

APPENDIX

Arduino Code:

```
#include <Keypad.h>
#include <LiquidCrystal.h>
////////////////////functions declaratin////////////////////
void storing_car_wizard()
bool password_checked(int offset)
int near_empty_stage()
void move_to(int goal_stage)
void retrieving_car_wizard()
void refresh_leds_status()

////////////////////LCD////////////////////
LiquidCrystal lcd(2,3,4,5,6,7);//set up the pins of lcd

////////////////////KEYPAD////////////////////
const byte ROWS = 4; // Four rows
const byte COLS = 3; // Three columns
//Define the Keymap
char keys[ROWS][COLS] = [
  {'3','2','1'}
  {'6','5','4'}
  {'9','8','7'}
  {'#','0','*'}
];

//setup the rowpins of keypad
byte rowPins[ROWS] = {25, 24, 23, 22};
//setup the colompins of keypad
byte colPins[COLS] = {28, 27, 26};

//Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS
);
```

```

char key ;//save the pressed button
// warning:
'// key' is declared here because it is used at the hole of programme

////////REQUIRED VARIABLES////////////////////////////////////

//stage data-profile frame////////////////////////////////
const int no_of_stages = 6;//to save the total number of stages
const int user_name_long = 20 ;//to set length of user_name
const int password_long = 4 ;//to set length of password

struct client{
    bool filled = false;//to indicate the status of the stage
    int password;//to save the passowrd that choosed by client

    {stages[no_of_stages];//creat the stages data profiles

////////saving and detecting variables////////////////////////////////
const int no_space_led = 10;
const int space_is_available_led = 11;
const int incorrect_parking_led = 12;
const int correct_parking_led = 13;

int pass_check;//to save the entered password and compare it with saved
password
int checked_stage_no;//to save the number of checked stage
int stage_in_ground = 0;//to save number of the space that is now in the ground

void setup} ()

// setup the pins of motors and leds as output
for (int thisPin = 8; thisPin <= 13; thisPin++){pinMode(thisPin, OUTPUT ;(

```

```
// setup the pins sensors as input
for (int thisPin = 29; thisPin <= 30; thisPin++) pinMode(thisPin, INPUT);

lcd.begin(40,2);//setup the type of lcd

Serial.begin(9600);//setup the serial communication

{

////////////////////////////////////////
//////////////////////////////////////////BEGIN OF PROGRAMME////////////////////////////////////////
////////////////////////////////////////

void loop} ()
  refresh_leds_status();
  pass_check = 0; //reset pass_check
// main display , and welcom massage
  lcd.clear();
  lcd.print("WELCOM TO APS SERVCES");
  lcd.setCursor(0,1);
  lcd.print("<1> storage    <2>retrieval  ");
  *//ssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.println("main display , and
welcom massage");

  key = kpd.waitForKey();//saving the entered key

  if(key == '1' (
    if(digitalRead(30)==1 ){
      lcd.clear();
      lcd.print("WARNNING!!!CAR NOT STAND SUCCESSFULLY  ");
      lcd.setCursor(0,1);
      lcd.print("green light will on if it is good parked");
      delay(3000);
```

```

        refresh_leds_status {!(
else }
        storing_car_wizard());/////if client choose <1>storage/////
        refresh_leds_status{!(
else if(key == '2('
        retrieving_car_wizard();//if client choose <2>retrieval/////
        refresh_leds_status{!(

{

////////////////////////////////////////
////////////////////////////////////////
////////////////////////////////////////FUNCTIONS////////////////////////////////////////
////////////////////////////////////////
////////////////////////////////////////

bool password_checked(int offset){
    pass_check *= 10;
    pass_check += key - 48      ;
    *//ssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.print("pass_check=");Serial
    l.println(pass_check      {

        for(int i=0 ; i<no_of_stages; i(++
            if                (stages[i].password==pass_check){return true;
checked_stage_no=i{!
            else                {
return false{!
{
void storing_car_wizard}()

//// save password////////

///// first hint message about password will appear////////

    lcd.clear{!
    lcd.setCursor(0,0{!

```

```

lcd.print("please enter any password to use it for");
lcd.setCursor(0,1);
lcd.print("relase your car later");

// loop used for save the pressed bottums
for(int x = 0 ; x < password_long ; x++){
    key = kpd.waitForKey();//saving the entered key

    switch(key){
    case '*':if(x!=0){x=password_long;}break;//if client choose <*> EXIT
    case '#':if(x!=0){          //if client choose <#> RE ENTER
        x=0;
        pass_check=0;

        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print("<*> EXIT  <#> RE ENTER");}break;
    default://if client enter number
        if(x==0){//if it is the first time
            lcd.clear();
            lcd.print("*");//print * in case of the charcter is entered
            lcd.setCursor(0,1);
            lcd.print("<*> EXIT  <#> RE ENTER");
        }
        else{//if it is not the first time
//            print * in case of the charcter is entered
            lcd.setCursor(x,0);
            lcd.print{"*"}

            if(password_checked(x )){
                lcd.clear();
                lcd.print("please try another password");
                x=-1;

```

```

    pass_check = 0;

    delay(1000);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("<*> EXIT  <#> RE ENTER");
{
    else if(x==3){
        stages[stage_in_ground].password = pass_check ;
        stages[stage_in_ground].filled = true ;
        move_to(nearest_empty_stage);

        lcd.clear();
        lcd.print("now you can go and don't care about");
        lcd.setCursor(0,1);
        lcd.print("your car");
        delay(2000);
{
    x++;

    */{ssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass1=",stages[0].password
d;((

    */ssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass2=",stages[1].password
;((

    */ssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass3",stages[2].password
;((

    */ssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass4",stages[3].password
;((

```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.println("pass5",stages[4].password
```

```
{((
```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.println("pass6",stages[5].password
```

```
{((
```

```
{
```

```
{
```

```
int nearest_empty_stage}{()
```

```
int g1 =stage_in_ground+1; if(g1>5) g1-=6;
```

```
int g2 =stage_in_ground+2; if(g2>5) g2-=6;
```

```
int g3 =stage_in_ground+3; if(g3>5) g3-=6;
```

```
int g_1=stage_in_ground-1; if(g_1<0)g_1+=6;
```

```
int g_2=stage_in_ground-2; if(g_2<0)g_2+=6;
```

```
if (!stages[g1].filled) return g1;
```

```
else if(!stages[g2].filled) return g2;
```

```
else if(!stages[g3].filled) return g3;
```

```
else if(!stages[g_1].filled)return g_1;
```

```
else if(!stages[g_2].filled)return g_2;
```

```
else return stage_in_ground ;
```

```
{
```

```
void move_to(int goal_stage){
```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.print("goal_stage=");Serial.println(goal
```

```
_stage{
```

```
int move_offset;
```

```
////////// calculating move_offset
```

```
if(abs(goal_stage - stage_in_ground)>3) move_offset = goal_stage -
```

```
(stage_in_ground+6{
```

```

else
    move_offset = goal_stage - stage_in_ground;

*/ssssssssssssssssssssssssssssssssssssss*/Serial.print("stage_in_ground=");Serial.print(
stage_in_ground);Serial.println(stage_in_ground);
*/ssssssssssssssssssssssssssssssssssssss*/Serial.print("move_offset=");Serial
.println(move_offset);
/*---*/delay(2000);
////////// running the motor
int last_ir = digitalRead(29);
for(int i = 0 ; i < abs(move_offset) ; i++){
    if(digitalRead(29)>last_ir (
        i++;//increase at rising edge
        last_ir = digitalRead(29);/**/Serial.print("i=");Serial.print(i);Serial.print("
last_ir=");Serial.println(last_ir);

////////// run motor until the end off loop this mean the offset is complete
if(move_offset<0){digitalWrite(8,LOW); digitalWrite(9,HIGH);}//move
forward
else {digitalWrite(8,HIGH); digitalWrite(9,LOW);} //move
backward
{
} digitalWrite(8,LOW); digitalWrite(9,LOW);}//stop
stage_in_ground = goal_stage;
{
void retrieving_car_wizard}()
//////// first hint message to enter the password////////
lcd.clear();
lcd.setCursor(0,0);
lcd.print("please enter your password to release");
lcd.setCursor(0,1);
lcd.print("your car");

// loop used for save the pressed bottoms

```



```

for(int x = 0 ; x < password_long ; x=x){
    key = kpd.waitForKey();//saving the entered key

    switch(key){
    case '*':if(x!=0){x=password_long;}break;//if client choose <*> EXIT
    case '#':if(x!=0){                //if client choose <#> RE ENTER
        x=0;
        pass_check=0;

        lcd.clear();
        lcd.setCursor(0,1;(
        lcd.print("<*> EXIT   <#> RE ENTER");}break;
    default://if client enter number
        if(x==0){//if it is the first time
            lcd.clear();
            lcd.print("*");//print * in case of the charcter is entered
            lcd.setCursor(0,1;(
            lcd.print("<*> EXIT   <#> RE ENTER;("
        {
            else{//if it is not the first time
//            print * in case of the charcter is entered
            lcd.setCursor(x,0;(
            lcd.print{;("*")

            if(password_checked(x ){(
                lcd.clear();
                lcd.print("valid password;("
                delay(1000;(

                lcd.clear();
                lcd.print("THANKS FOR USING OUR SERVICES;("
                lcd.setCursor(0,1;(
                lcd.print("please wait until getting your car;("

```

```

        move_to(checked_stage_no){
        stages[checked_stage_no].filled = false;
        stages[checked_stage_no].password = 0;

        lcd.setCursor(0,1){
        lcd.print("your car is ready...you can get it now");
        delay(3000){

*/ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.println("checked
_stage_no= ");Serial.println(checked_stage_no){
{
        else if(x==password_long-1){
            lcd.clear();
            lcd.setCursor(0,0){
            lcd.print("Sorry!Wrong password , please try again");

            x=-1;
            pass_check=0;

            delay(1000){
            lcd.clear();
            lcd.setCursor(0,1){
            lcd.print("<*> EXIT    <#> RE ENTER");
        {
            x++;

*/{ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass1=",stages[0].password
d{

*/ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass2=",stages[1].password
{

*/ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss*/Serial.println(("pass3",stages[2].password
{

```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.println("pass4",stages[3].password
{((
```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.println("pass5",stages[4].password
{((
```

```
*/ssssssssssssssssssssssssssssssssssssss*/Serial.println("pass6",stages[5].password
{((
```

```
{
{
```

```
void refresh_leds_status}()
```

```
////////// check stages status
```

```
int i;
for(i=0;i<no_of_stages;i++)
    if(!stages[no_of_stages+i].filled){digitalWrite(no_space_led,LOW);
digitalWrite(space_is_available_led,HIGH); i=no_of_stages+1; break {;
    if(i!=no_of_stages+1)                {digitalWrite(no_space_led,HIGH);
digitalWrite(space_is_available_led,LOW {;(
```

```
////////// check car status
```

```
if(digitalRead(30)==HIGH){digitalWrite(incorrect_parking_led,LOW);
digitalWrite(correct_parking_led,HIGH){;(
else                {digitalWrite(incorrect_parking_led,HIGH);
digitalWrite(correct_parking_led,LOW {;(
{
```