[100];
22        if (sscanf(line160, "%99s %99s", money160, no160) == 2)
23        {
24            if (!strcmp(money160, data160))
25            {
26                taka160 += atoi(no160);
27                continue;
28            }
29            else if (!strcmp(line160, " ")) continue;
30        }
31        fprintf(temp_file160, line160);
32    }
33    char string_num160[10000];
34    snprintf(string_num160, sizeof(string_num160), "%d", taka160);
35    fprintf(temp_file160, "%s %s\n", data160, string_num160);
36    fclose(original_file160);
37    fclose(temp_file160);
38    remove("data_statistics_available.txt");
39    rename("data_statistics_temp.txt", "data_statistics_available.txt");
40 }
41
42 void unused_card(char data160[])
43 {
44     FILE *original_file160 = fopen("data_statistics_available.txt", "r");
45     FILE *temp_file160 = fopen("data_statistics_temp.txt", "w");
46     if (original_file160 == NULL)
47     {
48         original_file160 = fopen("data_statistics_available.txt", "w");
49         fclose(original_file160);
50         original_file160 = fopen("data_statistics_available.txt", "r");
51     }
52     char line160[10000];
53     int taka160 = -1;
```

```

54     while (fgets(line160, sizeof(line160), original_file160))
55     {
56         char money160[100];
57         char no160[100];
58         if (sscanf(line160, "%99s %99s", money160, no160) == 2)
59         {
60             if (!strcmp(money160, data160))
61             {
62                 taka160 += atoi(no160);
63                 continue;
64             }
65             else if (!strcmp(line160, " ")) continue;
66         }
67         fprintf(temp_file160, line160);
68     }
69     char string_num160[10000];
70     snprintf(string_num160, sizeof(string_num160), "%d", taka160);
71     fprintf(temp_file160, "%s %s\n", data160, string_num160);
72     fclose(original_file160);
73     fclose(temp_file160);
74     remove("data_statistics_available.txt");
75     rename("data_statistics_temp.txt", "data_statistics_available.txt");
76 }
77 void statistics()
78 {
79     printf("Used Card list: ");
80     FILE *file160 = fopen("data_statistics_used.txt", "r");
81     char data160[100];
82     printf("\n");
83     while (fgets(data160, 100, file160))
84     {
85         char taka160[100];
86         char time160[100];
87         sscanf(data160, "%99s %99s", taka160, time160);
88         printf("                %s Minutes: %s times\n", taka160, time160);
89     }
90     fclose(file160);
91     printf("\nAvailable list:\n");
92     file160 = fopen("data_statistics_available.txt", "r");
93     printf("\n");
94     while (fgets(data160, 100, file160))
95     {
96         char no160[100];
97         char taka160[100];
98         sscanf(data160, "%99s %99s", taka160, no160);
99         printf("                %s Minutes card is available: %s \n",
100 taka160, no160);
101     }
102     fclose(file160);
103     file160 = fopen("data_statistics_totalbalance.txt", "r");
104     printf("\n");
105     while (fgets(data160, 100, file160))
106     {
107         char taka160[100];
108         sscanf(data160, "%99s", taka160);

```

```

108         printf("\nTotal earn %s taka\n", taka160);
109     }
110     fclose(file160);
111 }
112 int is_exist(char data160[], char location160[])
113 {
114     char line160[100];
115     FILE *file160 = fopen(location160, "r");
116     if (file160 == NULL)
117     {
118         file160 = fopen(location160, "w");
119         fclose(file160);
120     }
121     while (fgets(line160, sizeof(line160), file160))
122     {
123         char mobile160[100];
124         if (sscanf(line160, "%99s", mobile160) == 1)
125         {
126             for (int i = 0; mobile160[i] != '\0'; i++)
127             {
128                 if (mobile160[i] == ' ')
129                 {
130                     mobile160[i] = '\0';
131                     break;
132                 }
133             }
134             size_t data_len160 = strlen(data160);
135             size_t mobile_len160 = strlen(mobile160);
136             size_t min_len160 = data_len160 < mobile_len160 ? data_len160 :
mobile_len160;
137             if (strncmp(data160, mobile160, min_len160) == 0)
138             {
139                 fclose(file160);
140                 return 1;
141             }
142         }
143     }
144     fclose(file160);
145     return 0;
146 }
147 void unblock_user_list()
148 {
149     char filename160[100] = "data_lockmobilen.txt";
150     char line160[100];
151     FILE *original_file160 = fopen(filename160, "r");
152     if (original_file160 == NULL)
153     {
154         original_file160 = fopen(filename160, "w");
155         fclose(original_file160);
156         original_file160 = fopen(filename160, "r");
157     }
158     while (fgets(line160, sizeof(line160), original_file160))
159     {
160         printf("\t\t\t%s", line160);
161     }

```

```

162     printf("\n\n");
163     fclose(original_file160);
164 }
165 int unblock_user(char card_number160[])
166 {
167     int cnt160 = 0;
168     char filename160[100] = "data_lockmobilen0.txt";
169     char temp_filename160[100] = "data_temp.txt";
170     char line160[100];
171     card_number160[strcspn(card_number160, "\n")] = 0;
172     card_number160[strcspn(card_number160, " ")] = 0;
173     FILE *original_file160 = fopen(filename160, "r");
174     FILE *temp_file160 = fopen(temp_filename160, "w");
175     while (fgets(line160, sizeof(line160), original_file160))
176     {
177         char current_card160[100];
178         if (sscanf(line160, "%11s", current_card160))
179         {
180             current_card160[strcspn(current_card160, "\n")] = 0;
181             current_card160[strcspn(current_card160, " ")] = 0;
182             if (!strcmp(current_card160, card_number160))
183             {
184                 cnt160 = 1;
185                 continue;
186             }
187         }
188         fputs(line160, temp_file160);
189     }
190     fclose(original_file160);
191     fclose(temp_file160);
192     remove(filename160);
193     rename(temp_filename160, filename160);
194     return cnt160;
195 }
196
197 void newcard_generetor()
198 {
199     srand(time(NULL));
200     int n160, choice160;
201     printf("Available package: \n");
202     printf("\t1. 40 min 50 taka\n");
203     printf("\t2. 60 min 70 taka\n");
204     printf("\t3. 100 min 120 taka\n");
205     printf("\tHow many cards and choose a number: ");
206     scanf("%d %d", &n160, &choice160);
207     char taka160[100] = "";
208     if (choice160 == 1)
209     {
210         strcat(taka160, "40");
211     }
212     else if (choice160 == 2)
213     {
214         strcat(taka160, "60");
215     }
216     else if (choice160 == 3)

```

```

217     {
218         strcat(taka160, "100");
219     }
220     else
221     {
222         printf("Wrong submission——→returning admin panel");
223         return;
224     }
225     for (int i = 0; i < n160; i++)
226     {
227         unused_data2(taka160);
228         char fullfilename160[200] = "data_validcard.txt";
229         char card160[21];
230         for (int i = 0; i < 20; i++)
231         {
232             card160[i] = '0' + (rand() % 10);
233         }
234         card160[20] = '\0';
235         FILE *f1160 = fopen(fullfilename160, "a");
236         if (f1160 == NULL)
237         {
238             FILE *f2160 = fopen(fullfilename160, "w");
239             fprintf(f2160, "%s %s\n", card160, taka160);
240             fclose(f2160);
241         }
242         else
243         {
244             fprintf(f1160, "%s %s\n", card160, taka160);
245             fclose(f1160);
246         }
247     }
248     printf("%d Cards Creation Successful \n ", n160);
249 }
250
251 int remove_card(char card_number160[])
252 {
253     char filename160[100] = "data_validcard.txt";
254     char temp_filename160[100];
255     char line160[100];
256     int card_found160 = 0;
257     int changes_made160 = 0;
258     snprintf(temp_filename160, 100, "%s.temp", filename160);
259     card_number160[strcspn(card_number160, "\n")] = 0;
260     FILE *original_file160 = fopen(filename160, "r");
261     FILE *temp_file160 = fopen(temp_filename160, "w");
262     while (fgets(line160, 100, original_file160))
263     {
264         char current_card160[100];
265         char taka160[100];
266         if (sscanf(line160, "%31s %99s", current_card160, taka160) == 2)
267         {
268             if (strcmp(current_card160, card_number160) == 0)
269             {
270                 card_found160 = 1;
271                 changes_made160 = 1;

```

```

272         unused_card(taka160);
273         continue;
274     }
275 }
276 fputs(line160, temp_file160);
277 }
278 fclose(original_file160);
279 fclose(temp_file160);
280 if (changes_made160)
281 {
282     if (remove(filename160) != 0)
283     {
284         return 1;
285     }
286     if (rename(temp_filename160, filename160) != 0)
287     {
288         return 1;
289     }
290     return 1;
291 }
292 else
293 {
294     if (remove(filename160) != 0)
295     {
296         return 1;
297     }
298     if (rename(temp_filename160, filename160) != 0)
299     {
300         return 1;
301     }
302     return 0;
303 }
304 }
305 void total_profit(char data160[])
306 {
307     char taka160[100] = "";
308     if (!strcmp(data160, "40"))
309     {
310         strcat(taka160, "50");
311     }
312     if (!strcmp(data160, "60"))
313     {
314         strcat(taka160, "70");
315     }
316     if (!strcmp(data160, "100"))
317     {
318         strcat(taka160, "120");
319     }
320     FILE *original_file160 = fopen("data_statistics_totalbalance.txt", "r");
321     FILE *temp_file160 = fopen("data_statistics_temp.txt", "w");
322     char line160[100];
323     if (original_file160 == NULL)
324     {
325         original_file160 = fopen("data_statistics_totalbalance.txt", "w");
326         fprintf(original_file160, "0");

```

```

327         fclose(original_file160);
328         original_file160 = fopen("data_statistics_totalbalance.txt", "r");
329     }
330     while (fgets(line160, sizeof(line160), original_file160))
331     {
332         int sum160 = atoi(line160) + atoi(taka160);
333         char string_num160[12];
334         snprintf(string_num160, sizeof(string_num160), "%d", sum160);
335         fprintf(temp_file160, "%s", string_num160);
336     }
337     fclose(original_file160);
338     fclose(temp_file160);
339     remove("data_statistics_totalbalance.txt");
340     rename("data_statistics_temp.txt", "data_statistics_totalbalance.txt");
341 }
342 void user_current_balance_updater(char number160[], char taka160[])
343 {
344     char add160[100] = "data_statistics_useramount_";
345     strcat(add160, number160);
346     strcat(add160, ".txt");
347     FILE *original_file160 = fopen(add160, "r");
348     FILE *temp_file160 = fopen("data_statistics_useramount_temp.txt", "w");
349     if (original_file160 == NULL)
350     {
351         original_file160 = fopen(add160, "w");
352         fprintf(original_file160, "0");
353         fclose(original_file160);
354         original_file160 = fopen(add160, "r");
355     }
356     char line160[100];
357     while (fgets(line160, 100, original_file160))
358     {
359         int sum160 = atoi(line160) + atoi(taka160);
360         char string_num160[12];
361         snprintf(string_num160, sizeof(string_num160), "%d", sum160);
362         fprintf(temp_file160, string_num160);
363     }
364     fclose(original_file160);
365     fclose(temp_file160);
366     remove(add160);
367     rename("data_statistics_useramount_temp.txt", add160);
368 }
369 void used_card(char data160[])
370 {
371     FILE *original_file160 = fopen("data_statistics_used.txt", "r");
372     FILE *temp_file160 = fopen("data_statistics_temp.txt", "w");
373     if (original_file160 == NULL)
374     {
375         original_file160 = fopen("data_statistics_used.txt", "w");
376         fclose(original_file160);
377         original_file160 = fopen("data_statistics_used.txt", "r");
378     }
379     char line160[10000];
380     int taka160 = 1;
381     while (fgets(line160, sizeof(line160), original_file160))

```

```

382     {
383         char money160[100];
384         char no160[100];
385         if (sscanf(line160, "%99s %99s", money160, no160) == 2)
386         {
387             if (!strcmp(money160, data160))
388             {
389                 taka160 += atoi(no160);
390                 continue;
391             }
392             else if (!strcmp(line160, " ")) continue;
393         }
394         fprintf(temp_file160, line160);
395     }
396     char string_num160[10000];
397     snprintf(string_num160, sizeof(string_num160), "%d", taka160);
398     fprintf(temp_file160, "%s %s\n", data160, string_num160);
399     fclose(original_file160);
400     fclose(temp_file160);
401     remove("data_statistics_used.txt");
402     rename("data_statistics_temp.txt", "data_statistics_used.txt");
403 }
404 void dynamic_data_updater(char data160[], char number160[])
405 {
406     total_profit(data160);
407     user_current_balance_updater(number160, data160);
408     used_card(data160);
409 }
410 char* cardno_money(char cardno160[])
411 {
412     FILE *file160 = fopen("data_validcard.txt", "r");
413     char data160[100];
414     while (fgets(data160, 100, file160))
415     {
416         char current_card160[100];
417         char taka160[100];
418         sscanf(data160, "%99s %99s", current_card160, taka160);
419         if (!strcmp(current_card160, cardno160))
420         {
421             fclose(file160);
422             return strdup(taka160);
423         }
424     }
425     fclose(file160);
426     return NULL;
427 }
428 void full_time_history(char cardno160[], char moneey160[], char
mobilen160[])
429 {
430     FILE *file160 = fopen("data_history.txt", "r");
431     time_t rawtime160;
432     struct tm *timeinfo160;
433     char buffer160[80];
434     time(&rawtime160);
435     timeinfo160 = localtime(&rawtime160);

```



```

436     strftime(buffer160, sizeof(buffer160), "%Y-%m-%d\t%H:%M:%S",
timeinfo160);
437     if (file160 == NULL)
438     {
439         FILE *file160 = fopen("data_history.txt", "w");
440         fprintf(file160, "          Card No          Date          Time
Minute      mobile no");
441         fclose(file160);
442     }
443     file160 = fopen("data_history.txt", "a");
444     fprintf(file160, "\n%s\t%s\t%s\t%s", cardno160, buffer160, moneey160,
mobileno160);
445     fclose(file160);
446     return;
447 }
448 void user_full_time_history(char cardno160[], char mobileno160[])
449 {
450     time_t rawtime160;
451     struct tm *timeinfo160;
452     char buffer160[80];
453     time(&rawtime160);
454     timeinfo160 = localtime(&rawtime160);
455     strftime(buffer160, sizeof(buffer160), "%Y-%m-%d\t%H:%M:%S",
timeinfo160);
456     char data160[100] = "data_search_";
457     strcat(data160, mobileno160);
458     strcat(data160, ".txt");
459     FILE *file160 = fopen(data160, "r");
460     if (file160 == NULL)
461     {
462         FILE *file160 = fopen(data160, "w");
463         fprintf(file160, "          Card No          Date          Time
Minute      ");
464         fclose(file160);
465     }
466     file160 = fopen(data160, "a");
467     fprintf(file160, "\n%s\t%s\t%s", cardno160, buffer160,
cardno_money(cardno160));
468     fclose(file160);
469     return;
470 }
471 void user_panel()
472 {
473     while (1)
474     {
475         char mobileno160[100];
476         char cardno160[100];
477         char line160[100];
478         printf("Enter your mobile number or enter 0 for exit: ");
479         scanf("%s", mobileno160);
480         mobileno160[strcspn(mobileno160, "\n")] = 0;
481         char blocklocation160[100] = "data_lockmobileno.txt";
482
483         if (!strcmp("0", mobileno160))
484         {

```

```

485         printf("\n");
486         break;
487     }
488     if (strlen(mobileno160) != 11)
489     {
490         printf("\nEnter a valid mobile no.\n");
491     }
492     else if (is_exist(mobileno160, blocklocation160))
493     {
494         printf("\n\nYOU are a blocked user\n\n");
495     }
496     else
497     {
498         while (1)
499         {
500             if (is_exist(mobileno160, blocklocation160))
501             {
502                 printf("\n\nYOU are a blocked user\n\n");
503                 break;
504             }
505             int n;
506             printf("\nchoose a
option\n\t1.Balance\n\t2.Recharge\n\t3.logout\n\tEnter a valid digit: ");
507             scanf("%d", &n);
508             if (n == 1)
509             {
510                 char address160[100] = "data_statistics_useramount_";
511                 strcat(address160, mobileno160);
512                 address160[strlen(address160, "\n")] = 0;
513                 strcat(address160, ".txt");
514                 FILE *file160 = fopen(address160, "r");
515                 if (file160 == NULL)
516                 {
517                     file160 = fopen(address160, "w");
518                     if (file160 != NULL)
519                     {
520                         fprintf(file160, "0");
521                         fclose(file160);
522                         file160 = fopen(address160, "r");
523                     }
524                 }
525                 if (file160 != NULL)
526                 {
527                     char line160[100];
528                     if (fgets(line160, 100, file160))
529                     {
530                         line160[strlen(line160, "\n")] = 0;
531                         printf("Your total Balance is %s minutes\n",
line160);
532                     }
533                     fclose(file160);
534                 }
535             }
536             else if (n == 2)
537             {

```

```

538         int t = 3;
539         int flag = 1;
540         while (t--)
541         {
542             printf("Enter Your card no: ");
543             scanf("%s", cardno160);
544             if (strlen(cardno160) != 20)
545             {
546                 printf("Enter a valid card no...\n");
547             }
548             else if (is_exist(cardno160, "data_validcard.txt"))
549             {
550                 for (int i = 0; i < strlen(cardno160); i++)
551                 {
552                     if (cardno160[i] == '\n') cardno160[i] =
553                         '\0';
554                 }
555                 for (int i = 0; i < strlen(mobileno160); i++)
556                 {
557                     if (mobileno160[i] == '\n') mobileno160[i] =
558                         '\0';
559                 }
560                 dynamic_data_updater(cardno_money(cardno160),
561                     mobileno160);
562                 full_time_history(cardno160,
563                     cardno_money(cardno160), mobileno160);
564                 user_full_time_history(cardno160, mobileno160);
565                 printf("%s minutes recharge successful\n",
566                     cardno_money(cardno160));
567                 remove_card(cardno160);
568                 flag = 0;
569                 break;
570             }
571             printf("You have only %d attempts left\n", t);
572         }
573         if (flag)
574         {
575             FILE *file160 = fopen("data_lockmobileno.txt", "a");
576             fprintf(file160, "%s\n", mobileno160);
577             printf("\n\nYou have been blocked.\n\n");
578             fclose(file160);
579             break;
580         }
581     }
582     else if (n == 3)
583     {
584         return;
585     }
586     else
587     {
588         printf("\nEnter a valld digit!!!\n\n");
589     }
590 }
591 }
592 }

```

```

588 }
589 void admin_panel()
590 {
591     srand(time(NULL));
592     printf("Welcome to the Admin Panel: \n");
593     while (1)
594     {
595         printf("\n\t1. Generate new Card\n");
596         printf("\t2. Delete Card\n");
597         printf("\t3. Unblock mobile number\n");
598         printf("\t4. History\n");
599         printf("\t5. Statistics\n");
600         printf("\t6. search\n");
601         printf("\t7. Exit\n");
602         printf("\t Enter your option: ");
603         int option;
604         scanf("%d", &option);
605         if (option == 1)
606         {
607             newcard_generetor();
608         }
609         else if (option == 2)
610         {
611             char data160[100];
612             printf("Enter card no: ");
613             scanf("%s", data160);
614             if (remove_card(data160))
615             {
616                 printf("Card %s was found and deleted.\n", data160);
617             }
618             else
619             {
620                 printf("Card %s was invalid!!!!.\n", data160);
621             }
622         }
623         else if (option == 3)
624         {
625             while (1)
626             {
627                 printf("Choose a option: \n");
628                 printf("\t1.showing block user:\n");
629                 printf("\t2.unblock user\n");
630                 printf("\t3.exit\n");
631                 printf("\tEnter a VALid digit: ");
632                 int x;
633                 scanf("%d", &x);
634                 if (x == 3) return;
635                 else if (x == 1)
636                 {
637                     printf("Block users list:\n");
638                     unblock_user_list();
639                 }
640                 else if (x != 2)
641                 {
642                     printf("\nEnter a valid digit!!!1\n");

```

```

643     }
644     else
645     {
646         printf("Enter the mobile no: ");
647         char mobile_no160[100];
648         scanf("%s", mobile_no160);
649         if (unblock_user(mobile_no160))
650         {
651             printf("%s unblock successful.\n", mobile_no160);
652         }
653         else
654         {
655             printf("%s is not a block user..\n", mobile_no160);
656         }
657     }
658 }
659 }
660 else if (option == 4)
661 {
662     FILE *file160 = fopen("data_history.txt", "r");
663     char line160[100];
664     if (file160 == NULL)
665     {
666         file160 = fopen("data_history.txt", "w");
667         fprintf(file160, "          Card No          Date
Time          Minute          mobile no");
668         fclose(file160);
669         file160 = fopen("data_history.txt", "r");
670     }
671     while (fgets(line160, sizeof(line160), file160))
672     {
673         printf("%s", line160);
674     }
675     fclose(file160);
676 }
677 else if (option == 5) statistics();
678 else if (option == 6)
679 {
680     printf("Enter the mobile no: ");
681     char mobileno160[100];
682     scanf("%s", mobileno160);
683     if (strlen(mobileno160) == 11)
684     {
685         char data160[100] = "data_search_";
686         strcat(data160, mobileno160);
687         strcat(data160, ".txt");
688         FILE *file160 = fopen(data160, "r");
689         if (file160 == NULL)
690         {
691             file160 = fopen(data160, "w");
692             fprintf(file160, "          Card No          Date
Time          Minute          ");
693             fclose(file160);
694             file160 = fopen(data160, "r");
695         }

```

```

696         char line160[100];
697         while (fgets(line160, sizeof(line160), file160))
698         {
699             printf("%s", line160);
700         }
701         fclose(file160);
702     }
703     else
704     {
705         printf("Enter a valid mobile number!!!");
706     }
707 }
708 else if (option == 7) return;
709 else
710 {
711     printf("\nPlease Enter a valid number!!!\n\n");
712 }
713 }
714 }
715 int main()
716 {
717     while (1)
718     {
719         printf("Prepared by Nilay Paul Partha(Roll:2403160)\n\n");
720         printf("Select Option\n");
721         printf("\t1.Admin panel\n");
722         printf("\t2.User panel\n");
723         printf("\t3.Exit\n");
724         printf("\tEnter a digit: ");
725         int option;
726         scanf("%d", &option);
727         if (option == 1)
728         {
729             int t = 3;
730             while (t--)
731             {
732                 printf("Enter Your Admin panel password(2403160) : ");
733                 int password;
734                 int flag = 0;
735                 scanf("%d", &password);
736                 if (password == 2403160)
737                 {
738                     admin_panel();
739                     flag = 1;
740                 }
741                 else
742                 {
743                     printf("Wrong password. You have %d attempts
744 remaining.\n", t);
745                     if (flag) break;
746                 }
747             }
748             else if (option == 2)
749             {

```

```
750         user_panel();
751     }
752     else if (option == 3) return 0;
753     else
754     {
755         printf("\nEnter a valid number!!!\n\n");
756     }
757 }
758 }
759
```