PROJECT TITLE

CAR RENTAL SYSTEM

Language : C

Model Type : Individual

College : Lakshmi Narain College of Technology

**Report Format**

Submitted Department : T & P

Submitted To : Rambabu Lovewanshi

Name of student : Adarsh Kumar

Branch : EC

Semester : 5th

Enrollment No: : 0103EC181006

Submission Date : 31-DEC-2020

Contents

Topic

1. Introduction …………………………………..……………………………………………….………….3
2. Software and Hardware Requirement…………………………………….……..6

3. Literature Survey………………………………………………………………………………….…..7

3.1 Technical feasibility…………………………………………………………………..7

* 1. Operational feasibility…………………………………………………….....……8
  2. Economical Feasibility…………………………………………………………..…9

1. Code Optimization…………………….……………………………………………..……………..10
2. Output Screen………………………………………..……………………………………….……….45
3. Conclusion…………………………………..………………………….………………………………….61
4. Future Enhancement………………………………………………………………...………….62
5. APPENDICES …………………………………………………………………………..………..………63
6. Reference………………………………….…………………………………………...…………..…….64
7. Bibliography ………………………………………………….………….……...………………...…65

INTRODUCTION

Car Rental System , is a management application that is used to manage the records of the customer and the number of cars available. We all know that nowadays managing records or the data on notebooks become very old and tedious work. And, nobody likes to do the calculations so this application will help all those people who are facing the same issue.

In this application, there are two panels one for the admin and the other is for the employee.

Admin Role:

The Admin is able to add employees, new cars, update the information of cars and employees, and also able to see the number of bookings and the total number of cars available for rent.

Instead of that if the admin wants, he/she can also remove the employee or can delete the cars from the list whichever he/she wants.

Admin Options:

1. Add Employee  
2. Add Car Details  
3. Show Employee

4. Show Car Details

5. Delete Employee

6. Delete Car Details

7. Exit

**Employee Role:**

The employee is able to rent an available car to the customer and insert the important information of the customer to the file like his name of the customer, name of the car which he has taken for rent, the pickup location from where he picked the car, the drop location, the date of hiring the car, the date of returning and etc in the file. So, he does not have to maintain a record book because all

this data has been saved in his computer in binary files, so whenever he wants, he can fetch these records.

Also, these record files are saved in binary format so it is very difficult for other peoples to see those records or manipulates them, it also provides more security if we compare it with our traditional way.

Employee Options:

1. Rent A Car  
2. Booking Details  
3. Available Car Details

4. Show All Car Details

5. Exit

**File Handling:**

File handling in programming is a mechanism using which a programmer can save all the data (values stored in variables, array, structure, etc) in the secondary memory of our computer so that we can retrieve it in the future.

In simple words, we can say that file handling provides data persistence.

File handling in C refers to the task of storing data in the form of input or output produced by running C programs in data files, namely, a text file or a binary file for future reference and analysis.

So far, the operations using the C program are done on a prompt / terminal which is not stored anywhere. But in the software industry, most of the programs are written to store the information fetched from the program.

Software and Hardware requirements

SOFTWARE REQUIREMENTS SPECIFICATION

Operating System : macOS, Linux, and Windows

Programming Language : C

User Interface : CLI

Database : File Handling

HARDWARE REQUIREMENTS SPECIFICATION

Processor : Pentium IV processor or higher

RAM : Minimum of 1GB RAM

Memory : 500 MB or higher

1. **Literature survey / Review of Literature**

**INTRODUCTION**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analysed carefully. There are 3 parts in feasibility study.

**3.1 TECHNICAL FEASIBILITY**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?.

**3.2 OPERATIONAL FEASIBILITY**

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

* Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
* Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
* Have the user been involved in the planning and development of the project?
* Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

* 1. **ECONOMIC FEASIBILITY**

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

CODE SNIPPETS FOR TRAINING

Car.c

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<string.h>
4. #include<time.h>
5. #include"car.h"
6. #include"adarsh.h"
7. #include <termios.h>
8. #include <unistd.h>
9. #define BLUE(string) "\x1b[34m" string "\x1b[0m"
10. #define RED(string) "\x1b[31m" string "\x1b[0m"
11. #define ANSI\_COLOR\_RED "\x1b[31m"
12. #define ANSI\_COLOR\_GREEN "\x1b[32m"
13. #define ANSI\_COLOR\_YELLOW "\x1b[33m"
14. #define ANSI\_COLOR\_BLUE "\x1b[34m"
15. #define ANSI\_COLOR\_MAGENTA "\x1b[35m"
16. #define ANSI\_COLOR\_CYAN "\x1b[36m"
17. #define ANSI\_COLOR\_RESET "\x1b[0m"

20. void addAdmin()
21. {
22. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/admin.bin", "rb");
23. if(fp==NULL){
24. fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/admin.bin", "wb");
25. User u[2]={{"adarsh", "abc", "Adarsh"},{"admin", "abc", "Adarsh Kumar"}};
26. fwrite(u, sizeof(u), 1, fp);
27. fclose(fp);
28. getch();
29. }
31. else {
32. fclose(fp);
33. }
35. }
37. User\* getInput(){
39. int i;
40. clrscr1();
41. gotoxy(32,1);
42. printf(RED("CAR RENTAL SYSTEM\n") );
43. for(i=1; i<=80; i++)
44. printf("%c", '=');
45. gotoxy(32,4);
46. printf("\033[22;34m\* LOGIN PANEL \*\033[0m");
47. gotoxy(1,6);
48. for(i=1; i<80; i++)
49. printf("%c",'=');
50. gotoxy(1,14);
51. for(i=1; i<80; i++)
52. printf("%c",'=');
53. gotoxy(60,8);
54. printf(ANSI\_COLOR\_MAGENTA "Press 0 to exit" ANSI\_COLOR\_RESET "\n");
55. gotoxy(25,10);
56. printf(ANSI\_COLOR\_CYAN "Enter User ID:" ANSI\_COLOR\_RESET);
57. fflush(stdin);
59. static User usr;
61. fgets(usr.userid,20,stdin);
62. char \*pos;
63. pos=strchr(usr.userid, '\n');
64. \*pos='\0';
66. if(strcmp(usr.userid,"0")==0){
68. gotoxy(30,17);
69. printf(RED("Login Cancelled!") );
70. getch();
71. return NULL;
73. }
75. gotoxy(25,11);
76. printf(ANSI\_COLOR\_CYAN "Enter Password:" ANSI\_COLOR\_RESET);
77. fflush(stdin);
79. i=0;
80. for(;;){
81. usr.pwd[i]=getch();
83. if(usr.pwd[i]=='\n'){
84. break;
85. }
86. if(usr.pwd[i]==127 || usr.pwd[i]=='\b'){
88. if (usr.pwd [i]!= 0){
89. printf("\b \b");
90. i=i-2;
91. }
92. }
93. else
94. printf("\*");
95. if(usr.pwd[i]=='\r')
96. break;
97. i++;
98. }
99. usr.pwd[i]='\0';
100. if(strcmp(usr.pwd,"0")==0){
102. gotoxy(30,17);
103. printf(RED("Login Cancelled!") );
104. getch();
105. return NULL;
106. }
107. return &usr;
109. }

112. int checkUserExist(User u, char \*usertype){
114. if(strcmp(u.userid, "")==0 || strcmp(u.pwd, "")==0){
116. gotoxy(28,20);
117. printf(RED("BOTH FIELDS ARE MANDATORY") );
118. getch();
119. gotoxy(28,20);
120. printf("\t\t\t\t\t\t\t\t\t");
121. return 0;
122. }
124. int found=0;
126. if(!(strcmp(usertype, "admin"))){
128. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/admin.bin", "rb");
129. User user;
131. while(fread(&user, sizeof(User),1,fp)==1){
133. if(strcmp(u.userid,user.userid)==0 && strcmp(u.pwd,user.pwd)==0 ){
135. found=1;
136. break;
137. }
139. }
141. if(found==0){
143. gotoxy(27,20);
144. printf(RED("INVALID USERID OR PASSWORD") );
145. getch();
146. fclose(fp);
147. return 0;
148. }
150. fclose(fp);
151. }



156. else if(!(strcmp(usertype, "emp"))){
158. // Code for reading emp.bin
159. FILE \*fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "rb");
160. User user2;
162. while(fread(&user2, sizeof(User),1,fp1)==1){
164. if(strcmp(u.userid,user2.userid)==0 && strcmp(u.pwd,user2.pwd)==0 ){
166. found=1;
167. break;
168. }
170. }
171. if(found==0){
173. gotoxy(27,20);
174. printf(RED("INVALID USERID OR PASSWORD") );
175. getch();
176. fclose(fp1);
177. return 0;
178. }
180. fclose(fp1);
182. }
184. return 1;
185. }


189. int adminMenu(){
191. int choice,i;
192. gotoxy(32,2);
193. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM" ANSI\_COLOR\_RESET);
194. gotoxy(35,6);
195. printf("ADMIN MENU\n");
196. for(i=0; i<=79; i++)
197. printf("\*");
198. gotoxy(32,8);
199. printf(ANSI\_COLOR\_YELLOW "1. Add Employee" ANSI\_COLOR\_RESET);
200. gotoxy(32,9);
201. printf(ANSI\_COLOR\_YELLOW "2. Add Car Details" ANSI\_COLOR\_RESET);
202. gotoxy(32,10);
203. printf(ANSI\_COLOR\_YELLOW "3. Show Employee" ANSI\_COLOR\_RESET);
204. gotoxy(32,11);
205. printf(ANSI\_COLOR\_YELLOW "4. Show Car Details" ANSI\_COLOR\_RESET);
206. gotoxy(32,12);
207. printf(ANSI\_COLOR\_YELLOW "5. Delete Employee" ANSI\_COLOR\_RESET);
208. gotoxy(32,13);
209. printf(ANSI\_COLOR\_YELLOW "6. Delete Car Details" ANSI\_COLOR\_RESET);
210. gotoxy(32,14);
211. printf(ANSI\_COLOR\_YELLOW "7. Exit" ANSI\_COLOR\_RESET);
212. gotoxy(32,15);
213. printf(ANSI\_COLOR\_YELLOW "Enter Choice:" ANSI\_COLOR\_RESET);
214. scanf("%d", &choice);
215. return choice;
217. }


221. int empMenu(){
223. int choice,i;
224. gotoxy(32,2);
225. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM" ANSI\_COLOR\_RESET);
226. gotoxy(35,6);
227. printf(ANSI\_COLOR\_GREEN "EMPLOYEE MENU\n" ANSI\_COLOR\_RESET);
228. for(i=0; i<=79; i++)
229. printf("\*");
230. gotoxy(32,8);
231. printf(ANSI\_COLOR\_YELLOW "1. Rent A Car" ANSI\_COLOR\_RESET);
232. gotoxy(32,9);
233. printf(ANSI\_COLOR\_YELLOW "2. Booking Details" ANSI\_COLOR\_RESET);
234. gotoxy(32,10);
235. printf(ANSI\_COLOR\_YELLOW "3. Available Car Details" ANSI\_COLOR\_RESET);
236. gotoxy(32,11);
237. printf(ANSI\_COLOR\_YELLOW "4. Show All Car Details" ANSI\_COLOR\_RESET);
238. gotoxy(32,12);
239. printf(ANSI\_COLOR\_YELLOW "5. Exit" ANSI\_COLOR\_RESET);
240. gotoxy(32,15);
241. printf("Enter Choice:");
242. scanf("%d",&choice);
243. return choice;
245. }

248. void addCarDetails(){
250. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "rb");
251. int id=1;
252. char uchoice;
253. Car c1,c2;
255. if(fp==NULL){
257. fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "ab");
258. id=1;
259. c1.car\_id=id;

262. }
264. else{
266. fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "ab+");
267. fseek(fp, -1L\*sizeof(c1), SEEK\_END);
268. fread(&c2, sizeof(c2), 1, fp);
269. id=c2.car\_id;
270. printf("%d",id);
271. id++;
272. c2.car\_id=id;
273. c1.car\_id=c2.car\_id;
275. }

278. do{
280. clrscr1();
281. c1.car\_id=id;
282. gotoxy(32,2);
283. printf(ANSI\_COLOR\_RED "CAR RENTAL APP" ANSI\_COLOR\_RESET);
284. int i;
285. gotoxy(1,3);
286. for(i=0; i<80; i++)
287. printf("~");
288. gotoxy(25, 5);
289. printf("\*\*\*\*\* ADD CAR DETAILS \*\*\*\*\*\*");
290. gotoxy(1,8);
291. printf(ANSI\_COLOR\_CYAN "Enter Car Name :" ANSI\_COLOR\_RESET);
292. fflush(stdin);
293. fgets(c1.car\_name, 50, stdin);
294. char \*pos;
295. pos=strchr(c1.car\_name, '\n');
296. \*pos='\0';
297. printf(ANSI\_COLOR\_CYAN "Enter Capacity:" ANSI\_COLOR\_RESET);
298. fflush(stdin);
299. scanf("%d",&c1.capacity);
300. printf(ANSI\_COLOR\_CYAN "Enter Number of Cars:" ANSI\_COLOR\_RESET);
301. scanf("%d",&c1.car\_count);
302. printf(ANSI\_COLOR\_CYAN "Enter Price:" ANSI\_COLOR\_RESET);
303. scanf("%d",&c1.price);
304. fseek(fp, 0, SEEK\_END);
305. fwrite(&c1, sizeof(Car), 1,fp);
306. gotoxy(30,15);
307. printf(ANSI\_COLOR\_GREEN "CAR ADDED SUCCESSFULLY" ANSI\_COLOR\_RESET);
308. printf("\n CAR ID IS: %d",c1.car\_id);
309. getch();
310. gotoxy(1,20);
311. printf(ANSI\_COLOR\_RED "DO YOU WANT TO ADD MORE CAR (Y/N) : ");
312. fflush(stdin);
313. scanf("%c", &uchoice);
314. id++;
315. c1.car\_id=id;
317. }while(uchoice=='Y' || uchoice=='y');
319. fclose(fp);
320. }
322. void addEmployee(){
324. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "rb");
325. int id;
326. char uchoice;
327. char emp[10]="EMP-";
328. char empid[10];
329. User u,ur;
331. if(fp==NULL){
333. fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "ab");
334. id=1;
335. sprintf(empid, "%d", id); //(empid=1)
336. strcat(emp, empid); //emp=EMP-1
337. strcpy(u.userid, emp); //u.userid=EMP-1
338. strcpy(emp, "EMP-"); //emp=EMP-
340. }
342. else{
344. fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "ab+");
345. fseek(fp, -60, SEEK\_END);
346. fread(&ur, sizeof(ur), 1, fp);//ur
347. char sub[3];
348. char str[20];
349. strcpy(str, ur.userid); //str=EMP-1
350. sub\_str(str, sub, '-');
351. id=myAtoi(sub);
352. id++;
353. sprintf(empid, "%d", id); //EMP-2
354. strcat(emp,empid);
355. strcpy(u.userid,emp);
356. strcpy(emp, "EMP-");
358. }
360. do{
362. clrscr1();
363. gotoxy(32,2);
364. printf(ANSI\_COLOR\_RED "CAR RENTAL APP" ANSI\_COLOR\_RESET);
365. int i;
366. gotoxy(1,3);
367. for(i=0; i<80; i++)
368. printf("~");
369. gotoxy(25, 5);
370. printf("\*\*\*\*\* ADD EMPLOYEE DETAILS \*\*\*\*\*\*");
371. gotoxy(1,8);
372. printf(ANSI\_COLOR\_CYAN "Enter Employee Name :" ANSI\_COLOR\_RESET);
373. fflush(stdin);
374. fgets(u.name, 20, stdin);
375. char \*pos;
376. pos=strchr(u.name, '\n');
377. \*pos='\0';
378. printf(ANSI\_COLOR\_CYAN "Enter Employee Pwd:" ANSI\_COLOR\_RESET);
379. fflush(stdin);
380. fgets(u.pwd,20,stdin);
381. pos=strchr(u.pwd, '\n');
382. \*pos='\0';
383. fseek(fp, 0, SEEK\_END);
384. fwrite(&u, sizeof(User), 1,fp);
385. gotoxy(30,15);
386. printf(ANSI\_COLOR\_GREEN "EMPLOYEE ADDED SUCCESSFULLY" ANSI\_COLOR\_RESET);
387. printf("\n EMPLOYEE ID IS: %s",u.userid);
388. getch();
389. gotoxy(1,20);
390. printf(ANSI\_COLOR\_RED "DO YOU WANT TO ADD MORE EMPLOYEE(Y/N) : ");
391. fflush(stdin);
392. scanf("%c", &uchoice);
393. id++;
394. sprintf(empid, "%d", id);
395. strcat(emp, empid);
396. strcpy(u.userid, emp);
397. strcpy(emp, "EMP-");
399. }while(uchoice=='Y' || uchoice=='y');
401. fclose(fp);
403. }


407. void sub\_str(char \*str, char \*sub, char c)
408. {
409. int i=0, f=0, j=0;
410. for(i=0; str[i]!=c; i++){
411. f++;
412. }
413. for(i=++f; str[i]!='\0'; i++){
415. sub[j]=str[i];
416. j++;
417. }
418. sub[j]='\0';
419. }


423. void showCarDetails(){
425. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "rb");
426. Car ur;
427. int i;
428. gotoxy(32,1);
429. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
430. for(i=1; i<=80; i++)
431. printf("\*");
432. gotoxy(31,5);
433. printf(ANSI\_COLOR\_YELLOW "\* CAR DETAILS \*" ANSI\_COLOR\_RESET);
434. gotoxy(1,7);
435. for(i=0; i<=80; i++)
436. printf("\*");
437. gotoxy(1,8);
438. printf(" CAR ID\t\tName\t\tCapacity\tNumber\t\tPrice");
439. gotoxy(1,9);
440. for(i=1;i<=80; i++)
441. printf("\*");
442. int x=10;
443. while(fread(&ur, sizeof(ur), 1,fp) == 1)
444. {
445. gotoxy(5,x);
446. printf("%d", ur.car\_id);
447. gotoxy(17,x);
448. printf("%s", ur.car\_name);
449. gotoxy(35, x);
450. printf("%d", ur.capacity);
451. gotoxy(50, x);
452. printf("%d", ur.car\_count);
453. gotoxy(65, x);
454. printf("%d", ur.price);
455. x++;
456. }
458. getch();
459. fclose(fp);
460. getch();
461. }


465. void viewEmployee(){
467. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "rb");
468. User ur;
469. int i;
470. gotoxy(32,1);
471. printf(ANSI\_COLOR\_YELLOW "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
472. for(i=1; i<=80; i++)
473. printf("\*");
474. gotoxy(31,5);
475. printf(ANSI\_COLOR\_YELLOW "\* EMPLOYEE DETAILS \*" ANSI\_COLOR\_RESET);
476. gotoxy(1,7);
477. for(i=0; i<=80; i++)
478. printf("\*");
479. gotoxy(1,8);
480. printf(" Employee ID\t\t\tName\t\t\tPassword");
481. gotoxy(1,9);
482. for(i=1;i<=80; i++)
483. printf("\*");
484. int x=10;
485. while(fread(&ur, sizeof(ur), 1,fp) == 1)
486. {
487. gotoxy(2,x);
488. printf("%s", ur.userid);
489. gotoxy(33,x);
490. printf("%s", ur.name);
491. gotoxy(57, x);
492. printf("%s", ur.pwd);
493. x++;
494. }
496. getch();
497. fclose(fp);
498. getch();
500. }


504. int deleteCarModel(){
506. FILE \*fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "rb");
508. int id,i;
510. gotoxy(32,1);
511. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
512. for(i=1; i<=80; i++)
513. printf("\*");
514. gotoxy(29,5);
515. printf(ANSI\_COLOR\_BLUE "\* DELETE CAR RECORD \*" ANSI\_COLOR\_RESET);
516. gotoxy(1,7);
517. for(i=1; i<=80; i++)
518. printf("\*");
519. gotoxy(1,12);
520. for(i=1; i<80; i++)
521. printf("\*");
522. if(fp1==NULL){
524. printf(ANSI\_COLOR\_RED "\nNo Car Added Yet!" ANSI\_COLOR\_RESET);
525. return -1;
527. }
529. FILE \*fp2=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/temp.bin", "wb+");
530. gotoxy(10,9);
531. printf(ANSI\_COLOR\_YELLOW "Enter Car ID to Delete the Record:" ANSI\_COLOR\_RESET);
532. scanf("%d",&id);
534. Car U;
535. int found=0;
536. while(fread(&U, sizeof(U), 1, fp1)==1){
538. if(U.car\_id!=id){
539. fwrite(&U, sizeof(U), 1, fp2);
540. }
541. else{
542. found=1;
543. }
545. }
547. fclose(fp1);
549. if(found==1){
551. rewind(fp2);
552. fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "wb");
553. while(fread(&U, sizeof(U), 1, fp2) ==1){
554. fwrite(&U, sizeof(U), 1,fp1);
555. }
556. fclose(fp1);
558. }
560. fclose(fp2);
561. remove("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/temp.bin");
562. return found;
564. }






572. int deleteEmp(){
574. FILE \*fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "rb");
575. char empid[10];
576. int i;
577. gotoxy(32,1);
578. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
579. for(i=1; i<=80; i++)
580. printf("\*");
581. gotoxy(29,5);
582. printf(ANSI\_COLOR\_BLUE "\* DELETE EMPLOYEE RECORD \*" ANSI\_COLOR\_RESET);
583. gotoxy(1,7);
584. for(i=1; i<=80; i++)
585. printf("\*");
586. gotoxy(1,12);
587. for(i=1; i<80; i++)
588. printf("\*");
589. if(fp1==NULL){
591. printf(ANSI\_COLOR\_RED "\nNo Employee Added Yet!" ANSI\_COLOR\_RESET);
592. return -1;
594. }
596. FILE \*fp2=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/temp.bin", "wb+");
597. gotoxy(10,9);
598. printf(ANSI\_COLOR\_YELLOW "Enter Employee ID to Delete the Record:" ANSI\_COLOR\_RESET);
599. scanf("%s",empid);
600. User U;
601. int found=0;
602. while(fread(&U, sizeof(U), 1, fp1)==1){
604. if(strcmp(U.userid,empid) != 0){
605. fwrite(&U, sizeof(U), 1, fp2);
606. }
607. else{
608. found=1;
609. }
611. }
613. fclose(fp1);
614. if(found==1){
616. rewind(fp2);
617. fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/emp.bin", "wb");
618. while(fread(&U, sizeof(U), 1, fp2) ==1){
619. fwrite(&U, sizeof(U), 1,fp1);
620. }
621. fclose(fp1);
623. }
625. fclose(fp2);
626. remove("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/temp.bin");
627. return found;
629. }













644. int selectCarModel(){
646. int flag=0;
647. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "rb");
648. Car C;
649. int choice,x=9;
650. gotoxy(34,x);
651. while(fread(&C, sizeof(C), 1,fp)==1){
653. if(C.car\_count>0){
654. printf("%d . %s", C.car\_id,C.car\_name);
655. gotoxy(34,++x);
657. }
658. }
660. gotoxy(34, x+2);
661. printf("Enter Your Choice:");
663. while(1){
665. flag=0;
666. scanf("%d",&choice);
667. rewind(fp);
669. while(fread(&C, sizeof(C), 1, fp)==1){
671. if(C.car\_id==choice && C.car\_count>0){
673. flag=1;
674. break;
675. }
676. }
678. if(flag==1){
680. fclose(fp);
681. return choice;
682. }
684. else{
686. gotoxy(37, x+4);
687. printf(ANSI\_COLOR\_RED "Wrong Input" ANSI\_COLOR\_RESET);
688. getch();
689. gotoxy(35, x+4);
690. printf("\t\t");
691. gotoxy(52, x+2);
692. printf("\t");
693. gotoxy(52, x+2);
695. }
696. }
698. }

701. int isValidDate(struct tm dt){
703. if(dt.tm\_year>=2020 && dt.tm\_year<=2022){
705. if(dt.tm\_mon>=1 && dt.tm\_mon<=12){
707. if((dt.tm\_mday>=1 && dt.tm\_mday<=31) && (dt.tm\_mon==1 || dt.tm\_mon==3 || dt.tm\_mon==5 || dt.tm\_mon==7 || dt.tm\_mon==8 || dt.tm\_mon==10 || dt.tm\_mon==12))
708. return 1;
709. else if((dt.tm\_mday>=1 && dt.tm\_mday<=30) && (dt.tm\_mon==4 || dt.tm\_mon==6 || dt.tm\_mon==9 || dt.tm\_mon==11))
710. return 1;
711. else if((dt.tm\_mday>=1 && dt.tm\_mday<=28) && (dt.tm\_mon==2))
712. return 1;
713. else if(dt.tm\_mday==29 && dt.tm\_mon==2 && (dt.tm\_year%400==0 || (dt.tm\_year%4==0 && dt.tm\_year%100!=0)))
714. return 1;
716. else
717. return 0;
719. }
721. else {
723. return 0;
724. }
726. }
728. else{
730. return 0;
731. }
732. }


736. void updateCarCount(int c){
738. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin" , "rb+");
739. Car C;
740. while(fread(&C, sizeof(Car), 1, fp)==1){
742. if(C.car\_id==c){
744. fseek(fp, -8, SEEK\_CUR);
745. int cc\_new=C.car\_count-1;
746. printf("%d",C.car\_count);
747. printf("%d",cc\_new);
748. fwrite(&cc\_new, sizeof(cc\_new), 1, fp);
749. }
750. }
751. fclose(fp);
752. }

755. char\* getCarName(int car\_id){
757. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin" , "rb");
758. static Car C;
760. while(fread(&C, sizeof(C), 1, fp) ==1){
762. if(C.car\_id==car\_id){
763. break;
764. }
765. }
767. fclose(fp);
768. return C.car\_name;
769. }


773. void availableCarDetails(){
775. static int a,b,I,s,r=0;
776. int i=0; int x=10;
777. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/customer.bin" , "rb");
778. FILE \*fp1=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/car.bin", "rb");
779. Car ur;
780. Customer\_Car\_Details CC;
781. while(fread(&CC, sizeof(Customer\_Car\_Details), 1, fp)==1){
783. if(strcmp(getCarName(CC.car\_id),"Audi")==0){
784. a++;
785. printf("a = %d\n",a);
787. }
789. if(strcmp(getCarName(CC.car\_id),"BMW")==0){
790. b++;
791. printf("b= %d\n",b);
792. }
794. if(strcmp(getCarName(CC.car\_id),"i10")==0){
795. I++;
796. printf("I= %d\n",I);
797. }
799. if(strcmp(getCarName(CC.car\_id),"Swift")==0){
800. s++;
801. printf("s = %d\n",s);
802. }
804. if(strcmp(getCarName(CC.car\_id),"Lexus")==0){
805. r++;
806. printf("r= %d\n",r);
807. }
809. }

812. clrscr1();
813. gotoxy(32,1);
814. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
815. for(i=1; i<=80; i++)
816. printf("\*");
817. gotoxy(31,5);
818. printf(ANSI\_COLOR\_YELLOW "\* CAR DETAILS \*" ANSI\_COLOR\_RESET);
819. gotoxy(1,7);
820. for(i=0; i<=80; i++)
821. printf("\*");
822. gotoxy(1,8);
823. printf(" CAR ID\t\tName\t\tCapacity\tAvailable\t\tPrice");
824. gotoxy(1,9);
825. for(i=1;i<=80; i++)
826. printf("\*");
828. while(fread(&ur, sizeof(ur), 1,fp1) == 1)
829. {
830. gotoxy(5,x);
831. printf("%d", ur.car\_id);
832. gotoxy(17,x);
833. printf("%s", ur.car\_name);
834. gotoxy(35, x);
835. printf("%d", ur.capacity);
836. gotoxy(50, x);
838. if(ur.car\_id==1){
839. ur.car\_count=ur.car\_count-a;
840. printf("%d", ur.car\_count);
841. a-=a;
843. }
845. if(ur.car\_id==2){
846. ur.car\_count=ur.car\_count-b;
847. printf("%d", ur.car\_count);
848. b-=b;
849. }
851. if(ur.car\_id==3){
852. ur.car\_count=ur.car\_count-I;
853. printf("%d", ur.car\_count);
854. I-=I;
855. }
857. if(ur.car\_id==4){
858. ur.car\_count=ur.car\_count-s;
859. printf("%d", ur.car\_count);
860. s-=s;
862. }
864. if(ur.car\_id==5){
865. ur.car\_count=ur.car\_count-r;
866. printf("%d", ur.car\_count);
867. r-=r;
868. }
870. gotoxy(65, x);
871. printf("%d", ur.price);
872. x++;
873. }

876. getch();
877. fclose(fp1);
878. getch();
879. }






887. void bookedCarDetails(){
889. clrscr1();
890. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/customer.bin" , "rb");
891. Customer\_Car\_Details CC;
892. int i;
893. gotoxy(32,1);
894. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM\n" ANSI\_COLOR\_RESET);
895. for(i=1; i<=79; i++)
896. printf("\*");
897. gotoxy(32,5);
898. printf(ANSI\_COLOR\_BLUE "\* BOOKED CAR DETAILS \*" ANSI\_COLOR\_RESET);
899. gotoxy(1,7);
900. for(i=1; i<=79;i++)
901. printf("\*");
902. gotoxy(1,8);
903. printf("Model\t Cust Name\t Pick Up\t Drop\t\t S\_Date\t E\_Date");
904. gotoxy(1,9);
905. for(i=1; i<=79;i++)
906. printf("\*");
907. int x=10;
908. printf("\n\n");
909. while(fread(&CC, sizeof(Customer\_Car\_Details), 1, fp)==1){
911. gotoxy(1,x);
912. printf("%s",getCarName(CC.car\_id));
913. gotoxy(13,x);
914. printf("%s", CC.cust\_name);
915. gotoxy(27,x);
916. printf("%s",CC.pick);
917. gotoxy(44,x);
918. printf("%s",CC.drop);
919. gotoxy(58,x);
920. printf("%d/%d/%d",CC.sd.tm\_mday, CC.sd.tm\_mon, CC.sd.tm\_year);
921. gotoxy(70,x);
922. printf("%d/%d/%d",CC.ed.tm\_mday, CC.ed.tm\_mon, CC.ed.tm\_year);
923. x++;
925. }
927. fclose(fp);
928. getch();
929. }










941. int rentCar(){
943. Customer\_Car\_Details CC;
944. //char pick[30], drop[30];
945. int c,i;
946. gotoxy(32,2);
947. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM" ANSI\_COLOR\_RESET);
948. gotoxy(35,6);
949. printf(ANSI\_COLOR\_GREEN "EMPLOYEE MENU\n" ANSI\_COLOR\_RESET);
950. for(i=0;i<80;i++)
951. printf("\*");
952. gotoxy(32,8);
953. c=selectCarModel();
954. CC.car\_id=c;
955. clrscr1();
957. gotoxy(32,2);
958. printf(ANSI\_COLOR\_RED "CAR RENTAL SYSTEM" ANSI\_COLOR\_RESET);
959. gotoxy(35,6);
960. printf(ANSI\_COLOR\_GREEN "EMPLOYEE MENU\n" ANSI\_COLOR\_RESET);
961. for(i=0;i<79;i++)
962. printf("\*");
963. gotoxy(1,17);
964. for(i=0;i<80;i++)
965. printf("\*");
966. gotoxy(27,9);
967. printf(ANSI\_COLOR\_BLUE "Enter Customer Name:" ANSI\_COLOR\_RESET);
968. fflush(stdin);
969. fgets(CC.cust\_name,30,stdin);
970. char \*pos;
971. pos=strchr(CC.cust\_name,'\n');
972. \*pos='\0';
973. gotoxy(27,10);
974. printf(ANSI\_COLOR\_BLUE "Enter Pickup Point:" ANSI\_COLOR\_RESET);
975. fflush(stdin);
976. fgets(CC.pick,30,stdin);
977. pos=strchr(CC.pick,'\n');
978. \*pos='\0';
979. gotoxy(27,11);
980. printf(ANSI\_COLOR\_BLUE "Enter Drop Point:" ANSI\_COLOR\_RESET);
981. fflush(stdin);
982. fgets(CC.drop,30,stdin);
983. pos=strchr(CC.drop,'\n');
984. \*pos='\0';
985. gotoxy(27,12);
986. printf(ANSI\_COLOR\_BLUE "Enter Start Date (dd/m/yyyy):" ANSI\_COLOR\_RESET);
988. do{
990. scanf("%d/%d/%d",&CC.sd.tm\_mday,&CC.sd.tm\_mon,&CC.sd.tm\_year);
991. int datevalid=isValidDate(CC.sd);
993. if(datevalid==0){
994. gotoxy(27,18);
995. printf(ANSI\_COLOR\_RED "Wrong Date" ANSI\_COLOR\_RESET);
996. getch();
997. gotoxy(27,18);
998. printf("\b\b\b");
999. gotoxy(56,12);
1000. printf("\b\b\b");
1001. gotoxy(56,12);
1003. }
1005. else
1006. break;
1008. }while(1);
1010. gotoxy(27,13);
1011. printf(ANSI\_COLOR\_BLUE "Enter End Date (dd/m/yyyy):" ANSI\_COLOR\_RESET);
1013. do{
1015. scanf("%d/%d/%d",&CC.ed.tm\_mday,&CC.ed.tm\_mon,&CC.ed.tm\_year);
1016. gotoxy(56,13);
1017. if(CC.ed.tm\_mday>=CC.sd.tm\_mday){
1019. if(CC.ed.tm\_mon>=CC.sd.tm\_mon){

1022. if(CC.ed.tm\_year>=CC.sd.tm\_year){
1024. int datevalid=isValidDate(CC.ed);
1025. if(datevalid==0){
1027. gotoxy(27,18);
1028. printf(ANSI\_COLOR\_RED "Wrong Date !!!" ANSI\_COLOR\_RESET);
1029. getch();
1030. getch();
1031. gotoxy(27,18);
1032. printf(ANSI\_COLOR\_CYAN "Please Try Again !! " ANSI\_COLOR\_RESET);
1033. getch();
1034. gotoxy(27,18);
1035. printf("\t\t\t\t");
1036. gotoxy(61,13);
1037. printf("\b\b\b\b\b\b\b\b");
1038. gotoxy(54,13);
1040. }
1041. else
1042. break;


1046. }
1048. else
1049. {
1050. gotoxy(27,18);
1051. printf(ANSI\_COLOR\_RED "Wrong Date !!!" ANSI\_COLOR\_RESET);
1052. getch();
1053. getch();
1054. gotoxy(27,18);
1055. printf(ANSI\_COLOR\_CYAN "Please Try Again !! " ANSI\_COLOR\_RESET);
1056. getch();
1057. gotoxy(27,18);
1058. printf("\t\t\t\t");
1059. gotoxy(61,13);
1060. printf("\b\b\b\b\b\b\b\b");
1061. gotoxy(54,13);
1062. }
1064. }
1066. else
1067. {
1068. gotoxy(27,18);
1069. printf(ANSI\_COLOR\_RED "Wrong Date !!!" ANSI\_COLOR\_RESET);
1070. getch();
1071. getch();
1072. gotoxy(27,18);
1073. printf(ANSI\_COLOR\_CYAN "Please Try Again !! " ANSI\_COLOR\_RESET);
1074. getch();
1075. gotoxy(27,18);
1076. printf("\t\t\t\t");
1077. gotoxy(61,13);
1078. printf("\b\b\b\b\b\b\b\b");
1079. gotoxy(54,13);
1080. }
1082. }
1083. else
1084. {
1085. gotoxy(27,18);
1086. printf(ANSI\_COLOR\_RED "Wrong Date !!!" ANSI\_COLOR\_RESET);
1087. getch();
1088. getch();
1089. gotoxy(27,18);
1090. printf(ANSI\_COLOR\_CYAN "Please Try Again !! " ANSI\_COLOR\_RESET);
1091. getch();
1092. gotoxy(27,18);
1093. printf("\t\t\t\t");
1094. gotoxy(61,13);
1095. printf("\b\b\b\b\b\b\b\b");
1096. gotoxy(54,13);
1097. }
1099. }while(1);

1102. FILE \*fp=fopen("/Users/adarshkumar/Documents/Project/C/Car\_Rental\_System/customer.bin" , "ab");
1103. fwrite(&CC, sizeof(Customer\_Car\_Details), 1, fp);
1104. printf("\nPress any key to Contiue ...");
1105. getch();
1106. getch();
1107. fclose(fp);
1108. updateCarCount(c);
1109. bookedCarDetails();
1111. return 1;

1114. }

Adarsh.c

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include"adarsh.h"
4. #include"car.h"
5. #include <termios.h>
6. #include <unistd.h>
7. #define BLUE(string) "\x1b[34m" string "\x1b[0m"
8. #define RED(string) "\x1b[31m" string "\x1b[0m"
9. #define ANSI\_COLOR\_RED "\x1b[31m"
10. #define ANSI\_COLOR\_GREEN "\x1b[32m"
11. #define ANSI\_COLOR\_YELLOW "\x1b[33m"
12. #define ANSI\_COLOR\_BLUE "\x1b[34m"
13. #define ANSI\_COLOR\_MAGENTA "\x1b[35m"
14. #define ANSI\_COLOR\_CYAN "\x1b[36m"
15. #define ANSI\_COLOR\_RESET "\x1b[0m"
17. void gotoxy(int x, int y) {
18. printf("\x1b[%d;%df", y, x);
19. }

22. void clrscr1(){
23. int i;
24. for (i=0; i < 1; i++)
25. {
27. printf("\e[1;1H\e[2J");
28. }
29. }



34. int myAtoi(char\* str)
35. {
36. int res = 0; // Initialize result
38. // Iterate through all characters of input string and
39. // update result
40. for (int i = 0; str[i] != '\0'; ++i)
41. res = res \* 10 + str[i] - '0';
43. // return result.
44. return res;
45. }




51. int getch() {
52. struct termios oldtc;
53. struct termios newtc;
54. int ch;
55. tcgetattr(STDIN\_FILENO, &oldtc);
56. newtc = oldtc;
57. newtc.c\_lflag &= ~(ICANON | ECHO);
58. tcsetattr(STDIN\_FILENO, TCSANOW, &newtc);
59. ch=getchar();
60. tcsetattr(STDIN\_FILENO, TCSANOW, &oldtc);
61. return ch;
62. }

Car.h

1. #ifndef CAR\_H\_INCLUDED
2. #define CAR\_H\_INCLUDED
3. #include<time.h>
4. #include <termios.h>
5. #include <unistd.h>
6. #define BLUE(string) "\x1b[34m" string "\x1b[0m"
7. #define RED(string) "\x1b[31m" string "\x1b[0m"
8. #define ANSI\_COLOR\_RED "\x1b[31m"
9. #define ANSI\_COLOR\_GREEN "\x1b[32m"
10. #define ANSI\_COLOR\_YELLOW "\x1b[33m"
11. #define ANSI\_COLOR\_BLUE "\x1b[34m"
12. #define ANSI\_COLOR\_MAGENTA "\x1b[35m"
13. #define ANSI\_COLOR\_CYAN "\x1b[36m"
14. #define ANSI\_COLOR\_RESET "\x1b[0m"
16. struct User{
18. char userid[20], pwd[20], name[20];
19. };
21. struct Car{
22. int car\_id, capacity, car\_count, price;
23. char car\_name[50];
24. };
26. struct Customer\_Car\_Details {
28. int car\_id;
30. char cust\_name[30], pick[30], drop[30];
32. struct tm sd,ed;
34. };

37. typedef struct User User;
38. typedef struct Car Car;
39. typedef struct Customer\_Car\_Details Customer\_Car\_Details;


43. void sub\_str(char \*, char\*, char);
45. void addAdmin();
47. User\* getInput();
49. int checkUserExist(User, char\*);
51. int adminMenu();
53. int empMenu();
55. void addEmployee();
57. void viewEmployee();
59. void addCarDetails();
61. void showCarDetails();
63. void availableCarDetails();
65. int deleteEmp();
67. int deleteCarModel();

70. int rentCar();
72. int selectCarModel();
74. void updateCarCount(int);
76. void bookedCarDetails();
78. char \* getCarName(int);
80. int isValidDate(struct tm);
81. #endif // CAR\_H\_INCLUDED

Adarsh.h

1. #ifndef ADARSH\_H\_INCLUDED
2. #define ADARSH\_H\_INCLUDED
4. #define ANSI\_COLOR\_RED "\x1b[31m"
5. #define ANSI\_COLOR\_GREEN "\x1b[32m"
6. #define ANSI\_COLOR\_YELLOW "\x1b[33m"
7. #define ANSI\_COLOR\_BLUE "\x1b[34m"
8. #define ANSI\_COLOR\_MAGENTA "\x1b[35m"
9. #define ANSI\_COLOR\_CYAN "\x1b[36m"
10. #define ANSI\_COLOR\_RESET "\x1b[0m"
12. void gotoxy(int , int );
14. void clrscr1();
16. int getch();
18. int myAtoi(char\* str);


22. #endif // ADARSH\_H\_INCLUDED

Main.c

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include"car.h"
4. #include"adarsh.h"
5. #include <termios.h>
6. #include <unistd.h>
7. #define BLUE(string) "\x1b[34m" string "\x1b[0m"
8. #define RED(string) "\x1b[31m" string "\x1b[0m"
9. #define ANSI\_COLOR\_RED "\x1b[31m"
10. #define ANSI\_COLOR\_GREEN "\x1b[32m"
11. #define ANSI\_COLOR\_YELLOW "\x1b[33m"
12. #define ANSI\_COLOR\_BLUE "\x1b[34m"
13. #define ANSI\_COLOR\_MAGENTA "\x1b[35m"
14. #define ANSI\_COLOR\_CYAN "\x1b[36m"
15. #define ANSI\_COLOR\_RESET "\x1b[0m"
17. int main()
18. {
19. gotoxy(25,10);
20. printf(RED("WELCOME TO CAR RENTAL SYSTEM") );
21. gotoxy(20,13);
22. printf(ANSI\_COLOR\_GREEN "\* RENT A CAR AND GO WHEREVER YOU NEED \*" ANSI\_COLOR\_RESET "\n");
23. addAdmin();
24. getch();

27. User \*usr; //User
28. int result,result1;
29. int i,type;
30. int choice;
32. while(1){
34. clrscr1();
35. gotoxy(32,2);
36. printf(RED("CAR RENTAL SYSTEM"));
38. //UPPER LINE
40. gotoxy(1,8);
41. for( i=0; i<80; i++)
42. printf("\*");
44. //LOWER LINE
46. gotoxy(1,17);
47. for(i=0; i<80; i++)
48. printf("\*");
50. gotoxy(32,10);
51. printf("\033[22;34m1. ADMIN \033[0m");
52. gotoxy(32,12);
53. printf("\033[22;34m2. EMPLOYEE \033[0m");
54. gotoxy(32,14);
56. int k;
58. printf("Select Your Role :");
60. do{
62. scanf("%d",&type);
64. k=0;
66. if(type==1){

69. do{
71. usr=getInput();
72. if(usr!=NULL){
74. //Code for Validating
75. k=checkUserExist(\*usr, "admin");
77. }
79. else{
81. break;
82. }
84. }while(k==0);
86. if(k==1){
88. gotoxy(30,14);
89. printf(ANSI\_COLOR\_GREEN "Login Accepted!" ANSI\_COLOR\_RESET );
90. gotoxy(1,20);
91. printf("Press any key to continue");
92. getch();

95. while(1){
97. clrscr1();
98. choice=adminMenu();
99. if(choice==7){
100. clrscr1();
101. break;
102. }
104. switch(choice){
106. case 1:
107. clrscr1();
108. addEmployee();
109. break;
110. case 2:
111. clrscr1();
112. addCarDetails();
113. break;
114. case 3:
115. clrscr1();
116. viewEmployee();
117. break;
118. case 4:
119. clrscr1();
120. showCarDetails();
121. break;
122. case 5:
123. clrscr1();
124. result=deleteEmp();
125. if(result==0){
126. gotoxy(15,14);
127. printf("Sorry! No Employee Found with the given Employee ID");
128. printf("\n\nPress Any Key to go back to the Main Menu");
129. getch();
130. }
131. else if(result==1){
132. gotoxy(25,14);
133. printf(ANSI\_COLOR\_GREEN "Record Deleted Successfully" ANSI\_COLOR\_RESET);
134. printf("\n\nPress Any Key to go Back to the Main Menu");
135. getch();
137. }
138. break;
139. case 6:
140. clrscr1();
141. result1=deleteCarModel();
142. if(result1==0){
143. gotoxy(15,14);
144. printf("Sorry! No Car Found with the given Car ID");
145. printf("\n\nPress Any Key to go back to the Main Menu");
146. getch();
147. }
148. else if(result1==1){
149. gotoxy(25,14);
150. printf(ANSI\_COLOR\_GREEN "Record Deleted Successfully" ANSI\_COLOR\_RESET);
151. printf("\n\nPress Any Key to go Back to the Main Menu");
152. getch();
154. }
156. break;
158. default:
159. printf("Invalid Input");
160. getch();
162. } //Switch Close
164. }
166. }

169. }

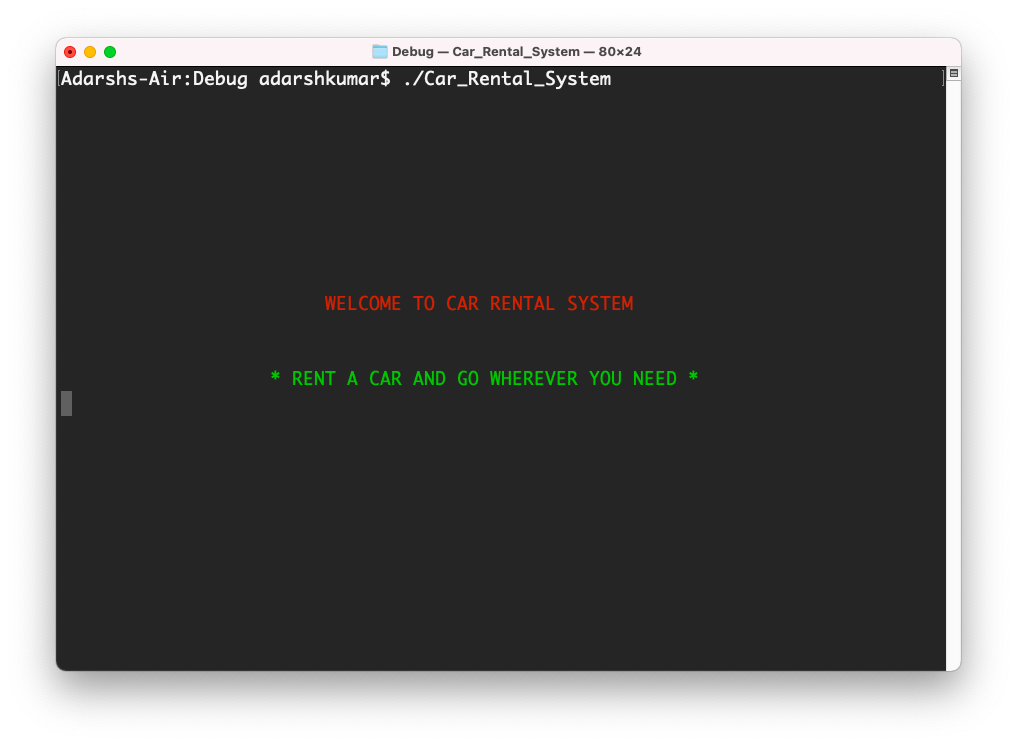
172. else if(type==2){
174. do{
176. usr=getInput();
177. if(usr!=NULL){
179. k=checkUserExist(\*usr, "emp");
180. }
181. else{
183. break;
184. }
185. }while(k==0);

188. if(k==1){
190. gotoxy(30,14);
191. printf(ANSI\_COLOR\_GREEN "Login Accepted" ANSI\_COLOR\_RESET);
192. gotoxy(1,20);
193. printf("Press Any Key TO Continue");
194. getch();
196. while(1){
198. clrscr1();
199. choice=empMenu();
200. if(choice==5){
202. clrscr1();
203. break;
204. }
205. switch(choice){
207. case 1:
208. clrscr1();
209. int j;
210. do{
212. clrscr1();
213. j=rentCar();
214. if(j==0)
215. printf("Booking Cancelled\nTry Again");
216. getch();
217. }while(j==0);
218. getch();
219. break;
220. case 2:
221. clrscr1();
222. bookedCarDetails();
223. getch();
224. break;
225. case 3:
226. clrscr1();
227. availableCarDetails();
228. break;
229. case 4:
230. clrscr1();
231. showCarDetails();
232. break;
233. default:
234. printf("Incorrect Choice:");

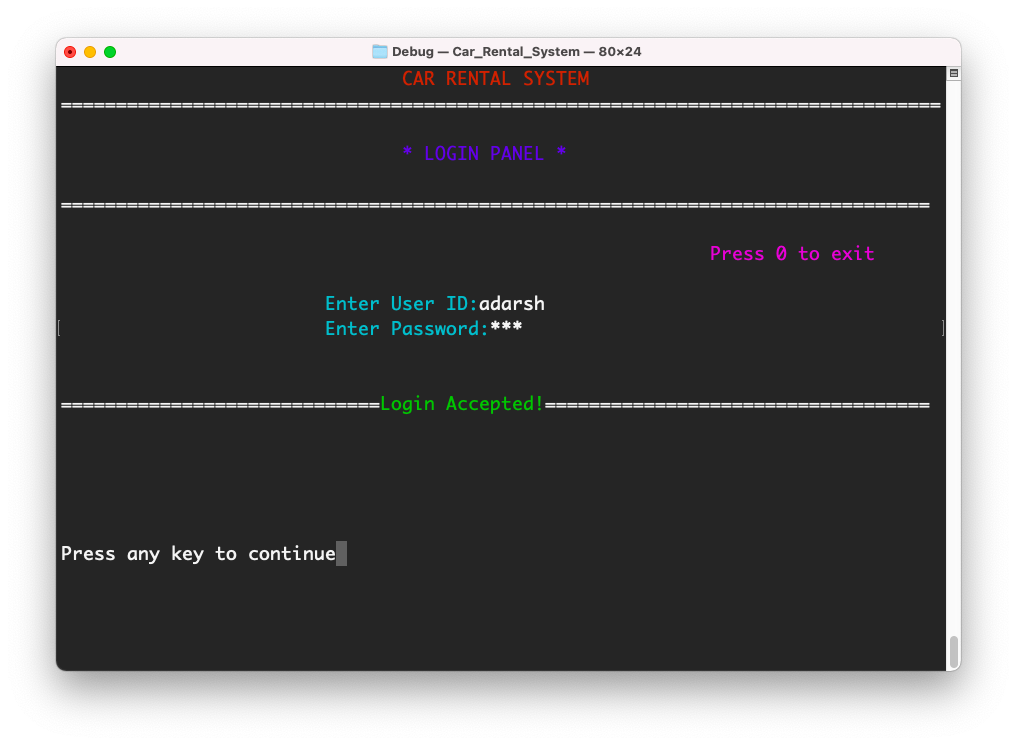
237. } //Switch
239. }
240. }
242. }

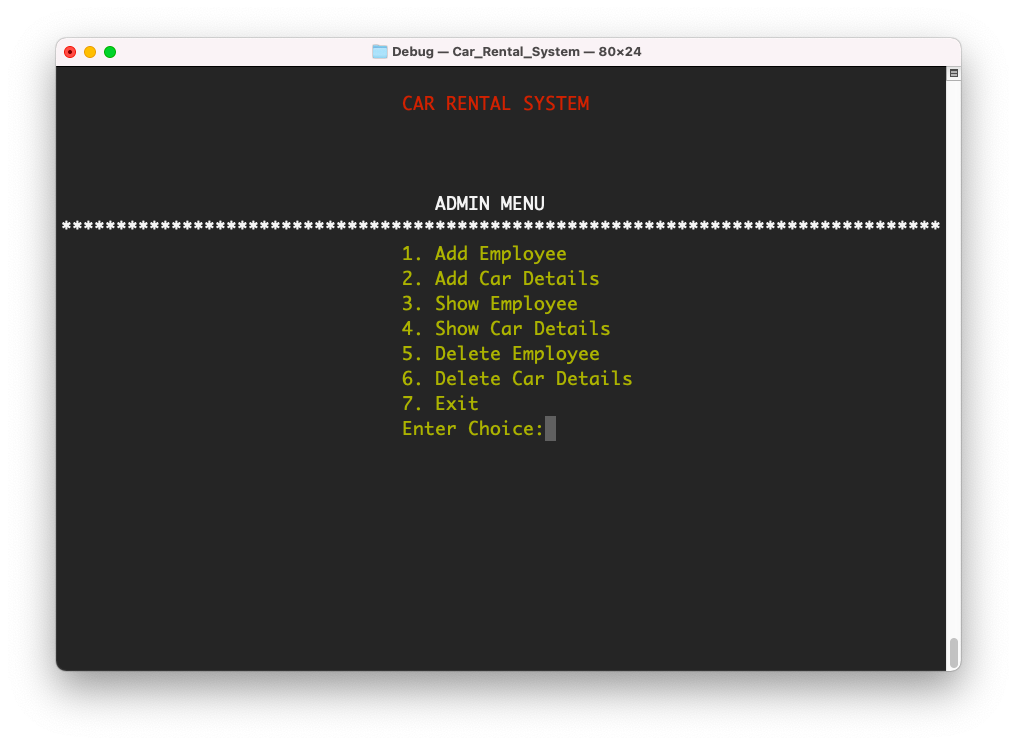
245. else{
247. gotoxy(30,20);
248. printf(RED("INVALID USER TYPE") );
249. getch();
250. gotoxy(30,20);
251. printf("\t\t\t");
252. gotoxy(50,14);
253. printf("\t");
254. gotoxy(50,14);
256. }
258. }
259. while(type!=1 && type!=2);
260. }
261. return 0;
262. }

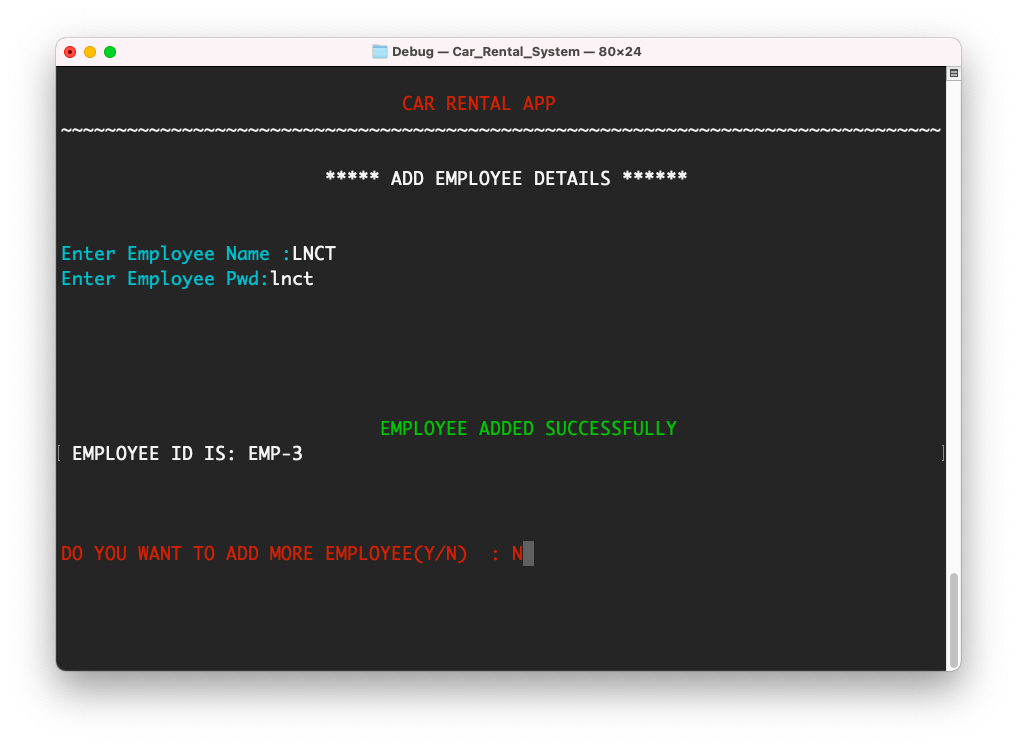
**OUTPUT SCREENS**

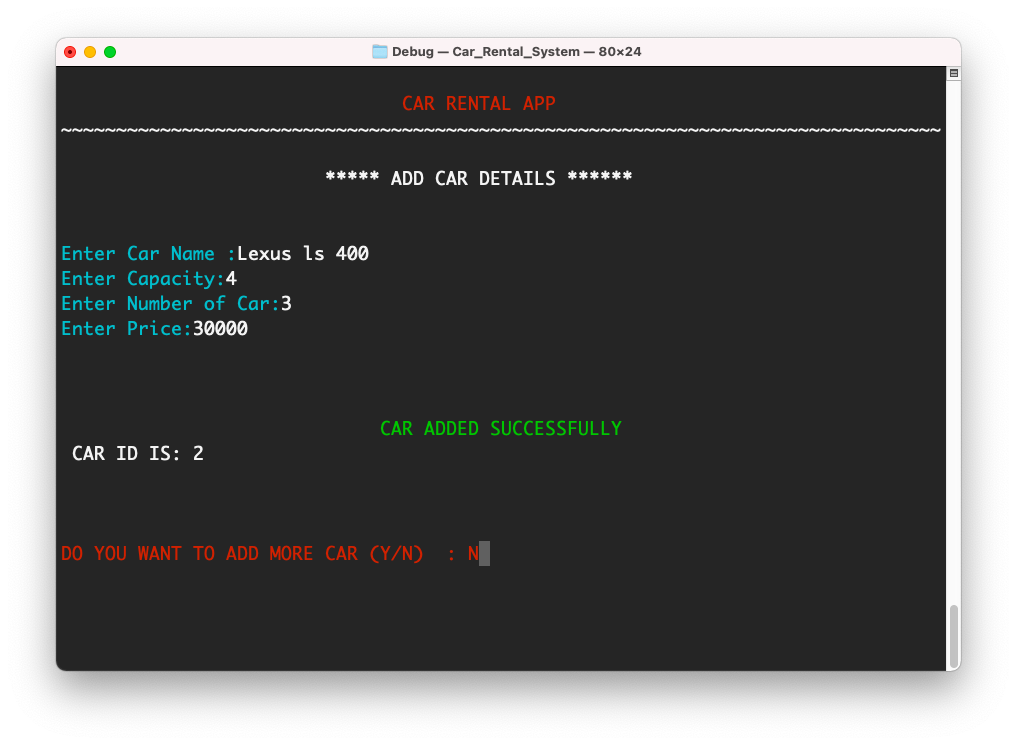
****

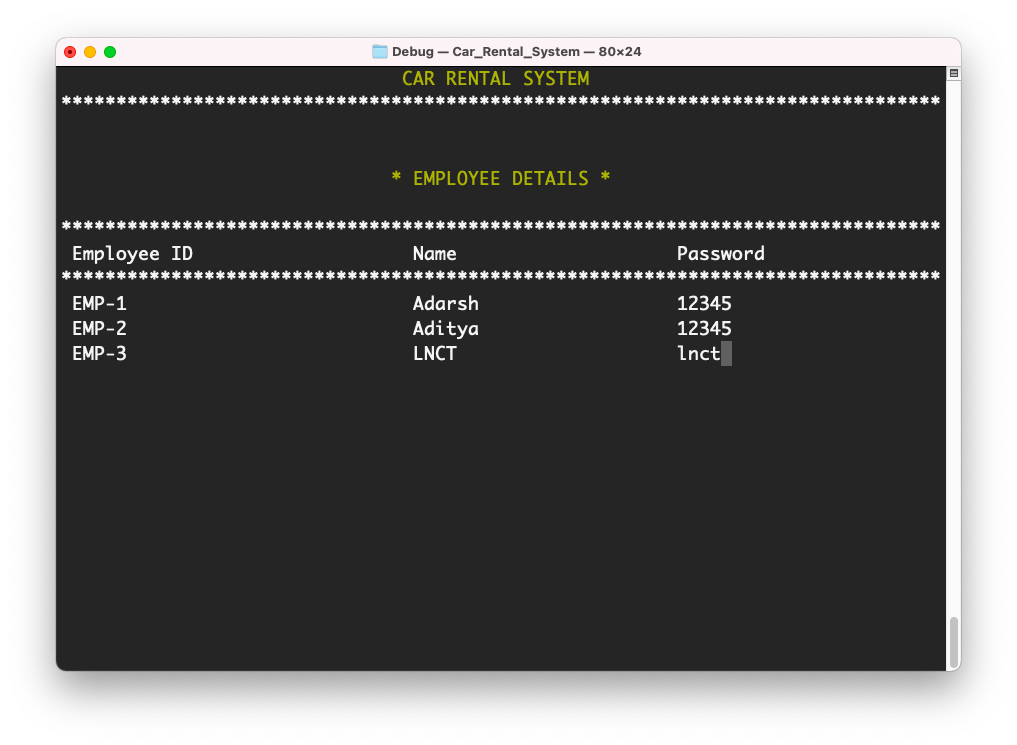




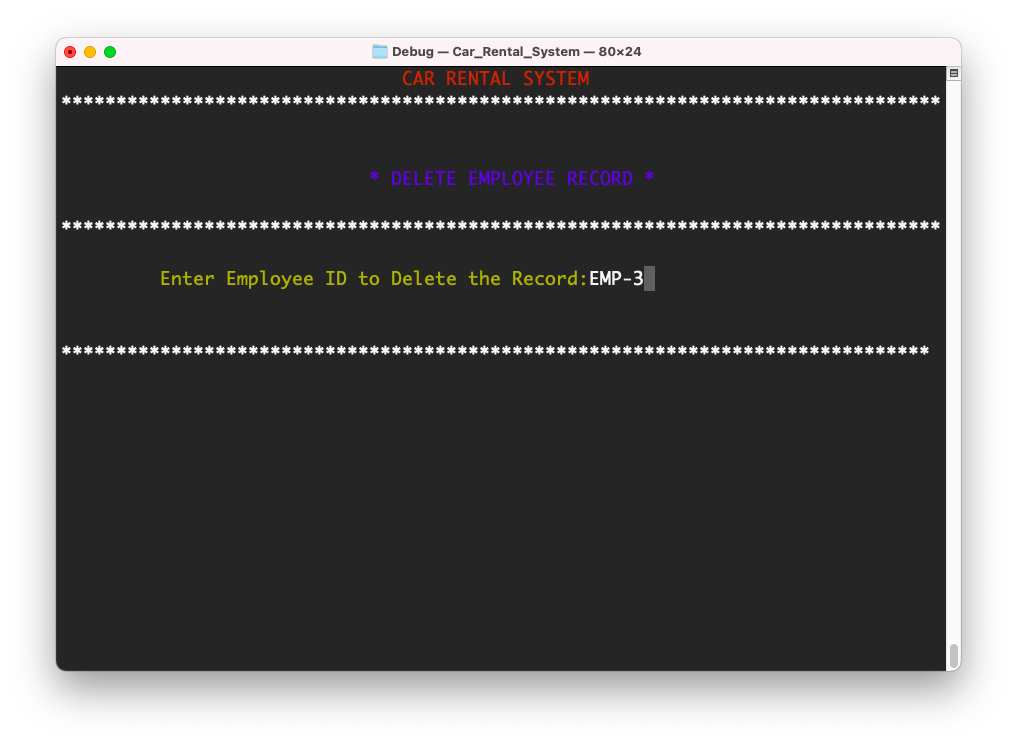




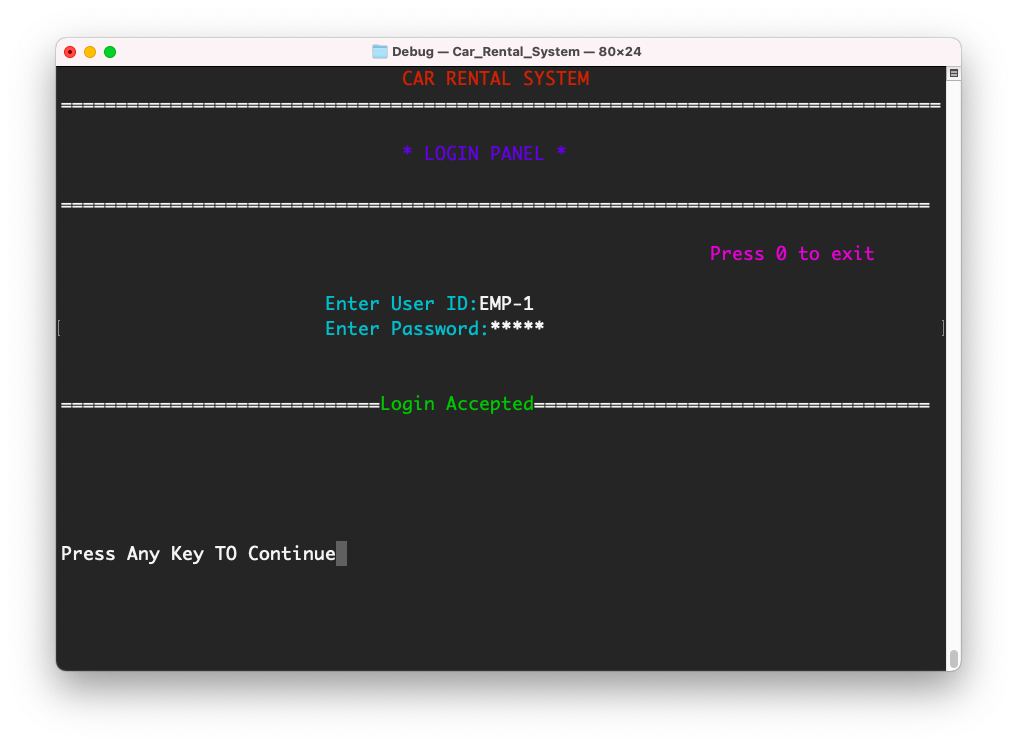


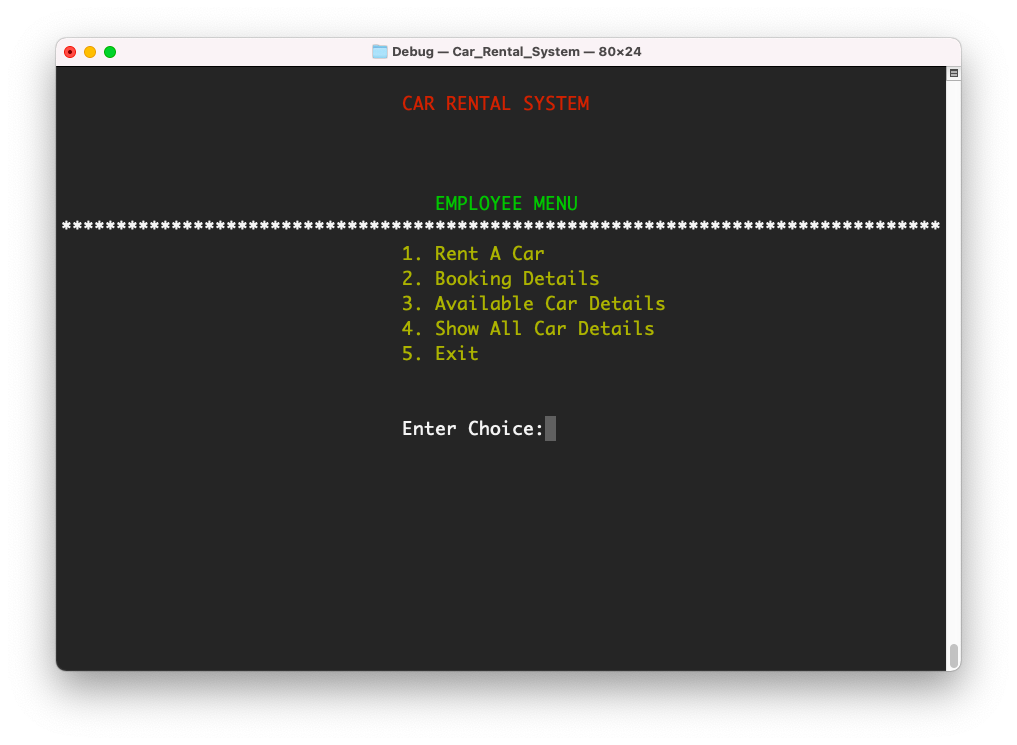


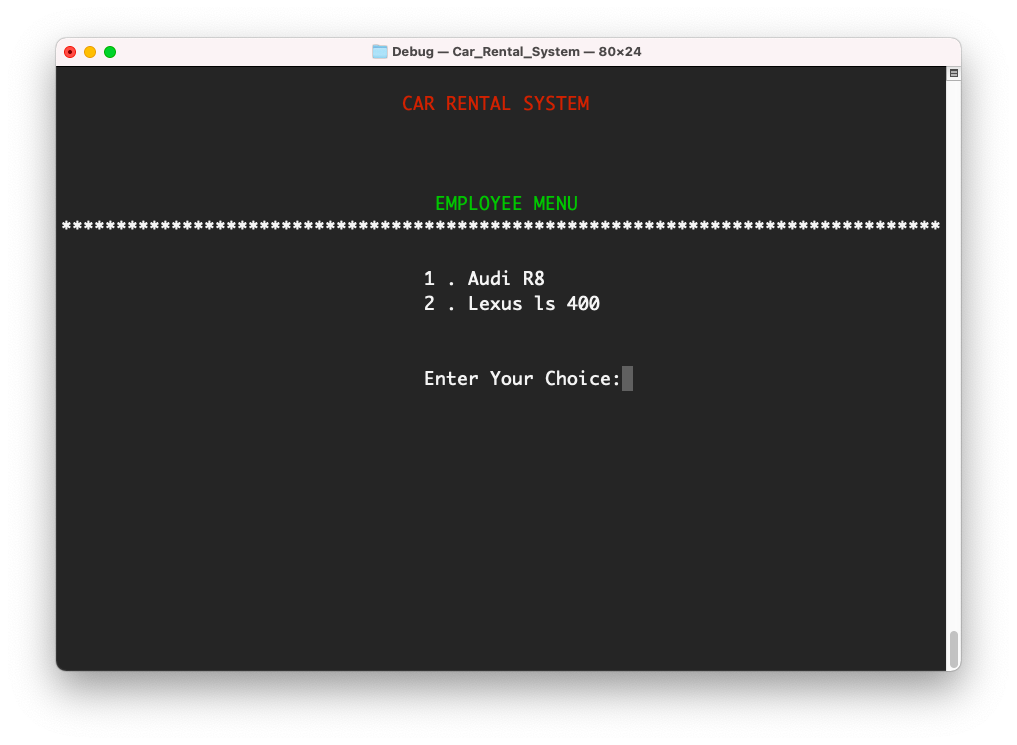


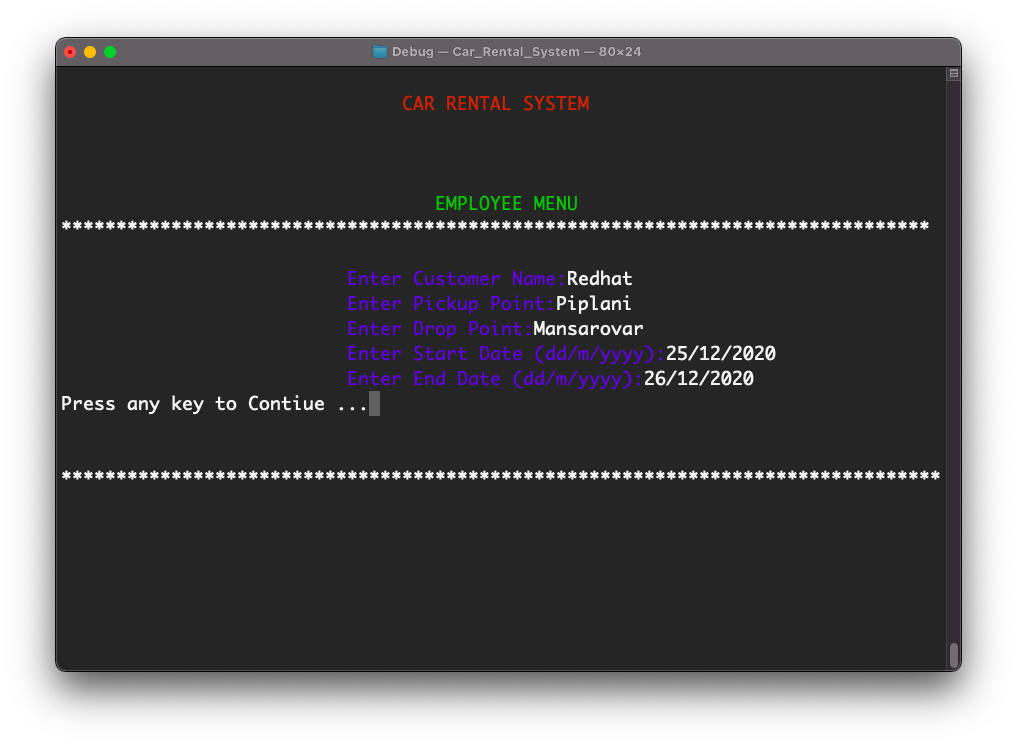
















**Conclusion :**

* Every Work at the present time all the work performs by computerize and by this application our work is converted to totally computerized.
* So that by this application we save our time & paper works.
* Due to the advent of computer, it has become very easy for Add Records, know the current available cars for rent, Adding New Employees, Deleting Employee and Maintain the record.

**FUTURE ENHANCEMENT**

* **We provide the facility of Calculating the dues of the customers.**
* **Convert this console application to an Android Application .**

**Appendix A: Glossary**

TERMS

          All the terms and abbreviations in the project are specified clearly. For further development of project evolved definitions will be specified

ACRONYMS

         IEEE:   Institute of Electrical and Electronics Engineers

UML : Unified Modeling Language

C : C Programming Language

CLI :Command Line Interface

**Appendix B: Analysis Models**

This includes all the pertinent analysis models, such as data flow diagrams, class diagrams, use case diagrams, interaction diagrams and state-chart diagrams.

**Appendix C: To Be Determined List**

**References**

**Under this references section, we have mentioned various references from which we collected our problem and several others that supported us to design the solution for our problem. These references include either books, papers published through some standards and several websites links with URL’s: -**

* **For the complete reference and understanding of adarsh.h** [**https://www.stackoverflow.com**](https://www.stackoverflow.com)

**BIBILOGRAPHY**



* **C Programming Language**

By Dennis M. Ritchie