

عنوان پروژه : سیستم پایش نشانگرهای حیاتی ساده

توضیح پروژه: برنامه‌ای که داده‌های شبیه‌سازی شده نشانگرهای حیاتی (مثل ضربان قلب، فشار خون، اکسیژن خون) را دریافت کرده و آنالیز اولیه انجام می‌دهد.

وظیفه ۱: ساختار اصلی و منو

- منو:

```
1. Enter Patient Data
2. Analyze Current Data
3. Check for Alerts
4. Display Patient History
5. Exit
```

وظیفه ۲: دریافت داده‌های بیمار (شبیه‌سازی سنسور)

- از کاربر (به عنوان اپراتور) بخواهید مقادیر را وارد کند.
- داده‌ها را در یک struct ذخیره کنید یا در آرایه‌های موازی.

```
struct Patient {
    int heartRate;
    int systolicBP; // فشار سیستولی
    int diastolicBP; // فشار دیاستولی
    int bloodOxygen;
};
```

وظیفه ۳: آنالیز داده‌ها با توابع

- تابع `checkHeartRate`: ضربان قلب را بگیرد و اگر خارج از محدوده ۶۰-۱۰۰ بود، هشدار دهد.
- تابع `checkBloodPressure`: فشار خون را بگیرد و وضعیت (کم/نرمال/بالا) را برگرداند.
- تابع `checkOxygen`: درصد اکسیژن خون را بگیرد و وضعیت (نرمال و غیر نرمال) را گزارش می‌دهد.
- تابع `checkTemperature`: دمای بدن را بگیرد و وضعیت را گزارش دهد. (تب دارد یا ندارد)

وظیفه ۴: سیستم هشدار (Alert System)

- در گزینه ۳ منو، برنامه تمام پارامترها را چک می‌کند و اگر حتی یکی خارج از محدوده نرمال بود، پیغام هشدار قرمز چاپ می‌کند.

مشخصات فنی پروژه:

۱. ساختار داده‌ها (Data Structures)

```
struct Patient {
    int id;
    char name[50];
    int age;
    int heartRate;    // ضربان قلب (BPM)
    int systolicBP;   // فشار خون سیستولی (mmHg)
    int diastolicBP;  // فشار خون دیاستولی (mmHg)
    int bloodOxygen;  // اکسیژن خون (SpO2 %)
    float bodyTemp;   // دمای بدن (°C)
};
```

۲. محدوده‌های نرمال (Normal Ranges)

```
#define MIN_HR 60      // حداقل ضربان نرمال
#define MAX_HR 100     // حداکثر ضربان نرمال
#define MIN_SBP 90     // حداقل فشار سیستولی نرمال
#define MAX_SBP 120    // حداکثر فشار سیستولی نرمال
#define MIN_DBP 60     // حداقل فشار دیاستولی نرمال
#define MAX_DBP 80     // حداکثر فشار دیاستولی نرمال
#define MIN_SPO2 95    // حداقل اکسیژن خون نرمال
#define MAX_TEMP 37.5  // حداکثر دمای نرمال
```

گام به گام انجام پروژه:

۱. ایجاد منوی اصلی

```
void displayMenu() {
    printf("\n=== Patient Monitoring System ===\n");
    printf("1. Add New Patient\n");
    printf("2. Enter Vital Signs\n");
    printf("3. View Patient Data\n");
    printf("4. Check Health Status\n");
    printf("5. Generate Alerts\n");
    printf("6. Exit\n");
    printf("Choose an option: ");
}
```

۲. ساختار اصلی برنامه با switch-case

```

int main() {
    struct Patient patients[100];
    int patientCount = 0;
    int choice;

    do {
        displayMenu();
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                addPatient(patients, &patientCount);
                break;
            case 2:
                enterVitalSigns(patients, patientCount);
                break;
            // سایر موارد ...
        }
    } while(choice != 6);

    return 0;
}

```

۳. تابع اضافه کردن بیمار

```

void addPatient(struct Patient patients[], int *count) {
    if(*count >= 100) {
        printf("Database full!\n");
        return;
    }

    printf("Enter patient ID: ");
    scanf("%d", &patients[*count].id);
    printf("Enter patient name: ");
    scanf("%s", patients[*count].name);
    printf("Enter age: ");
    scanf("%d", &patients[*count].age);

    (*count)++;
    printf("Patient added successfully!\n");
}

```

۴. تابع وارد کردن نشانه‌های حیاتی

```

printf("Enter patient ID: ");
scanf("%d", &id);

for(int i = 0; i < count; i++) {
    if(patients[i].id == id) {
        printf("Enter heart rate (BPM): ");
        scanf("%d", &patients[i].heartRate);
        printf("Enter systolic BP: ");
        scanf("%d", &patients[i].systolicBP);
        printf("Enter diastolic BP: ");
        scanf("%d", &patients[i].diastolicBP);
        printf("Enter blood oxygen (%): ");
        scanf("%d", &patients[i].bloodOxygen);
        printf("Enter body temperature: ");
        scanf("%f", &patients[i].bodyTemp);
        found = 1;
        break;
    }
}

if(!found) {
    printf("Patient not found!\n");
}
}

```

۵. توابع بررسی سلامت

```

int checkHeartRate(int hr) {
    return (hr >= MIN_HR && hr <= MAX_HR);
}

int checkBloodPressure(int systolic, int diastolic) {
    return (systolic >= MIN_SBP && systolic <= MAX_SBP) &&
        (diastolic >= MIN_DBP && diastolic <= MAX_DBP);
}

int checkOxygen(int spo2) {
    return spo2 >= MIN_SPO2;
}

int checkTemperature(float temp) {
    return temp <= MAX_TEMP;
}

```

۶. سیستم هشدار

```

void generateAlerts(struct Patient patients[], int count) {
    printf("\n=== HEALTH ALERTS ===\n");
    int alertsFound = 0;

    for(int i = 0; i < count; i++) {
        int hasAlert = 0;

        if(!checkHeartRate(patients[i].heartRate)) {
            printf(" 🚨 Patient %s: Abnormal heart rate (%d BPM)\n",
                patients[i].name, patients[i].heartRate);
            hasAlert = 1;
        }

        if(!checkBloodPressure(patients[i].systolicBP, patients[i].diastolicBP)) {
            printf(" 🚨 Patient %s: Abnormal BP (%d/%d mmHg)\n",
                patients[i].name, patients[i].systolicBP, patients[i].diastolicBP);
            hasAlert = 1;
        }

        if(!checkOxygen(patients[i].bloodOxygen)) {
            printf(" 🚨 Patient %s: Low oxygen level (%d%%)\n",
                patients[i].name, patients[i].bloodOxygen);
            hasAlert = 1;
        }

        if(hasAlert) {
            alertsFound = 1;
        }
    }

    if(!alertsFound) {
        printf(" 🟢 No critical alerts found.\n");
    }
}

```