№6 ЗЕРТХАНАЛЫҚ ЖҰМЫС
CSV файлдарынан көлік деректерін импорттау және өңдеу

Құралдар: Python 3, Google Colab, csv кітапханасы

1. Жұмыстың мақсаты
Python тілінде CSV форматындағы файлдарды құру, оқу, жазу және өңдеу әдістерін меңгеру. Көлік жүйелерінің өлшеу деректерін CSV файлынан импорттап, олардың минималды, максималды және орташа мәндерін есептеуді, қауіпті көрсеткіштерді анықтауды және өңделген нәтижелерді жаңа CSV файлына сақтауды үйрену.

2. Оқыту нәтижелері
Зертханалық жұмысты орындағаннан кейін білім алушы:
- CSV файлының құрылымын түсіндіре алады;
- open() арқылы файлды аша алады;
- with open() конструкциясын қолдана алады;
- csv.reader() және csv.DictReader() арқылы деректерді оқи алады;
- csv.writer() көмегімен файлға деректер жаза алады;
- мәтіндік деректерді int() және float() арқылы санға түрлендіре алады;
- CSV деректерін тізімге сақтап, талдай алады;
- қауіпті мәндерді анықтап, нәтижелерді жаңа файлға сақтай алады.

CSV дегеніміз не?

CSV (Comma-Separated Values) — кестелік деректерді мәтін түрінде сақтауға арналған файл форматы.
CSV файлында деректер жолдар мен бағандар түрінде ұйымдастырылады. Бағандар көбінесе үтірмен (`,`), кейде нүктелі үтірмен (`;`) бөлінеді.

CSV және Excel айырмашылығы

CSV (.csv)	Excel (.xlsx)
Қарапайым мәтіндік формат	Электрондық кесте форматы
Негізінен деректерді сақтайды	Деректер, формулалар, диаграммалар және форматтау
Python арқылы өңдеуге ыңғайлы	Күрделі кестелермен жұмыс істеуге ыңғайлы
Кез келген мәтіндік редакторда ашылады	Excel немесе басқа кестелік бағдарламамен ашылады



Тапсырма №1. CSV файлын құру
Міндет: қозғалтқыш температурасының 10 өлшеуін CSV файлына сақтау.

```
import csv

temperature = [88, 92, 95, 101, 106, 99, 87, 110, 103, 94]

with open("temperature.csv", "w", newline="") as file:

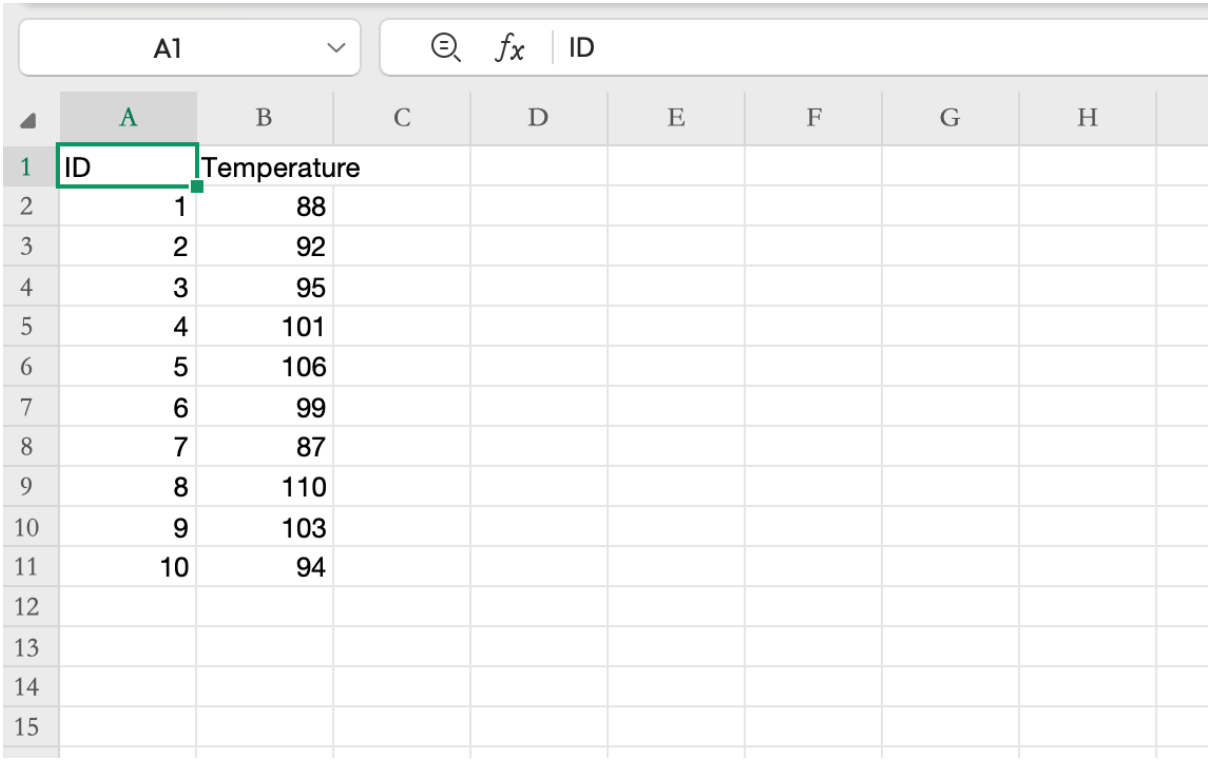
    writer = csv.writer(file)

    writer.writerow(["ID", "Temperature"])

    for i in range(len(temperature)):
        writer.writerow([i + 1, temperature[i]])

print("CSV file created successfully")
```

Бағдарлама temperature.csv файлын құрады.



	A	B	C	D	E	F	G	H
1	ID	Temperature						
2	1	88						
3	2	92						
4	3	95						
5	4	101						
6	5	106						
7	6	99						
8	7	87						
9	8	110						
10	9	103						
11	10	94						
12								
13								
14								
15								

```
import os
print(os.path.abspath("temperature.csv"))
```

арқылы дискте орналасуын тексеруге болады
/content/temperature.csv

```python
import csv  # CSV файлдарымен жұмыс істеуге арналған модуль

# Температураның 10 өлшенген мәні
temperature = [88, 92, 95, 101, 106, 99, 87, 110, 103, 94]

# CSV файлын құру және жазу режимінде ашу
with open("temperature.csv", "w", newline="") as file:

    # CSV файлына деректер жазатын объект құру
    writer = csv.writer(file)

    # Кестенің бірінші жолына баған атауларын жазу
    writer.writerow(["ID", "Temperature"])

    # Тізімдегі барлық температура мәндерін кезекпен өңдеу
    for i in range(len(temperature)):  # len() = 10, i = 0, 1, 2, ..., 9
```

```python
        # ID нөмірін және температура мәнін файлға жазу
        writer.writerow([i + 1, temperature[i]])
        # i + 1          → өлшеудің реттік нөмірі (1–10)
        # temperature[i] → тізімдегі ағымдағы температура

# Файлдың сәтті құрылғаны туралы хабарлама шығару
print("CSV file created successfully")
```

Тапсырма №2. CSV файлынан деректерді оқу

Енді алдыңғы тапсырмада құрылған файлды оқимыз.

```
import csv

with open("temperature.csv", "r") as file:

  reader = csv.reader(file)

  for row in reader:
    print(row)
```

```
['ID', 'Temperature']
['1', '88']
['2', '92']
['3', '95']
['4', '101']
['5', '106']
['6', '99']
['7', '87']
['8', '110']
['9', '103']
['10', '94']
```

Назар аударыңыз: CSV файлынан оқылған мәндер бастапқыда мәтін (str) ретінде алынады.
Мысалы:
value = "95"

Оны санға түрлендіру үшін: value = float(value)

қолданылады.

```
import csv  # CSV файлдарымен жұмыс істеуге арналған модульді қосу

# temperature.csv файлын оқу режимінде ашу
with open("temperature.csv", "r") as file:  # "r" – файлды оқу режимі (read)

    reader = csv.reader(file)  # CSV файлындағы деректерді оқитын объект құру

    for row in reader:  # Файлдағы әрбір жолды кезекпен оқу

        print(row)  # Оқылған жолдың мазмұнын экранға шығару
```

Маңызды айырмашылық

Код	Қызметі
open("temperature.csv", "w")	Файлға деректер жазу
open("temperature.csv", "r")	Файлдағы деректерді оқу
csv.writer(file)	CSV файлына деректер жазу
csv.reader(file)	CSV файлынан деректер оқу
writer.writerow()	Файлға бір жол жазу
for row in reader:	Файлдағы жолдарды кезекпен оқу
print(row)	Оқылған жолды экранға шығару

Тапсырма №3. CSV деректерін Python тізіміне сақтау
CSV файлындағы температура мәндерін жеке тізімге жинаймыз.

```
import csv
temperature = []

with open("temperature.csv", "r") as file:
    reader = csv.reader(file)
    next(reader)
    for row in reader:
        value = float(row[1])
        temperature.append(value)
print("Temperature:", temperature)
```

Нәтиже:
Temperature: [88.0, 92.0, 95.0, 101.0, 106.0, 99.0, 87.0, 110.0, 103.0, 94.0]

```python
import csv  # CSV файлдарымен жұмыс істеуге арналған модульді қосу

temperature = []  # Температура мәндерін сақтау үшін бос тізім құру

# CSV файлын оқу режимінде ашу
with open("temperature.csv", "r") as file:

    reader = csv.reader(file)  # CSV файлындағы деректерді оқитын объект құру

    next(reader)  # Бірінші жолды (ID, Temperature) өткізіп жіберу

    for row in reader:  # Файлдағы қалған жолдарды кезекпен оқу

        value = float(row[1])  # Екінші бағандағы температураны нақты санға (float) айналдыру

        temperature.append(value)  # Алынған температура мәнін тізімнің соңына қосу

print("Temperature:", temperature)  # Барлық температура мәндерін экранға шығару
```

Маңызды командалар

Команда	Түсіндірме
temperature = []	Бос тізім құру

next(reader)	CSV файлының бірінші жолын өткізіп жіберу
for row in reader:	Файлдағы әрбір жолды кезекпен оқу
row[1]	Екінші бағандағы мәнді алу
float(row[1])	Мәтінді нақты санға айналдыру
temperature.append(value)	Жаңа мәнді тізімнің соңына қосу
print()	Нәтижені экранға шығару

Тапсырма №4. CSV деректерін талдау

Алдыңғы тапсырмадағы temperature тізімін пайдаланыңыз. Мыналарды анықтау қажет:

1. өлшеулер саны;
2. минималды температура;
3. максималды температура;
4. орташа температура;
5. 100°C-тан жоғары мәндер саны.

```
print("Measurements:", len(temperature))
print("Minimum:", min(temperature))
print("Maximum:", max(temperature))

average = sum(temperature) / len(temperature)

print("Average:", average)

dangerous = []

for value in temperature:

    if value > 100:
        dangerous.append(value)

print("Dangerous values:", dangerous)
print("Dangerous count:", len(dangerous))
```

Нәтиже:
Measurements: 10
Minimum: 87.0
Maximum: 110.0
Average: 97.5
Dangerous values: [101.0, 106.0, 110.0, 103.0]
Dangerous count: 4

Тапсырма №5. DictReader көмегімен деректерді оқу

CSV файлында бірнеше баған болса, оларды атауы арқылы оқу ыңғайлы.

Мысалы:
ID,Temperature
1,88
2,92
3,95

csv.DictReader() көмегімен:

```
import csv
with open("temperature.csv", "r") as file:
    reader = csv.DictReader(file)
    for row in reader:
        print(row["ID"], row["Temperature"])
```

Сонда DictReader әр жолды сөздік ретінде қарастырады.
Мысалы:

```
{
    "ID": "1",
    "Temperature": "88"
}
```

Тапсырма №6. Өлшеулерді классификациялау

Әр температураны келесі шарттар бойынша тексеріңіз:

Температура	Күйі
<90°C	LOW
90–100°C	NORMAL
>100°C	OVERHEATING

Алдыңғы зертханалық жұмыста өткен функцияны пайдаланамыз:

```
def check_temperature(temp):

    if temp < 90:
        return "LOW"

    elif temp <= 100:
        return "NORMAL"

    else:
        return "OVERHEATING"
```

```
def check_temperature(temp):  # Температураны тексеретін функция құру
                              # temp — функцияға берілетін температура мәні
```

Енді CSV файлындағы әрбір мәнді тексереміз:

```
import csv

with open("temperature.csv", "r") as file:

    reader = csv.DictReader(file)

    for row in reader:

        temp = float(row["Temperature"])

        status = check_temperature(temp)

        print(temp, "->", status)
```

```python
import csv  # CSV файлдарымен жұмыс істеуге арналған модульді қосу

# temperature.csv файлын оқу режимінде ашу
with open("temperature.csv", "r") as file:

    reader = csv.DictReader(file)  # CSV деректерін баған атаулары арқылы оқу

    for row in reader:  # Файлдағы әрбір жолды кезекпен оқу

        # "Temperature" бағанындағы мәнді алып, нақты санға айналдыру
        temp = float(row["Temperature"])

        # Температураны тексеретін функцияны шақыру
        status = check_temperature(temp)
        # Функция LOW, NORMAL немесе OVERHEATING нәтижесін қайтарады

        # Температура мен оның күйін экранға шығару
        print(temp, "->", status)
```

Негізгі командалардың түсіндірмесі

Команда	Қызметі
csv.DictReader(file)	CSV файлын сөздік (dictionary) түрінде оқиды
for row in reader:	Әрбір жолды кезекпен өңдейді
row["Temperature"]	Temperature бағанындағы мәнді алады
float()	Мәтінді нақты санға айналдырады
check_temperature(temp)	Температураның күйін анықтайтын функцияны шақырады
status	Функция қайтарған нәтижені сақтайды
print(temp, "->", status)	Температура мен оның күйін экранға шығарады

csv.reader() және csv.DictReader() айырмашылығы

Бұрынғы бағдарламада:
reader = csv.reader(file)

temp = float(row[1]) # Екінші бағандағы мәнді алу

Ал жаңа бағдарламада:
reader = csv.DictReader(file)
temp = float(row["Temperature"]) # Баған атауы арқылы мәнді алу

Негізгі айырмашылық:
csv.reader() бағандарды индекс арқылы оқиды (row[1]),
ал csv.DictReader() бағандарды атауы арқылы оқиды (row["Temperature"]).

DictReader() бірінші жолдағы баған атауларын автоматты түрде пайдаланатындықтан, next(reader) командасы қажет емес.

Тапсырма №7. Нәтижелерді жаңа CSV файлына сақтау
Енді бастапқы CSV файлына талдау нәтижелерін қосып, жаңа файл құрамыз.

Жаңа файлдың құрылымы:

ID	Temperature	Status
1	88	LOW
2	92	NORMAL
3	95	NORMAL
4	101	OVERHEATING
5	106	OVERHEATING

Бағдарлама:

```
import csv

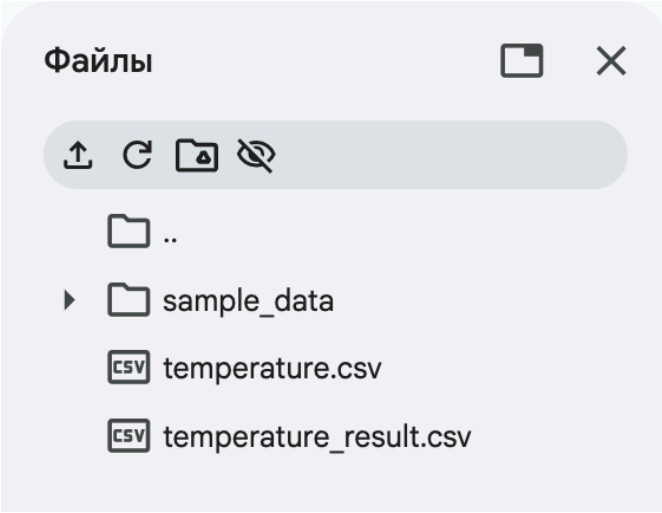
def check_temperature(temp):

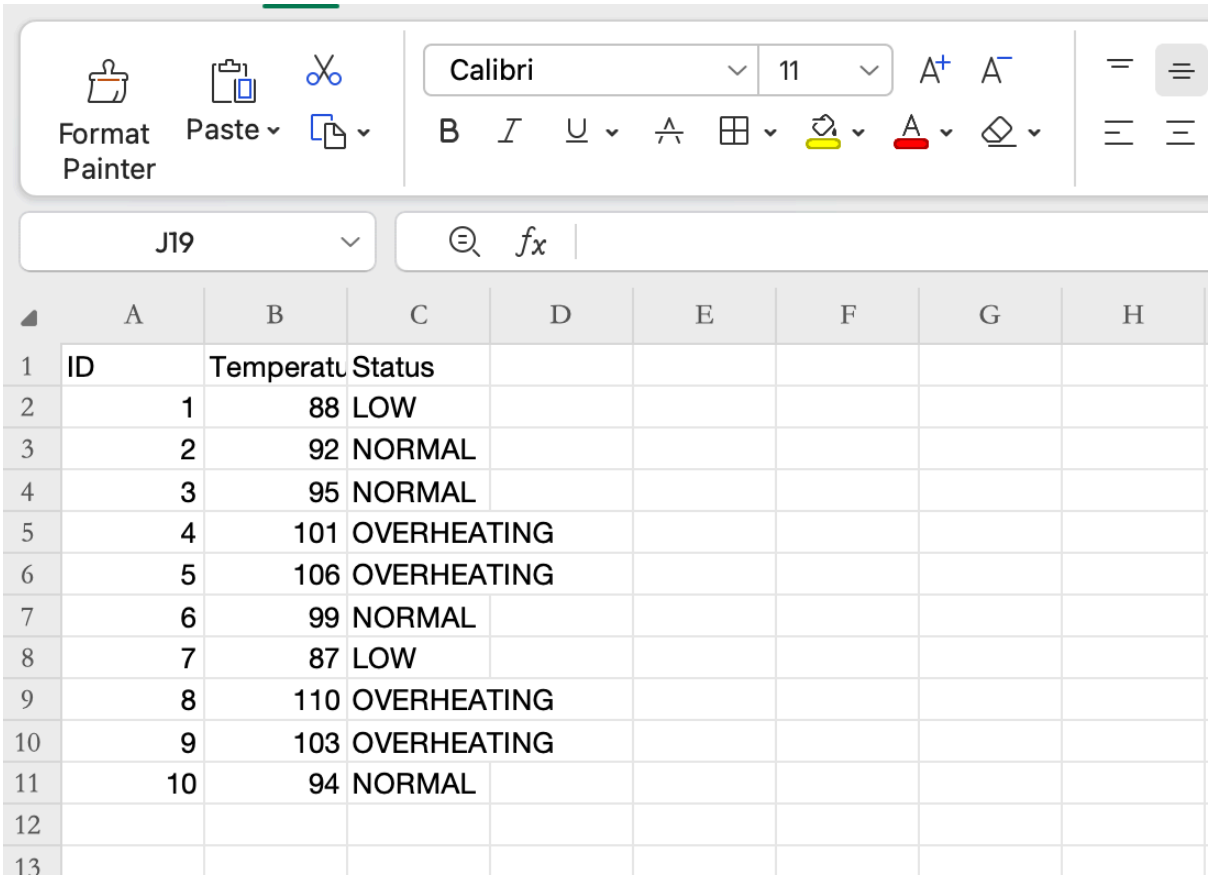
    if temp < 90:
        return "LOW"

    elif temp <= 100:
        return "NORMAL"
```

```
    else:
        return "OVERHEATING"


with open("temperature.csv", "r") as input_file:
    reader = csv.DictReader(input_file)
    with open("temperature_result.csv", "w", newline="") as output_file:
        writer = csv.writer(output_file)
        writer.writerow(["ID", "Temperature", "Status"])
        for row in reader:
            temp = float(row["Temperature"])
            status = check_temperature(temp)
            writer.writerow([row["ID"], temp, status])
print("Result file created successfully")
```





	A	B	C	D	E	F	G	H
1	ID	Temperatu	Status					
2	1	88	LOW					
3	2	92	NORMAL					
4	3	95	NORMAL					
5	4	101	OVERHEATING					
6	5	106	OVERHEATING					
7	6	99	NORMAL					
8	7	87	LOW					
9	8	110	OVERHEATING					
10	9	103	OVERHEATING					
11	10	94	NORMAL					
12								
13								

Нәтижесінде temperature_result.csv файлы пайда болады.

Негізгі жеке тапсырма
Әр білім алушы өз нұсқасына сәйкес көлік жүйесінің өлшеу деректерін CSV файлы арқылы өңдейтін бағдарлама әзірлеуі тиіс.
Бағдарлама келесі әрекеттерді орындауы қажет:

1. 10 өлшеуден тұратын бастапқы CSV файлын құру.
2. CSV файлын оқу.
3. Өлшеу мәндерін тізімге сақтау.
4. Минималды, максималды және орташа мәндерді есептеу.
5. Әр өлшеуді берілген диапазондарға сәйкес классификациялау.
6. Әр санаттағы мәндер санын анықтау.
7. Қауіпті немесе қалыпты диапазоннан тыс мәндерді табу.
8. Өлшеу мәні мен күйі көрсетілген жаңа CSV файлын құру.
9. Нәтижелерді экранға шығару.
10. Қорытынды жасау.

Программа должна выполнять следующие действия: (на русском)

1. Создать исходный CSV-файл, содержащий 10 измерений.
2. Прочитать данные из CSV-файла.
3. Сохранить измеренные значения в список.
4. Рассчитать минимальное, максимальное и среднее значения.
5. Классифицировать каждое измерение в соответствии с заданными диапазонами.
6. Определить количество значений в каждой категории.
7. Выявить опасные значения или значения, выходящие за пределы нормального диапазона.
8. Создать новый CSV-файл, содержащий измеренные значения и соответствующие им состояния.
9. Вывести результаты обработки данных на экран.
10. Сделать вывод по результатам выполненной работы.

Жеке тапсырмалар №1–10

№	Тақырып	CSV бағандары	Классификация шарттары
1	Қозғалтқыш температурасын бақылау	ID,Temperature	<90 LOW, 90–100 NORMAL, >100 OVERHEATING
2	Автомобиль жылдамдығын талдау	ID,Speed	<50 LOW, 50–90 NORMAL, >90 HIGH
3	Май қысымын бақылау	ID,Pressure	<2 LOW, 2–4 NORMAL, >4 HIGH
4	EV батареясының кернеуін талдау	ID,Voltage	<250 LOW, 250–450 NORMAL, >450 HIGH
5	Батарея зарядын бақылау	ID,SOC	<20 LOW, 20–80 NORMAL, >80 HIGH
6	Шина қысымын бақылау	ID,TirePressure	<2.0 LOW, 2.0–2.5 NORMAL, >2.5 HIGH
7	EV батареясының температурасын бақылау	ID,BatteryTemperature	<15 LOW, 15–45 NORMAL, >45 HIGH

8	Отын шығынын талдау	ID,FuelConsumption	<6 LOW, 6–10 NORMAL, >10 HIGH
9	Қозғалтқыш айналымын бақылау	ID,RPM	<1000 LOW, 1000–4000 NORMAL, >4000 HIGH
10	Салқындатқыш сұйықтық температурасын бақылау	ID,CoolantTemperature	<80 LOW, 80–100 NORMAL, >100 OVERHEATING

Күрделендірілген жеке тапсырмалар №11–20
Бұл тапсырмаларда CSV файлында екі өлшенетін параметр болады. Студент әр жолды талдап, екі параметрдің комбинациясы бойынша жүйенің күйін анықтауы тиіс.

№	Тақырып	CSV бағандары	Шарттар
11	EV батареясының кернеуі мен температурасын бақылау	ID,Voltage,Temperature	Кернеу 300–420 V, температура 15–45°C болса → NORMAL. Кемінде бір параметр диапазоннан тыс болса → WARNING. Температура >55°C болса → CRITICAL.
12	Қозғалтқыш температурасы мен май қысымын бақылау	ID,Temperature,OilPressure	Температура ≤100°C және қысым ≥2 bar болса → NORMAL. Температура >110°C немесе қысым <1.5 bar болса → CRITICAL. Қалған жағдайлар → WARNING.

13	Жылдамдық пен қозғалтқыш айналымын талдау	ID,Speed,RPM	Жылдамдық ≤90 км/сағ және RPM ≤4000 болса → NORMAL. Жылдамдық >120 немесе RPM >5000 болса → CRITICAL. Қалған жағдайлар → WARNING.
14	EV батареясының SOC және температурасы	ID,SOC,Temperature	SOC ≥20% және температура ≤45°C болса → NORMAL. SOC <10% немесе температура >55°C болса → CRITICAL. Қалған жағдайлар → WARNING.
15	Шина қысымы мен температурасы	ID,TirePressure,TireTemperature	Қысым 2.1–2.5 bar және температура ≤60°C болса → NORMAL. Қысым <1.7 bar немесе температура >80°C болса → CRITICAL. Қалған жағдайлар → WARNING.
16	Отын шығыны мен жылдамдықты талдау	ID,FuelConsumption,Speed	Шығын ≤9 л/100 км және жылдамдық ≤90 км/сағ болса → NORMAL. Шығын >12 немесе жылдамдық >120 болса → CRITICAL. Қалған жағдайлар → WARNING.
17	Салқындату жүйесінің температурасын талдау	ID,EngineTemp,CoolantTemp	Екі температура да ≤100°C болса → NORMAL. Кемінде бір температура

			>110°C болса → CRITICAL. Қалған жағдайлар → WARNING.
18	EV батареясының кернеуі мен SOC деңгейін бақылау	ID,Voltage,SOC	Кернеу 300–420 V және SOC ≥20% болса → NORMAL. Кернеу <250 V немесе SOC <10% болса → CRITICAL. Қалған жағдайлар → WARNING.
19	Автомобиль жылдамдығы мен кедергіге дейінгі қашықтық	ID,Speed,Distance	Қашықтық ≥100 cm болса → NORMAL. Қашықтық <20 cm болса → CRITICAL. Қалған жағдайлар → WARNING.
20	Көлік жүйесінің кешенді мониторингі	ID,Temperature,Voltage,Speed	Температура ≤100°C, кернеу 300–420 V, жылдамдық ≤90 км/сағ болса → NORMAL. Температура >110°C немесе кернеу <250 V болса → CRITICAL. Қалған жағдайлар → WARNING.

Бақылау сұрақтары

1. CSV форматы дегеніміз не?
2. CSV файлының Excel файлынан айырмашылығы қандай?
3. open() функциясы не үшін қолданылады?
4. "r", "w" және "a" режимдерінің айырмашылығы қандай?
5. with open() қолданудың артықшылығы қандай?
6. csv.reader() қандай қызмет атқарады?
7. csv.DictReader() қандай қызмет атқарады?
8. CSV файлының бірінші жолында әдетте қандай ақпарат сақталады?
9. next(reader) не үшін қолданылады?
10. CSV файлынан оқылған сандық мәндерді неге float() арқылы түрлендіреміз?
11. writer.writerow() не үшін қолданылады?
12. writer.writerows() пен writer.writerow() айырмашылығы қандай?
13. CSV файлынан алынған деректерді тізімге қалай сақтауға болады?
14. CSV деректерін өңдеу кезінде функцияларды қалай қолдануға болады?
15. Неліктен өңделген деректерді жаңа CSV файлына сақтау пайдалы?