

بسم الله الرحمن الرحيم

گزارشکار پروژه درس طراحی سیستم های ریزپردازنده

عنوان پروژه : طراحی وب سرور برای برد esp32 با html

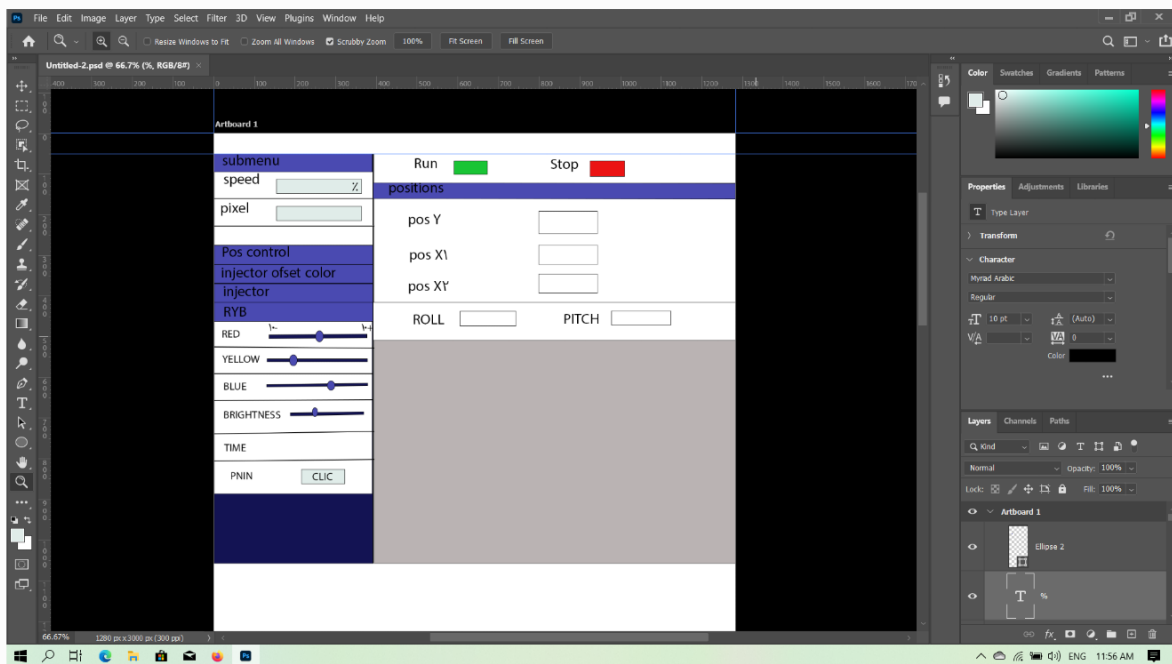
استاد : آقای دکتر محسن داودی

دانشجو : مبینا جلیوندشیرخانی تبار

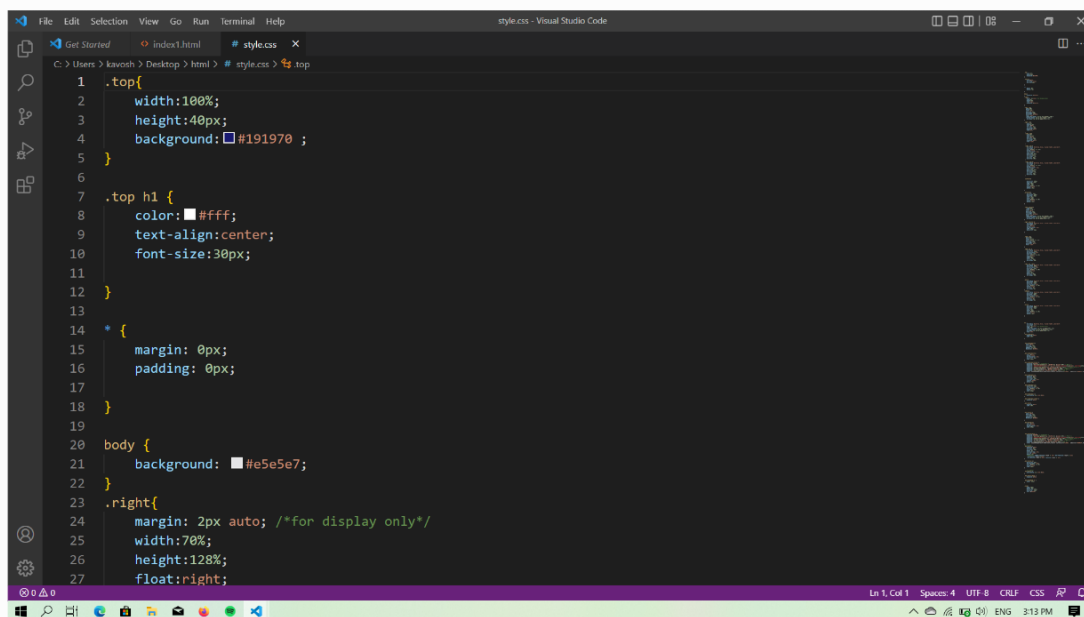
۹۷۶۱۲۳۰۳۲

اقدامات انجام شده :

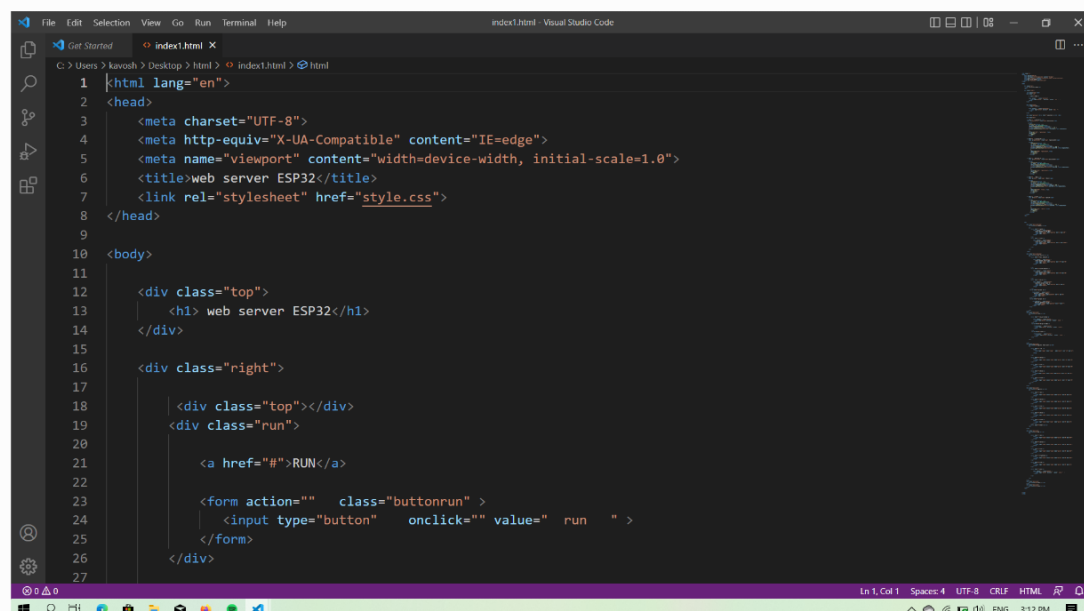
بعد از دریافت فرم کلی مورد نظر پروژه از آقای مهندس حسینی ،  
نصبت به طراحی قالب وب سرور در فتوشاپ و تایید نهایی آن اقدام شد  
قالب طراحی شده در زیر نشان داده شده است .



با تایید قالب طراحی شده شروع به اجرا و نوشتن کدهای مورد نیاز  
کردیم . باتوجه پیچیده بودن وب سرور مورد نظر باید از زبان های CSS و  
Java script هم استفاده می شد . قسمتی از کدهای نوشته شده در  
گزارش کار قرار داده شده است .



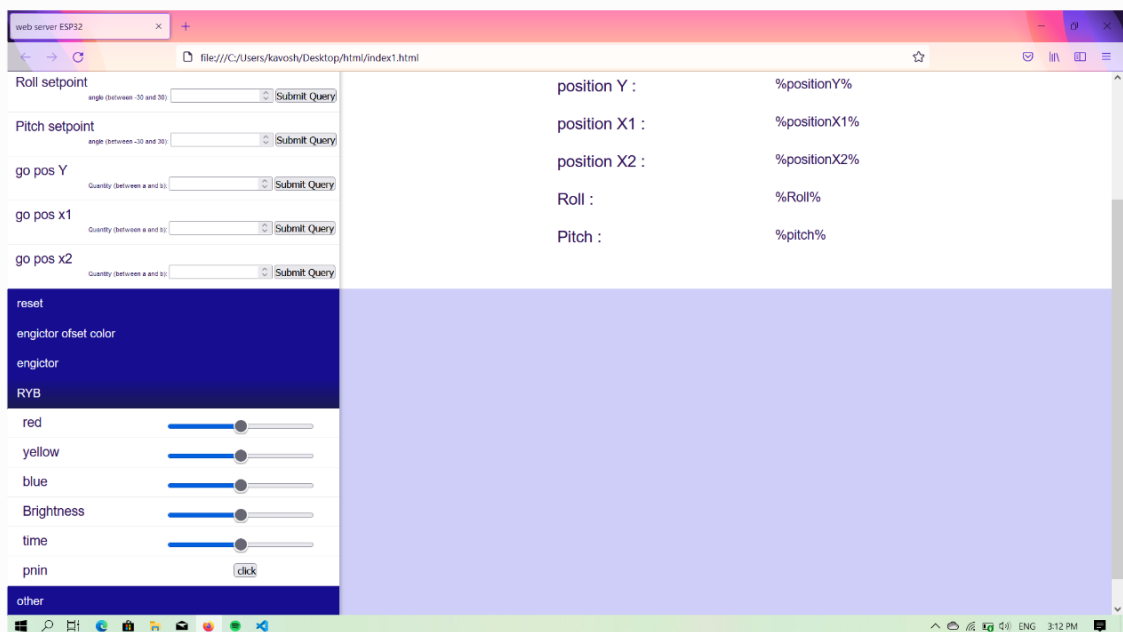
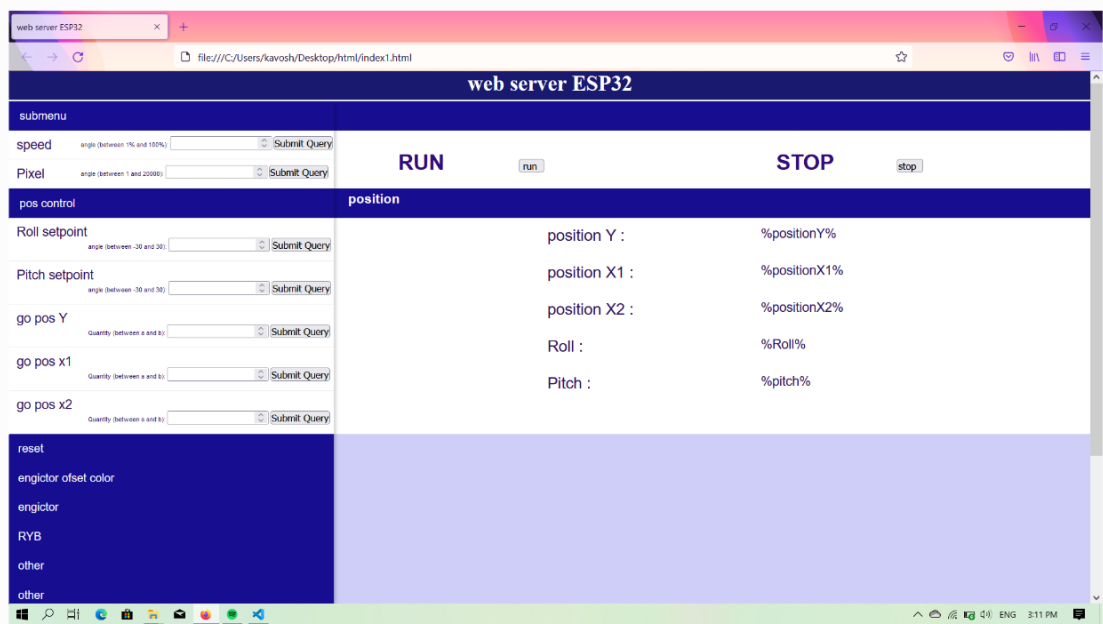
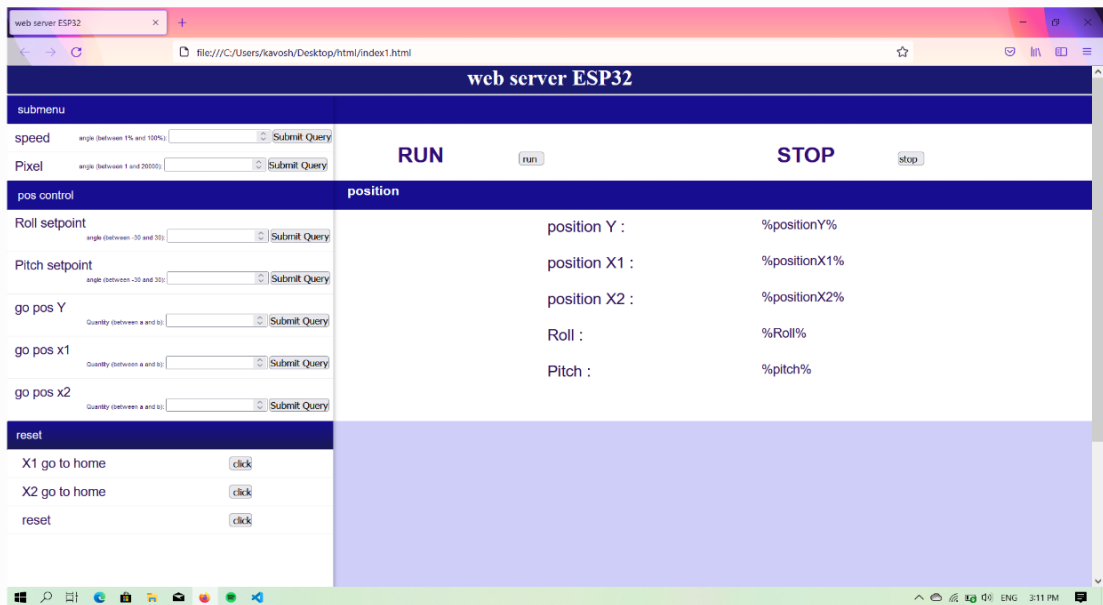
```
1 .top{
2   width:100%;
3   height:40px;
4   background: #191970 ;
5 }
6
7 .top h1 {
8   color: #fff;
9   text-align:center;
10  font-size:30px;
11 }
12
13
14 * {
15   margin: 0px;
16   padding: 0px;
17 }
18
19
20 body {
21   background: #e5e5e7;
22 }
23
24 .right{
25   margin: 2px auto; /*for display only*/
26   width:70%;
27   height:128%;
28   float:right;
```



```
1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
4   <meta http-equiv="X-UA-Compatible" content="IE=edge">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>web server ESP32</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9
10 <body>
11
12   <div class="top">
13     <h1> web server ESP32</h1>
14   </div>
15
16   <div class="right">
17
18     <div class="top"></div>
19     <div class="run">
20
21       <a href="#">RUN</a>
22
23       <form action="" class="buttonrun" >
24         <input type="button" onclick="" value=" run " >
25       </form>
26     </div>
27
```

وب سرور طراحی شده پس از تکمیل شدن بصورت زیر در آمده است. در این پروژه یک منو عمودی کشویی داریم که از چند زیر منو تشکیل شده اند این زیر منو ها مقادیر و پارامتر های مورد نظر را کنترل و تنظیم می کنند که با استفاده از یک زبان backend با سنسور ها و ربات در ارتباط اند . در هر کدام از این زیر منو ها چندین متغیر و پارامتر به عنوان ورودی به سیستم داده می شود و مورد تنظیم قرار می گیرند.

یک باکس برای run و stop کردن فعالیت ربات در نظر گرفته شده است و در باکس position موقعیت ربات را بصورت آنلاین نشان می دهد که اطلاعات خود را از سنسور ها گرفته و ارتباط بین سنسور ها و وب سرور از طریق زبان backend برقرار می شود. در این وب سرور یک باکس هم برای نمایش آنلاین تصاویر از موقعیت ربات تعبیه شده است .



برای راه اندازی وبسرور روی برد esp32 از روش زیر استفاده می کنیم  
برای این کار بهتر است که فایل های html و css و ... را ابتدا در یک SD  
قرار داده تا ESP32 فایل های ذخیره شده را برای Client ارسال کند .

برای اتصال SD کارت به ESP32 بهتر است از یک ماژول SD کارت  
استفاده کنیم. در این ماژول تمام حداقل های مورد نیاز برای راه اندازی  
SD وجود دارد در نتیجه نیازی به قطعات دیگه ای نخواهیم داشت و  
میتوانید ماژول را مستقیم به ESP32 متصل کنیم .

برای انجام پروژه مراحل زیر را انجام می دهیم:

- ۱\_ ماژول SD را به برد esp32 وصل میکنیم
  - ۲\_ کارت SD را به کامپیوتر وصل می کنیم ( حتما کارت باید فرمت شود  
(از FAT32 استفاده می کنیم )) .
  - ۳\_ سپس نوبت به کپی کردن فایل های HTML و CSS بر روی کارت SD  
است . برای این کار یک فایل txt در داخل درایو SD کارت میسازیم و  
کدهای HTML زیر را در آن کپی می کنیم و در آخر اسم فایل را  
به index.html تغییر میدهیم  
همین کار را برای فایل CSS نیز باید انجام میدهیم . برای این کار یک  
فایل txt بسازید و پس از کپی کردن کدهای زیر در آن ، اسم آن را به  
style.css تغییر میدهیم . هر دو فایل style.css و index.html باید  
در درون SD کارت موجود باشند .
- حال SD را در درون ماژول SD قرار داده و برد را به کامپیوتر متصل  
میکنیم و کدهای فایل آردوینو را بر روی برد ESP32 پروگرام می کنیم  
تمام فایل های کدنویسی شده در فایل پیوست قرار دارند .

```
sd_esp32 | Arduino 1.8.16
File Edit Sketch Tools Help

sd_esp32
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include "FS.h"
#include "SD.h"
#include "SPI.h"
// Replace with your network credentials
const char* ssid = "در این قسمت نام وایفای خود را وارد کنید";
const char* password = "در این قسمت رمز وایفای خود را وارد کنید";
// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
void initSDCard() {
  if(!SD.begin()) {
    Serial.println("Card Mount Failed");
    return;
  }
  uint8_t cardType = SD.cardType();
  if(cardType == CARD_NONE) {
    Serial.println("No SD card attached");
    return;
  }
  Serial.print("SD Card Type: ");
  if(cardType == CARD_MMC) {
    Serial.println("MMC");
  } else if(cardType == CARD_SD) {
    Serial.println("SDSC");
  } else if(cardType == CARD_SDHC) {
    Serial.println("SDHC");
  } else {
    Serial.println("UNKNOWN");
  }
  uint64_t cardSize = SD.cardSize() / (1024 * 1024);
  Serial.printf("SD Card Size: %lluMB", cardSize);
}
void initWiFi() {
  WiFi.begin(ssid, password);
  while(WiFi.status() != WL_CONNECTED) {
    delay(5000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}
```

قسمتی از کد های آردینو فقط برای نمایش وب سرور ، این کد ها را با توجه به نام و پسورد wifi مورد استفاده ویرایش میکنیم و روی برد esp32 پرگرم میکنیم .

برای استفاده از برد esp32cam ارتباط سنسور ها با وب سرور کد آردینو رو بصورت زیر تغییر میدهم در این کد فقط پین های یک برد esp32cam وارد شده اند این پین ها را با توجه به برد مورد استفاده تغییر داده و پین های سنسور ها هم با توجه به دیتابیس به که اضافه باید کرد . قسمت از کد ها را در عکس زیر مشاهده می کنید.

```
esp32cam | Arduino 1.8.16
File Edit Sketch Tools Help

esp32cam
#include "esp_camera.h"
#include "FS.h" // SD Card ESP32
#include "SD_MMC.h" // SD Card ESP32
#include "soc/soc.h" // Disable brownout problems
#include "soc/rtc_cntl_reg.h" // Disable brownout problems
#include "driver/rtc_io.h"

// Pin definition for CAMERA_MODEL_AI_THINKER
// change pin definition if you're using another ESP32 with camera module
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y5_GPIO_NUM 35
#define Y6_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y4_GPIO_NUM 36
#define Y3_GPIO_NUM 21
#define Y2_GPIO_NUM 19
#define Y1_GPIO_NUM 18
#define Y0_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

// Keep track of number of pictures
unsigned int pictureNumber = 0;

//Stores the camera configuration parameters
camera_config_t config;

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector
  Serial.begin(115200);
}
```

