

# **BIKE RENTAL SYSTEM**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the  
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

**SAGI HASINI - 1602-19-737-133**

**SHRAVANI MAHANTH – 1602-19-737-169**



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



## **DECLARATION BY THE CANDIDATE**

We, SAGI HASINI and SHRAVANI MAHANTH bearing hall ticket numbers, 1602-19-737-133 and 1602-19-737-169 , hereby declare that the project report entitled “BIKE RENTAL SYSTEM” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**SAGI HASINI**

**1602-19-737-133**

**SHRAVANI MAHANTH**

**1602-19-737-169**

(Faculty In-Charge)

(Head.Dept IT)

## **Acknowledgements**

The satisfaction that accompanies the successful completion of this project would not be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of this project.

We convey thanks to my project guide Mrs .Divya of Information technology Department for providing encouragement, constant support and guidance which was of a great help to complete this project successfully.

Last but not the least, we wish to thank our parents for financing our studies in this privileged Vasavi College of Engineering as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

## **Abstract**

Vehicle Rental System is based on a concept to maintain rental for a short period of time and generating total charge. Before stepping into the main system a user has to pass through a login system to get access, then a menu displays with the selective option of the Car models for rent. There are different models in this system which can be changed by the user according to their will. Talking about the features of this Mini Project before hiring a vehicle he/she has to choose the vehicle, each of the vehicle has its own rate, maximum speed, color, and model. After selecting a vehicle, the system wants a confirmation before proceeding. Afterward, the user has to provide his/her Personal details such as Name (first name without spaces), Citizen-ship Number and provide Payment Amount. If the amount provided by the user is less than the vehicle rate then the payment won't proceed for the rental. After all these procedures, the user can continue hiring more vehicles or either exit the system .

## **Table of contents**

<b>Sl.no</b>	<b>topic</b>	<b>page</b>
1.	Introduction	1-2
2.	Technology	3
3.	Proposed work	4-64
4.	Result	65-91
5.	Additional knowledge	92
6.	Conclusion and Future work	93
7.	References	94

## **Introduction**

Bike rental system is for those who are you planning to explore the city ,who is in a need of a two wheeler for emergency purpose ,who needs a vehicle for a short tour/vacation. It presents Bike rentals to make customer's job easy it offers to Enjoy the freedom of owning a two-wheeler without the hefty down-payments, EMIs and paperwork.

Firstly the user has to enter some details in order to rent a bike. It involves entering a valid driving license number , email id ,phone number and nationality as well . We believe that each one of us is unique. Being unique means we all have different likes and needs .So, we went out of our way to get something for everyone.

All you have to do is make up your mind and pick one! It doesn't matter if you're a college student, a working professional or a bike enthusiast who just wants to ride around the city for the weekend, pick something that suits your lifestyle and Wallet as well.

Initially user has to register if he/she is new to the system else they can simply sign-in . After registration is done user has to sign-in to the account to rent a bike.

Here we are trying to avoid taking the user's details multiple times if they already have an account. We offer a wide variety of vehicles starting from scooters such as Activa, Dio, luna, maestro, NS200,R15,access,pulsar, KTM Duke 200,etc.

We have access for both admin and user side .

### **USER SIDE:**

- 1.Rent a Bike
- 2.View rental fares
- 3.View rental records

4.Search rental record

5.Exit

Firstly ,for user side customer can Rent a bike where he/she can check for rental fares and choose one of his choice and can generate a bill.

Also, user/customer can search and view for rental records .If the stock of the bike chosen isn't available then customer will have to choose another one!

### **ADMIN SIDE:**

1.View rental records

2.View vehicle fares

3.Edit fares and stocks

4.Remove a user

5.Search user

6.Edit user

7.Exit

Firstly ,admin has to sign-in to the account with their credentials which are hardcoded (user name : bikerentalsystem ,password: miniproject).Then admin can view rental records and vehicle rental fares as well. If the user choose a specified bike then admin will have to change the stock of bikes available for next use. Also, he can change the bike fares.

Admin can remove the user if they return the bike and can also search the users . Admin can also edit the users details and fix them accordingly .Finally Admin can exit from the page and can view home page.

## **Technology**

Computer technology consists of the hardware of computers and computer-controlled devices and software like the operating systems, authoring tools, expert systems, and courseware use to support training technology. Many projects, for-profit organizations often require technology development involving hardware or software design.

### **Software requirements**

Software requirements deal with defining software resource requirements that need to be installed on the computer. These prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The following sub-sections discuss the various aspects of software requirements for bike rental system.

- Web Browser: Microsoft Internet Explorer, Mozilla, Google Chrome or later
- Operating System: Windows XP / Windows7/ Windows Vista

### **Hardware requirements**

The **hardware requirements** are the requirements of a hardware device. Most hardware only has operating system requirements or compatibility.

The following sub-sections discuss the various aspects of hardware requirements for bike rental system.

- 256 MB RAM
- 1 Gb hard free drive space
- Processor: Intel Core i5 and more



## Proposed work

### a. Design

#### i. Use case diagram and descriptions for all the use-cases

### USE CASES

#### *USER:*

- ✓ Register
- ✓ Login
- ✓ View Vehicles Fares
- ✓ Rent a bike
- ✓ View Rental Records
- ✓ Search Rental Records
- ✓ Logout

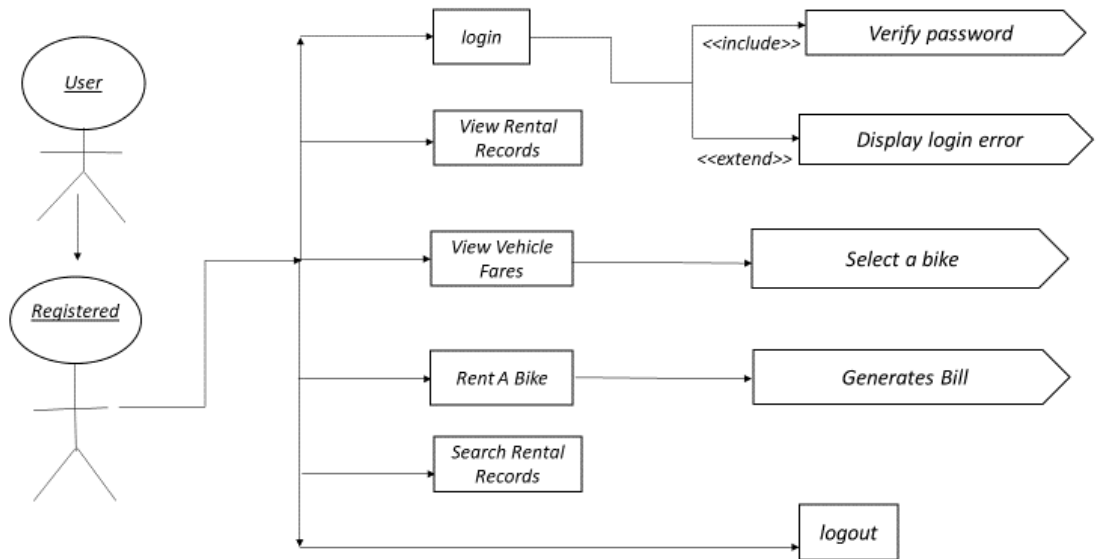
#### *Admin:*

- ✓ Login
- ✓ View Rental Records
- ✓ View Vehicles Fares
- ✓ Edit Fares and stock
- ✓ Remove a User
- ✓ Search a User
- ✓ Edit Records
- ✓ Exit

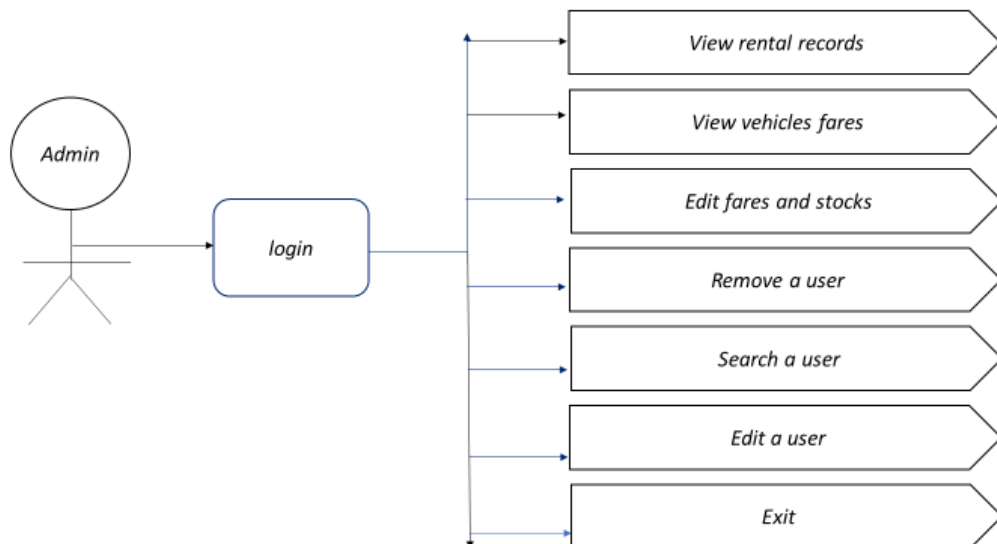
#### *DATA:*

- ✓ Stores the information .  
About different models of  
bikes
- ✓ Deletion and editing of  
Users details.

## USER – Use Case Diagram



## ADMIN – Use Case Diagram



### USER

#### Register

User has to registers to login ,To register user has to enter credentials (creates username and password to login and full name ,address ,contact number ,nationality ,email ,

driving license number). System Saves all the credentials entered by the user and this details are used to login.

## **Login**

User has to Enters credentials (username and password) to login , if the credentials are correct user can login successfully and gets access to program. If not user has to enter the credentials until they login.

## **View vehicle fares**

Before renting the bike user can see the available stocks of bikes and the fares of each bike so that user can choose the bike available and can rent the bike.

## **Rent a bike**

After choosing the bike in view vehicle fares now user can rent a bike of their valid choose and can generate a bill.

## **View Rental Records**

User can view the details of their rented bike i.e; (Bike name ,Full name , address , phone number , nationality , email, days, rented time).

## **Search Rental Records.**

User can search their rental records by entering the bike name.

## **Exit**

User can exit from the page after renting the bike they wanted and they are redirected to welcome page.

## **Admin**

### **Login**

Admin must login with their credentials (username and password) which are hard-coded(user name: bikerentalsystem , password : miniproject) to get access to program If not admin has to enter the credentials until they login.

## **View Rental Records**

Admin can view the details of all the rented bikes.

## **Edit fares and stocks**

Admin can edit the bikes fares and the count of bikes if they needed.

## **Remove a user**

Admin can remove the user details by entering username of the customer . If he/she returns the bike after their time duration.

## **Search a user**

Admin can search the rental records by entering user name and full name.

## **Edit user**

Admin can edit the user details if there is any mistake in the details.

## **Exit**

Admin can exit from the page so that admin can redirected to the home page.

## **b . IMPLEMENTATION**

### **i . Module wise code for the entire project**

```
int main()
{
    system("color 70");
    WelcomePage();
}
```

**FUNCTION NAME** : void Title()

```
void Title()
{
```



# WELCOME TO BIKE RENTAL SYSTEM

[illegible]

default:

```
printf("\nINVALID CHOICE!!!");
```

```
break;
```

$$\}$$
$$\}$$

**FUNCTION NAME** : void registerMenu()

```
void registerMenu()
```

$$\{$$

```
system("cls");
```

```
char user_name[30], password[20], UName[20], Pass[20];
```

```
int valid=0;
```

FILE \*usp, \*read;

```
usp = fopen("Register.txt", "a+");
```

```
read = fopen("Register.txt", "r");
```

```
printf("\n\n\n\t\t\t----- SIGNUP -----
```

-----\n");

A:

```
printf("\n\t\t\t\t\tENTER USERNAME : ");
```

```
gets(user_name);
```

```
while (fscanf(read, "%s %s %s %s %s %s %s %s\n",
```

```
s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.license)
e) != EOF)
```

 $\{$ 

```
if (strcmp(s.username, user_name) == 0)
```





```

    }

    if(!valid)

    {

        printf("\n\t\t\t\t\t Full Name contains an Invalid character...Enter
again.");

        goto B;

    }

C:

printf("\n\t\t\t\t\t ENTER ADDRESS : ");

gets(s.address);

s.address[0]=toupper(s.address[0]);

if(strlen(s.address)>30||strlen(s.address)<5)

{

    printf("\n\t\t\t\t\t Invalid...The address should be in range of 5 to
30");

    goto C;

}

D:

printf("\n\t\t\t\t\t Contact no: ");

gets(s.phonenumber);

if(strlen(s.phonenumber)>10||strlen(s.phonenumber)!=10)

{

    printf("\n\t\t\t\t\t Invalid...Contact must contain 10 digits.");

    goto D;

```

```
}
```

E:

```
printf("\n\t\t\t\t\tENTER NATIONALITY : ");  
gets(s.nationality);
```

F:

```
printf("\n\t\t\t\t\tENTER EMAIL : ");  
gets(s.email);
```

G:

```
printf("\n\t\t\t\t\tENTER DRIVING LICENSE NUMBER : ");  
gets(s.license);
```

```
fprintf(usp, "%s %s %s %s %s %s %s %s\n",  
user_name,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.license);
```

```
printf("\n\n\t\t\t----- ACCOUNT  
SUCCESSFULLY CREATED -----\\n");
```

```
getch();
```

```
fclose(usp);
```

```
WelcomePage();
```

```
}
```

**FUNCTION NAME** : void login()

```
void login()
```

```
{
```

```
Title();
```

```

int a = 0, i = 0, counter = 0, flag = 0;

char ch;

char un[30],pw[30];

FILE *fp;

fp = fopen("Register.txt", "r");

system("cls");

printf("\n\n\t\t\t\t\t***** LOGIN FORM
*****");

printf("\n\n\n\t\t\t\t\tENTER USERNAME:- ");

gets(un);

printf("\n\n\n\t\t\t\t\tENTER PASSWORD:- ");


while (ch != 13)

{

    ch = getch();

    if (ch != 13 && ch != 8)

    {

        printf("*");

        pw[i] = ch;

        i++;

    }

    else if (ch == 8)

    {

        if (i > 0)

```



[illegible]

**FUNCTION NAME** : void loginForAdmin()

```
void loginForAdmin()
```

```
{

    system("cls");

    char username[] = "bikerentalsystem";

    char pw[] = "miniproject";

    char UName[20], password[20], ch;

    int i, flag = 0, counter = 0;

    printf("\n\n\n");

    Title();

    printf("\n\n\n\n\t\t\t\t***** LOGIN FOR ADMIN
*****");

    printf("\n\n\n\t\t\t\t\tENTER USERNAME:- ");

    gets(UName);

    printf("\n\n\t\t\t\t\tENTER PASSWORD:- ");

    gets(password);

    if (strcmp(username, UName) == 0)

    {

        flag = 1;

        if (strcmp(password, pw) == 0)

        {

            printf("\n\n\n\t\t\t\t\t----- LOGIN SUCCESSFUL -----
---");
```







```

printf("\n\n\t\t\t\t\t ----->>>> |MAIN MENU| <<<<-----
----- \n");

printf("\n\t\t\t\t\t");

for (i = 0; i < 80; i++)

    printf("-");

printf("\n");

printf("\t\t\t\t\t *Select and Enter Your Choice From Menu*");

printf("\n\n");

printf("\n\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t Enter 1 ->> Rent a bike");

printf("\n\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t Enter 2 ->> View Vehicle Fares");

printf("\n\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t Enter 3 ->> View Your Rental Record");

printf("\n\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t Enter 4 ->> Search Your Rental Record");

printf("\n\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t Enter 5 ->> Logout");

printf("\n\t\t\t\t\t=====");

```

```

printf("\n \n\t\t\t\t\t");

for (i = 0; i < 80; i++)

    printf(">");

printf("\n\t\t\t\t\tCurrent date and time : %s", ctime(&t));

printf("\t\t\t\t\t");

for (i = 0; i < 80; i++)

    printf("<");

scanf("%d", &choice);

fflush(stdin);

switch (choice)
{

    case 1:

        RentBike(username);

        break;

    case 2:

        ViewFares();

        break;

    case 3:

        UserRentedList(username);

        break;

    case 4:

        UserSearch(username);

        break;
}

```

```

        case 5:

            system("cls");

            WelcomePage();

            break;

        default:

            system("cls");

            printf("Incorrect Input");

            printf("\n Press any key to continue");

            getch();

            break;

    }

}

```

**FUNCTION NAME** : void RentBike()

```

void RentBike(char username[30])

{

    FILE *f,*fp,*fp1,*fp2;

    time_t t;

    time(&t);

    int i, n, valid = 0, flag = 0;

    char test, bikename[50];

    char vehicle[30], price[7], stock[7],stockLeft1[7],time[50];

    char tempBike[20];

```

bikename:

24

```

        if (strcmp(tempBike, s.bikename) == 0)
        {
            flag = 1;
        }
    }

    fclose(fp);

    if (flag != 1)
    {
        printf("\n\t\t\t\t\t PLEASE ENTER A VALID BIKE NAME!");

        goto bikename;
    }

    fp = fopen("viewFares.txt", "r");
    fp2 = fopen("newFares.txt", "w+");

    flag = 0;

    while (fscanf(fp, "%s %s %s\n", vehicle, price, stock) != EOF)
    {
        if (strcmp(vehicle, s.bikename) == 0)
        {
            stockLeft = atoi(stock);

            if(stockLeft>0)
            {
                flag = 1;

                stockLeft--;
            }
        }
    }

```









```

        {

            bill = atoi(tempPrice) * period;

        }

    }

    system("cls");

    printf("\n\t\t\t\t|=====
=====|");

    printf("\n\t\t\t\t|          Bike Rental - Customer Invoice
|");

    printf("\n\t\t\t\t|=====
=====|");

    printf("\n\t\t\t\t\t > USER NAME          :      %s\n", username);

    printf("\n\t\t\t\t\t > CUSTOMER NAME          :      %s\n", s.fullname);

    printf("\n\t\t\t\t\t > BIKE MODEL            :      %s\n", s.bikename);

    printf("\n\t\t\t\t\t > ADDRESS              :      %s\n", s.address);

    printf("\n\t\t\t\t\t > PHONENUMBER          :      %s\n",
s.phonenumber);

    printf("\n\t\t\t\t\t > NATIONALITY          :      %s\n", s.nationality);

    printf("\n\t\t\t\t\t > EMAIL                :      %s\n", s.email);

    printf("\n\t\t\t\t\t > NUMBER OF DAYS        :      %i\n", s.period);

    printf("\n\t\t\t\t\t > RENTED DATE & TIME    :      %s\n", ctime(&t));

    printf("\n\t\t\t\t\t_____
_____ \n");

    printf("\n\t\t\t\t\t > TOTAL RENTAL AMOUNT    :      Rs. %f", bill);

```



```

while (fscanf(fp,"%s %s %s\n", vehicle,price,stock) != EOF)
{
    gotoxy(55,y); printf("%s",vehicle);

    gotoxy(75,y); printf("%s",price);

    gotoxy(95,y); printf("%s",stock);

    y++;

}

fclose(fp);

printf("\n\n\n\t\t\t\t\tENTER ANY KEY TO CONTINUE");

getch();

}

```

**FUNCTION NAME** : void UserRentedList(char[])

```
void UserRentedList(char username[30])
```

```

{
    char user_name[50],time[50];

    FILE *f,*fp;

    int i, q = 9;

    fp = fopen("Rentals.txt","r");

    Title();

    system("cls");

    gotoxy(1, 6);

    printf("BIKENAME ");

    gotoxy(15, 6);

```

```

printf("NAME ");

gotoxy(30, 6);

printf("ADDRESS ");

gotoxy(45, 6);

printf("PHONENUMBER ");

gotoxy(63, 6);

printf("NATIONALITY ");

gotoxy(83, 6);

printf("EMAIL ");

gotoxy(105, 6);

printf("PERIOD \n");

gotoxy(130, 6);

printf("RENTED TIME \n");

printf("\n");

for (i = 0; i < 155; i++)

{

    printf("-");

}

while(fscanf(fp,"%s %s %i %s\n", s.bikename, user_name, &s.period,time) !=
EOF)

{

    if(strcmp(user_name,username) == 0)

    {

        f = fopen("Register.txt", "r");

```

```

        while (fscanf(f, "%s %s %s %s %s %s %s %s\n",
s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.license) != EOF)

        {

            if(strcmp(user_name,s.username) == 0)

            {

                gotoxy(1, q);

                printf("%s", s.bikename);


                gotoxy(15, q);

                printf("%s", s.fullname);


                gotoxy(30, q);

                printf("%s", s.address);


                gotoxy(45, q);

                printf("%s", s.phonenumber);


                gotoxy(66, q);

                printf("%s", s.nationality);


                gotoxy(82, q);

                printf("%s", s.email);

```

```

        gotoxy(107, q);

        printf("%i", s.period);


        gotoxy(122, q);

        printf("%s",time);

        q++;

    }

}

fclose(f);

}

}

gotoxy(0,q);

for (i = 0; i < 155; i++)

    printf("-");

fclose(fp);

getch();

UserMenu(username);

}

```

**FUNCTION NAME** : void UserSearch(char [])

```
void UserSearch(char username[20])
```

```

{

    system("cls");

    FILE *f,*fp;

```

```

char bikename[50],user_name[30],time[50];

int flag = 1;

fp = fopen("Rentals.txt","r");

Title();

if (fp == NULL)

{

    printf("\n\n\t\t\t\t\tNO RECORD ADDED.");

    getch();

}

else

{

    printf("\n\n\t\t\t\t\tENTER BIKENAME YOU WANT TO SEARCH :

");

    gets(bikename);

    while(fscanf(fp,"%s %s %i

%s\n",s.bikename,user_name,&s.period,time) != EOF)

    {

        if (strcmp(bikename, s.bikename) == 0)

        {

            f = fopen("Register.txt", "r");

            while (fscanf(f, "%s %s %s %s %s %s %s %s\n",

s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.licens

e) != EOF)

            {

```







```

system("cls");

printf("\n\n\n\t\t\t\t\t");

for (i = 0; i < 80; i++)

    printf("-");

printf("\n\t\t\t\t\t");

printf(" ----->>>> |MAIN MENU| <<<<-----
----- ");

printf("\n\t\t\t\t\t");

for (i = 0; i < 80; i++)

    printf("-");

printf("\n");

printf("\t\t\t\t\t\t *Select and Enter Your Choice From Menu*");

printf("\n\n");


printf("\n\t\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t\t\t Enter 1 ->> View Rental Record");

printf("\n\t\t\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t\t\t Enter 2 ->> View Vehicle Fares");

printf("\n\t\t\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t\t\t Enter 3 ->> Edit Fares and stock");

printf("\n\t\t\t\t\t\t\t=====");

printf(" \n\t\t\t\t\t\t\t Enter 4 ->> Remove A User");

```



```

break;

case 2:

    ViewFares();

    break;

case 3:

    EditFares();

    break;

case 4:

    delete();

    break;

case 5:

    search();

    break;

case 6:

    edit();

    break;

case 7:

    WelcomePage();

    break;

default:

    system("cls");

    printf("\n\t\t\t\t\t Incorrect Input");

    printf("\n\t\t\t\t\t Press any key to continue");

```

```

                                getch();

                                adminMenu();

                                }

                                }

                                }

```

**FUNCTION NAME** : void listOfRentals()

```

void listOfRentals()

{

    FILE *f,*fp;

    int i, q = 9;

    char user_name[30],time[50];

    if ((fp = fopen("Rentals.txt", "r")) == NULL)

        exit(0);

    system("cls");

    Title();

    gotoxy(0,5);

    for (i = 0; i < 155; i++)

    {

        printf("-");

    }

    gotoxy(1, 6);

    printf("BIKENAME ");

    gotoxy(15, 6);

```

```

printf("NAME ");

gotoxy(30, 6);

printf("ADDRESS ");

gotoxy(45, 6);

printf("PHONENUMBER ");

gotoxy(63, 6);

printf("NATIONALITY ");

gotoxy(83, 6);

printf("EMAIL ");

gotoxy(105, 6);

printf("PERIOD \n");

gotoxy(130, 6);

printf("RENTED TIME \n");


for (i = 0; i < 155; i++)

{

    printf("-");

}

printf("\n");

while(fscanf(fp,"%s %s %i %s\n", s.bikename, user_name, &s.period,time) !=
EOF)

{

    f = fopen("Register.txt","r");

```

```

        while (fscanf(f, "%s %s %s %s %s %s %s %s\n",
s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.licens
e) != EOF)

        {

            if(strcmp(user_name,s.username) == 0)

            {

                gotoxy(1, q);

                printf("%s", s.bikename);


                gotoxy(15, q);

                printf("%s", s.fullname);


                gotoxy(30, q);

                printf("%s", s.address);


                gotoxy(45, q);

                printf("%s", s.phonenumber);


                gotoxy(66, q);

                printf("%s", s.nationality);


                gotoxy(82, q);

                printf("%s", s.email);

```



```

        gotoxy(107, q);

        printf("%i", s.period);


        gotoxy(122, q);

        printf("%s",time);

        q++;

    }

}

fclose(f);

}

printf("\n\n");

for (i = 0; i < 155; i++)

    printf("-");

fclose(fp);

getch();

}

```

**FUNCTION NAME** : void EditFares()

```
void EditFares()
```

```

{

    system("cls");

    Title();

    FILE *fp, *r;

    int flag = 0;

```

```
char vehicle[20], price[7], editVehicle[20], editPrice[7],stock[7],editStock[7];

fp = fopen("viewFares.txt", "r");

r = fopen("temp.txt", "w+");

printf("\n");

while (fscanf(fp, "%s %s %s\n", vehicle, price,stock) != EOF)

{

    printf("\n\t\t\t\t\t\t\t%s\t: %s\n", vehicle, price);

}

fclose(fp);

printf("\n\t\t\t\t\t");

for (int i = 0; i < 80; i++)

    printf("-");

fp = fopen("viewFares.txt", "r");

printf("\n\n\t\t\t\t\tENTER VEHICLE TO EDIT : ");

gets(editVehicle);

printf("\n\t\t\t\t\tENTER UPDATED PRICE : ");

gets(editPrice);

printf("\n\t\t\t\t\tENTER UPDATED STOCK : ");

gets(editStock);


while (fscanf(fp, "%s %s %s\n", vehicle, price,stock) != EOF)

{

    if (strcmp(vehicle, editVehicle) == 0)
```







```

        adminMenu();
    }

FUNCTION NAME : void search()

void search()
{
    system("cls");

    FILE *f,*fp;

    char username[50],name[50];

    char user_name[30],time[50];

    int flag = 1;

    fp = fopen("Rentals.txt","r");

    Title();

    if (f == NULL)
    {
        printf("\n\n\t\t\t\t\tNO RECORD ADDED.");

        getch();
    }

    else
    {
        printf("\n\n\t\t\t\t\tENTER USERNAME OF COSTUMER TO
SEARCH : ");

        gets(username);

        printf("\n\t\t\t\t\tENTER FULLNAME OF COSTUMER : ");

        gets(name);
    }
}

```



```

printf("\n\t\t\t\t\tPHONE NUMBER\t:
%s", s.phonenumber);

printf("\n\t\t\t\t\tNATIONALITY\t:
%s", s.nationality);

printf("\n\t\t\t\t\tEMAIL\t\t: %s",
s.email);

printf("\n\t\t\t\t\tLICENSE NO\t: %s",
s.license);

printf("\n\t\t\t\t\tPERIOD\t\t:
%i",s.period);

printf("\n\t\t\t\t\tRENTED TIME   :
%s",time);

    }

}

}

fclose(f);

}

if (flag == 1)

{

    printf("\n\t\t\t\t\tREQUESTED USERNAME COULD NOT
BE FOUND !");

}

else if(flag == 2)

{

    printf("\n\t\t\t\t\tTHIS PERSON DID NOT RENT ANY BIKE WITH THE
USERNAME YOU ENTERED !");

```









```

        fprintf(r, "%s %s %s %s %s %s %s %s\n",
s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.license);

    }

}

else

{

    fprintf(r, "%s %s %s %s %s %s %s %s\n",
s.username,s.password,s.fullname,s.address,s.phonenumber,s.nationality,s.email,s.license);

}

}

if (k == 0)

{

    printf("\n\nNO RECORD WITH THIS USERNAME !");

}

else if((k == 2) && (t != 1))

{

    printf("\n\nNO BIKES RENTED IN THIS ACCOUNT BY THE NAME YOU ENTERED !");

}

fclose(r);

fclose(f);

remove("Register.txt");

rename("temp.txt", "Register.txt");

```

```
        getch();  
    }  
    adminMenu();  
}
```

**FUNCTION NAME** : void gotoxy(int,int)

```
void gotoxy(int x, int y)
```

```
{  
    COORD CR;  
    CR.X = x;  
    CR.Y = y;  
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), CR)  
}
```

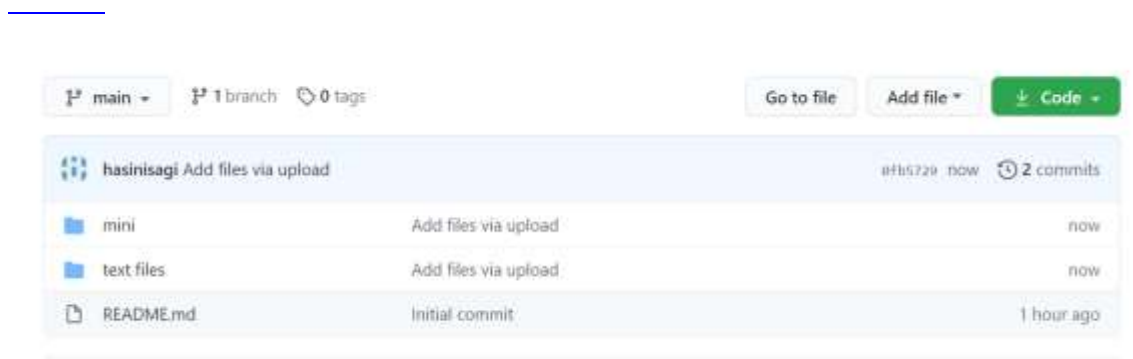
## B.Github structure/folder

We segregated the files of our mini project into folders. We also added report of our project on our github repository.

### Github Repository links

<https://github.com/shravanimahanth/Bike-rental-system>

<https://github.com/hasinisagi/bike-rental-system->



- **Code folder contents:**



- **Files folder contents**



## C.TESTING

TEST CASE TEMPLATE		
Test case ID : TC01		Use case ID : <b>UC01</b>
Test case title : Register		
Test case description : User has to enter details.		
Test steps	Expected result	Actual result
The system allows the user to enter the correct details to create an account with their user name and password.	If the user doesn't enter details in correct format then system won't allow to enter further details.	An error message saying invalid phone number or invalid name is displayed if they are in wrong format.

TEST CASE TEMPLATE		
Test case ID : TC01		Use case ID : <b>UC02</b>
Test case title : Login (User)		
Test case description : User has to enter password and user name.		
Test steps	Expected result	Actual result
The system prompts the user to enter password and user name correctly .	A message saying login is successful is displayed on the screen and redirects to users home page.	A message saying username not found is displayed.

Test case ID : TC02		Use case ID : <b>UC02</b>
Test case title : Login (user)		
Test case description : User has to enter password and user name.		
Test steps	Expected result	Actual result
The system prompts the user to enter password and user name correctly .	A message saying login is successful is displayed on the screen and redirects to users home page.	A message saying wrong password is displayed and allows to try again.

Test case ID : TC03		Use case ID : <b>UC02</b>
Test case title : Login		
Test case description : User has to enter password and user name.		
Test steps	Expected result	Actual result
The system prompts the user to enter password and user name correctly	A message saying login is successful is displayed on the screen and redirects to users home page.	A message saying login is successful is displayed on the screen and redirects to users home page.

Test case ID : TC01		Use case ID : <b>UC03</b>
Test case title : View Rental Fares		
Test case description : User has to click on view rental fares		
Test steps	Expected result	Actual result
After successful login the user is allowed to click on view rental fares to check price and stock available.	View Rental Fares page is displayed.	View Rental Fares page is displayed.



Test case ID : TC01		Use case ID : <b>UC04</b>
Test case title : Rent a Bike		
Test case description : User has to click on rent a bike		
Test steps	Expected result	Actual result
User is allowed to enter bike to be rented which is in the list of rental fares.	Message saying Desired Bike is rented is displayed if the bike chosen is in list of the available bikes and allows to enter time period	Message saying Desired Bike is rented is displayed if the bike chosen is in list of the available bikes and allows to enter time period

Test case ID : TC02		Use case ID : <b>UC04</b>
Test case title : Rent a Bike		
Test case description : User has to press any key to generate bill after renting		
Test steps	Expected result	Actual result
User is allowed to enter bike to be rented which is in the list of rental fares and press any key to generate bill	Bill page is displayed with respective time ,date, user details and the total amount	Bill page is displayed with respective time ,date, user details and the total amount

Test case ID : TC03		Use case ID : <b>UC04</b>
Test case title : rent a bike		
Test case description : User has to enter bike name		
Test steps	Expected result	Actual result
User is allowed to enter bike to be rented which is in the list of rental fares and press any key to generate bill.	Message saying Desired Bike is rented is displayed if the bike chosen is in list of the available bikes and allows to enter time period	If there is no availability of bikes then message saying no stock of this vehicle is displayed

Test case ID : TC01		Use case ID : <b>UC05</b>
Test case title : View Rental Records		
Test case description : User has to click on view rental records		
Test steps	Expected result	Actual result
User is allowed to view their rental records and verify.	Rental record page is displayed with the table displaying all the user details with respective date and time.	Rental record page is displayed with the table displaying all the user details with respective date and time.

<b>Test case ID</b> : TC01		Use case ID : <b>UC06</b>
<b>Test case title</b> : Search your rental record		
<b>Test case description</b> : User has to click on Search your rental record		
Test steps	Expected result	Actual result
The system prompts the user to check the records of specific bike by entering bike name.	Rental record page is displayed with all the rental records rented by the user on specific bike.	Rental record page is displayed with all the rental records rented by the user on specific bike.

<b>Test case ID</b> : TC02		Use case ID : <b>UC06</b>
<b>Test case title</b> : Search your rental record		
<b>Test case description</b> : User has to click on Search your rental record		
Test steps	Expected result	Actual result
The system prompts the user to check the rental records of specific bike by entering bike name if that user rented.	Rental record page is displayed with all the rental records rented by the user on specific bike.	Bike not rented by you message is displayed if the bike entered is not rented from your account.

Test case ID : TC01		Use case ID : <b>UC07</b>
Test case title : exit		
Test case description : Allows the user to exit from the user page .		
Test steps	Expected result	Actual result
Admin chooses the exit option.	It should redirect the user to the welcomepage.	It will redirect the user to the welcomepage

TEST CASE TEMPLATE		
Test case ID : TC01		Use case ID : <b>UC08</b>
Test case title : Login (Admin)		
Test case description : User has to enter password and user name.		
Test steps	Expected result	Actual result
The system prompts the Admin to enter password and user name .Admin enters the password and user name .Password entered is not same as the hard-coded one.	A message saying login is successful is displayed on the screen and redirects to Admin home page.	A message saying wrong password is displayed and allows to try again.

Test case ID : TC02		Use case ID : <b>UC08</b>
Test case title : Login (Admin)		
Test case description : User has to enter password and user name.		
Test steps	Expected result	Actual result
The system prompts the user to enter password and username .Admin enters the password and username. Password and user name entered is same as the hard-coded one.	A message saying login is successful is displayed on the screen and redirects to users home page.	A message saying login is successful is displayed on the screen and redirects to users home page.

Test case ID : TC01		Use case ID : <b>UC09</b>
Test case title : View rental records		
Test case description : User has to click on view rental records		
Test steps	Expected result	Actual result
Admin is allowed to view all the rental records.	Rental record page should be displayed with the table displaying all the user details with respective date and time.	Rental record page is displayed with the table displaying all the user details with respective date and time.

Test case ID : TC01		Use case ID : <b>UC010</b>
Test case title : view vehicles fares		
Test case description : User has to click on view rental fares		
Test steps	Expected result	Actual result
After successful login the user is allowed to click on view rental fares to check price and stock available to rent a bike.	View Rental Fares page is displayed.	View Rental Fares page is displayed.

Test case ID : TC01		Use case ID : <b>UC011</b>
Test case title : Edit Fares and Stock.		
Test case description : User has to click on edit fares and stocks.		
Test steps	Expected result	Actual result
The system prompts the admin to enter vehicle name .Admin enters the vehicle name .vehicle entered doesn't exist.	It should change the price and the stock of the vehicle name we entered.	It displays the message that the vehicle doesn't exist.

<b>Test case ID</b> : TC02		Use case ID : <b>UC011</b>
<b>Test case title</b> : Edit Fares and Stock.		
<b>Test case description</b> : User has to click on edit fares and stocks.		
Test steps	Expected result	Actual result
The system prompts the admin to enter vehicle name .Admin enters the vehicle name .vehicle entered exist and updates the price and stock.	It should change the price and stock of the vehicle name we entered.	It changes the price and the stock of the vehicle name we entered.

<b>Test case ID:</b> TC01		Use case ID: <b>UC012</b>
<b>Test case title:</b> Remove a user		
<b>Test case description:</b> User has to enter the username to delete the record.		
Test steps	Expected result	Actual result
The Admin is allowed to enter username. User enters the user name. Username entered doesn't exist in the records.	After entering the username all the record should be deleted on that username.	It displays a message that the records of this costumer are not found.

Test case ID: TC02		Use case ID : <b>UC012</b>
Test case title: Remove a user		
Test case description: User has to enter the username to delete the record.		
Test steps	Expected result	Actual result
Admin is allowed to enter username. User enters the user name .Username entered exist in the records.	After entering the username all the record should be deleted on that username.	After entering the username all the record are deleted on that username.

Test case ID : TC01		Use case ID : <b>UC013</b>
Test case title : Search a user		
Test case description : Admin chooses search a user option.		
Test steps	Expected result	Actual result
Admin is allowed to enter user name and full name of the customer. User name is not found.	The entered username and costumer records should be displayed .	The displays the messages that Required username could not be found.

Test case ID : TC02		Use case ID : <b>UC013</b>
Test case title : Search a user		
Test case description : Admin chooses search a user option.		
Test steps	Expected result	Actual result
Admin is allowed to enter user name and full name of the customer. User name is found.	The entered username and costumer records should be displayed .	The entered username and costumer records are being displayed .

Test case ID : TC01		Use case ID : <b>UC014</b>
Test case title : edit user		
Test case description : Admin chooses the edit user option.		
Test steps	Expected result	Actual result
Admin is allowed to enter user name and full name of the customer. User name is not found.	After entering the details it allows us to edit the details.	After entering the details is shows no records with this user name is displayed.

Test case ID : TC02		Use case ID : <b>UC014</b>
Test case title : edit user		
Test case description : Admin chooses the edit user option.		
Test steps	Expected result	Actual result
Admin is allowed to enter user name and full name of the customer. User name is found.	After entering the details it allows us to edit the details.	After entering the details it allowed us to edit the details.

Test case ID : TC01		Use case ID : <b>UC015</b>
Test case title : exit		
Test case description : Allows the user to exit from the user page .		
Test steps	Expected result	Actual result
Admin chooses the exit option.	It should redirect the user to the welcomepage.	It will redirect the user to the welcomepage

## RESULT





11000P

ENTER USERNAME : SARAH0001

ENTER PASSWORD : SARAH0001

ENTER FULL NAME : SAR, J

Full Name contains an invalid character...Enter again.

ENTER FULL NAME : SARAH0001

ENTER ADDRESS : 990

Invalid...The address should be in range of 5 to 30

ENTER ADDRESS : 00100

Contact no: 333

Invalid...Contact must contain 10 digits.

Contact no: 1234567890

Invalid...Contact must contain 10 digits.

Contact no: 1234567890

ENTER MATRIMONY : 300700

ENTER EMAIL : SARAH0001@CPS

ENTER OTHER PHONE NUMBER : 32340

ACCOUNT SUCCESSFULLY CREATED

```
***** LOGIN FORM *****  
  
ENTER USERNAME : admin  
ENTER PASSWORD : *****  
  
PASSWORD NOT FOUND!
```

```
***** LOGIN FORM *****  
  
ENTER USERNAME : admin  
ENTER PASSWORD : *****  
  
-----WRONG PASSWORD!!--  
----- TRY AGAIN -----
```



***** BIKE RENTAL SYSTEM *****		
*****		
***** PRICES OF VEHICLES PER DAY AND STOCK *****		
VEHICLE	PRICE	STOCK LEFT
ACTIVA	500	5
PAALISH	500	5
JETTER	500	2
DDI	500	2
PLEASURE	500	5
ACTIVA	500	5
PRISMA	500	5
HS200	500	5
HS-DARK	1200	5
HS	1200	5
*****		
PRESS ANY KEY TO CONTINUE		

```

*****
***** ENTER DETAILS *****
*****
ENTER RIDE NAME : HS2000
PLEASE ENTER A VALID RIDE NAME!
ENTER RIDE NAME : JETTER
ENTER TIME PERIOD(w/ days) : 3
*****
***** YOUR SELECED BIKE IS RENTED SUCCESSFULLY *****
*****
PRESS ANY KEY TO GO TO BILLING :

```



```

=====
ENTER DETAILS
=====

ENTER RENT AMT. : 100000
=====
ENTER TIME PERIOD('x' days) : 10
=====
              (TRUNCATE)
THE MAXIMUM PRICE TO RENT IS 10 DAYS,
=====
ENTER TIME PERIOD('x' days) : 9
=====

===== YOUR DESIRED RENT IS RENTED SUCCESSFULLY =====

PRESS ANY KEY TO GO TO STEERING :

```

ENTER DETAILS

ENTER RIDE NAME : JUPITER

NO STOCK ON THIS VEHICLE :

ENTER RIDE NAME : ACTIVA

NO STOCK ON THIS VEHICLE :

ENTER RIDE NAME : B00

ENTER TIME (HH:MM("a"day)) : 0

YOUR DESIRED RIDE IS RENTED SUCCESSFULLY

PLEASE APP KEY TO GO TO BILLING

STOCKNAME	NAME	ADDRESS	PHONE NUMBER	NATIONALITY	EMAIL	RENTED	RENTED TIME
JUPITER	SARASWATI	BEHAL	1111111111	INDIAN	HIGRAVAIL.COM	1	Fri-Dec-10-14:10:39-2020
JUPITER	SARASWATI	BEHAL	1234567890	INDIAN	HIGRAVAIL.COM	1	Fri-Dec-10-14:11:30-2020
B00	SARASWATI	BEHAL	1234567890	INDIAN	HIGRAVAIL.COM	0	Fri-Dec-10-14:12:38-2020

\*\*\*\*\*  
\*\*\*\*\* BITE RENTAL SYSTEM \*\*\*\*\*  
\*\*\*\*\*

ENTER KEYSAME YOU WANT TO SEARCH : JUPITER

\*\*\*\*\*  
RECORD FOUND

BIKE NAME	:	JUPITER
NAME	:	SHAWANE
ADDRESS	:	DELHI
PHONE NUMBER	:	1214557890
NATIONALITY	:	INDIAN
EMAIL	:	SHAWANE@GMAIL.COM
LICENSE NO	:	12145
PHONE	:	9
RENTED TIME	:	2014-06-18-14:10:10-2014

\*\*\*\*\*  
RECORD FOUND

BIKE NAME	:	JUPITER
NAME	:	SHAWANE
ADDRESS	:	DELHI
PHONE NUMBER	:	1214557890
NATIONALITY	:	INDIAN
EMAIL	:	SHAWANE@GMAIL.COM
LICENSE NO	:	12145
PHONE	:	9
RENTED TIME	:	2014-06-18-14:11:10-2014

\*\*\*\*\*  
\*\*\*\*\* BITE RENTAL SYSTEM \*\*\*\*\*  
\*\*\*\*\*

ENTER KEYSAME YOU WANT TO SEARCH : JUNE

THIS BIKES IS NOT REATED BY YOU !



```
.....>>>> BIKE RENTAL SYSTEM <<<<.....
.....>>>> BIKE RENTAL SYSTEM <<<<.....
```

```
.....
```

1. REGISTER/ CREATE AN ACCOUNT
2. LOGIN TO EXISTING ACCOUNT
3. LOGIN AS ADMIN
4. EXIT

```
.....
```

```
ENTER YOUR CHOICE : 4
```

```
.....>>>> BIKE RENTAL SYSTEM <<<<.....
.....>>>> BIKE RENTAL SYSTEM <<<<.....
```

```
.....
```

1. REGISTER/ CREATE AN ACCOUNT
2. LOGIN TO EXISTING ACCOUNT
3. LOGIN AS ADMIN
4. EXIT

```
.....
```

```
ENTER YOUR CHOICE : 3
```

```

-----
*****BIKE RENTAL SYSTEM*****
-----

***** LOGIN FOR ADMIN *****

ENTER USERNAME:- bikerentalsystem

ENTER PASSWORD:- abcd

-----WRONG PASSWORD!!!!-----

***** TRY AGAIN *****

```

```

.....
----->>>> BIKE RENTAL SYSTEM <<<<-----
.....

***** LOGIN FOR ADMIN *****

ENTER USERNAME:- efgh

ENTER PASSWORD:- e!ndproject

                                USERNAME NOT FOUND! )_

```

```

.....
-----BIKE RENTAL SYSTEM-----
.....

***** LOGIN FOR ADMIN *****

ENTER USERNAME:- bikarentalsystem

ENTER PASSWORD:- adminproject

----- LOGIN SUCCESSFUL -----

```

[illegible]

[illegible]

```

----->>>> BIKI RENTAL SYSTEM <<<<-----
-----
BIKENAME      NAME      ADDRESS      PHONENUMBER      NATIONALITY      EMAIL      PERIOD      RENTED TIME
-----
JUPITER      SHRAVANI      DELHI      1234567890      INDIAN      MS@GMAIL.COM      3      Fri-Dec-18-14:10:39-2020
JUPITER      SHRAVANI      DELHI      1234567890      INDIAN      MS@GMAIL.COM      9      Fri-Dec-18-14:15:16-2020
BII          SHRAVANI      DELHI      1234567890      INDIAN      MS@GMAIL.COM      8      Fri-Dec-18-14:12:28-2020
-----

```

2

ENTER ANY KEY TO CONTINUE

[illegible]

```

.....>>>> BIKE RENTAL SYSTEM <<<<.....
*****

ACTIVA : 600
MAESTRO : 700
JUPITER : 500
DIO : 500
PLEASURE : 600
ACCESS : 700
PULSAR : 800
NS200 : 500
KTM-DUKE : 1500
R15 : 1200

.....

ENTER VEHICLE TO EDIT | HERO
ENTER UPDATED PRICE : 500
ENTER UPDATED STOCK : 3

THE VEHICLE DOESN'T EXIST!!!

```

[illegible]

```

.....
----->>>> BIKE RENTAL SYSTEM <<<<-----
.....

ACTIVA : 600
HAESTRO : 700
JUPITER : 500
DIO : 500
PLEASURE : 500
ACCESS : 700
PULSAR : 600
NS200 : 500
KTM-DUKE : 1500
RLS : 1200

-----

ENTER VEHICLE TO EDIT : PLEASURE
ENTER UPDATED PRICE : 600
ENTER UPDATED STOCK : 7

```

=====2

ENTER ANY KEY TO CONTINUE.





```

=====BIKE RENTAL SYSTEM=====
----->>>>BIKE RENTAL SYSTEM<<<<-----

ENTER USERNAME OF COSTUMER TO SEARCH : SHRAVANI

ENTER FULLNAME OF COSTUMER : SHRAVANI

=====

RECORD FOUND

BIKE NAME      : JUPITER
NAME           : SHRAVANI
ADDRESS        : DELHI
PHONE NUMBER   : 1234567890
NATIONALITY    : INDIAN
EMAIL          : PS@GMAIL.COM
LICENSE NO     : 12345
PERIOD         : 3
RENTED TIME    : Fri-Dec-18-14:18:29-2028

=====

RECORD FOUND

BIKE NAME      : JUPITER
NAME           : SHRAVANI
ADDRESS        : DELHI
PHONE NUMBER   : 1234567890
NATIONALITY    : INDIAN
EMAIL          : PS@GMAIL.COM
LICENSE NO     : 12345
PERIOD         : 9
RENTED TIME    : Fri-Dec-18-14:11:34-2028

=====

RECORD FOUND

BIKE NAME      : GIGI
NAME           : SHRAVANI
ADDRESS        : DELHI
PHONE NUMBER   : 1234567890
NATIONALITY    : INDIAN
EMAIL          : PS@GMAIL.COM
LICENSE NO     : 12345
PERIOD         : 6
RENTED TIME    : Fri-Dec-18-14:12:28-2028

```







```
.....
----->>>> BIKI RENTAL SYSTEM <<<<-----
.....

ENTER USERNAME OF COSTUMER TO DELETE : HASINI

RECORDS OF THIS COSTUMER ARE NOT FOUND !_
```

[illegible]

[illegible]



WELCOME TO BIKE RENTAL SYSTEM

Current date and time : Fri Dec 18 23:29:41 2020

Press any other key to continue: \_

**What is the additional knowledge gained as a result of implementing this mini project apart from the syllabus covered in the course programming for problem solving?**

Apart from the syllabus covered in the course programming for problem solving we have learnt how to change the background colour of the console also we have learnt how to set the cursor position at desired location in c program. We have also learnt the functionality of <time.h> header file which contains time and date function declarations to provide standardized access to time/date.

This project really helped us to know the various real world application of c programming and know the essence of this language.

## **Conclusion and Future work**

While making of the project "BIKE RENTAL SYSTEM" we made our progress by solving a number of problems. Solution to each problem by ourselves was the most important part of the project and this provided us with experiences which will help us in future.

We came to know that how we can use concept of files in c programming language in storing data ,reading and writing in file as well . Also we learnt the use of structures , arrays in solving real life problems

Some important things that we learned include designing a good program architecture and converting real life situations into an efficient code, and how to write an good looking easily readable and understandable as well as time and memory efficient code.

Also, in future we are thinking to build a feature called Rent for friend which allows users friend to rent a bike through users account and also we are thinking to set up this console based project as an application based project

## References

The References we used in this project are

1. [www.google.com](http://www.google.com)
2. [www.geeksforgeeks.org](http://www.geeksforgeeks.org) c
3. Kernighan and Ritchie "The C Programming Language", 2nd ed, 1988, (The "K&R". The Bible for the C language)