



배터리팩 품질관리



프로그램

C# WinForms 기반

품질 데이터 분석 및 시각화 프로그램

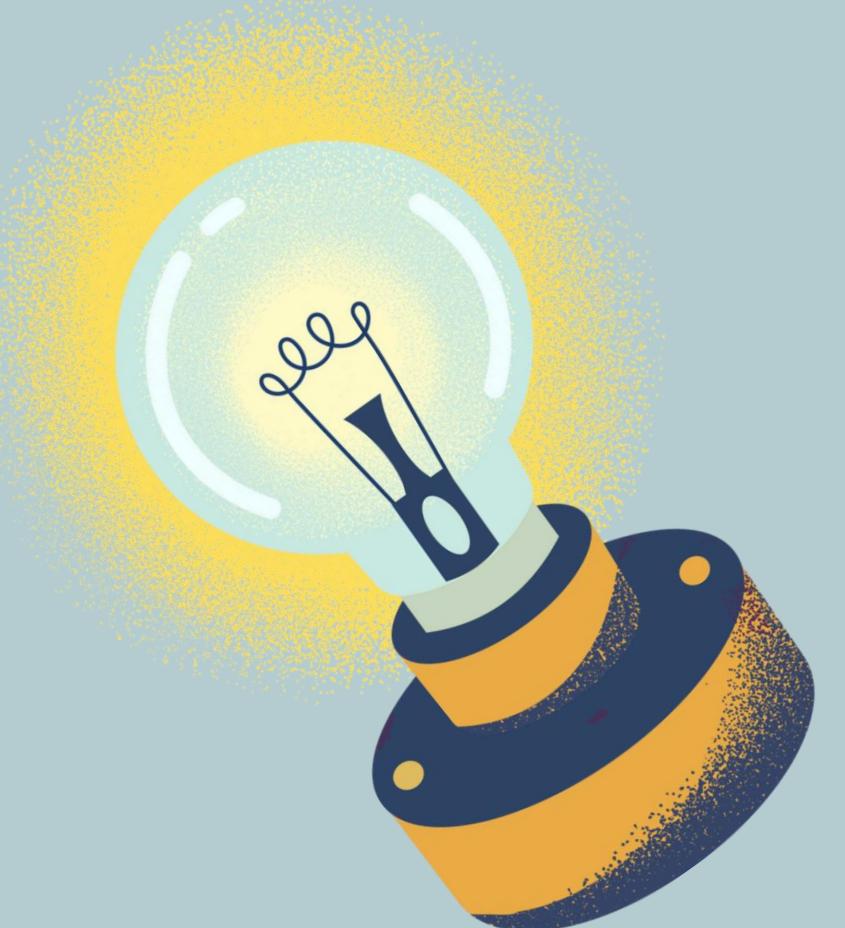
Tech Stack

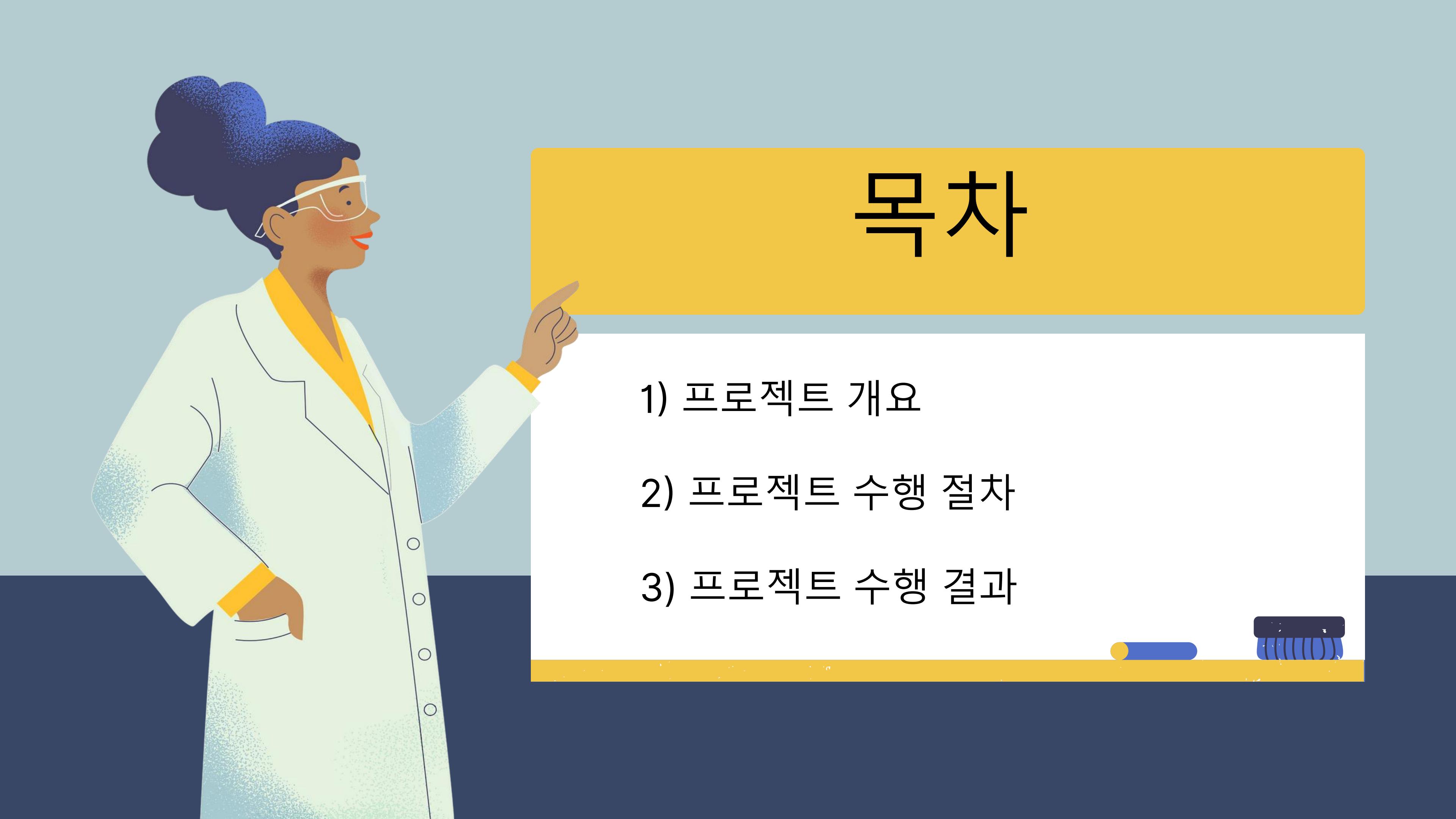
Language: C#

Framework: WinForms .NET Framework

Tools: Visual Studio GitHub

DB: MSSQL





목차

- 1) 프로젝트 개요
- 2) 프로젝트 수행 절차
- 3) 프로젝트 수행 결과

1) 프로젝트 개요

ANSWER
KEY

전자부품(배터리팩)의 품질관리

- 본 프로젝트는 배터리팩 생산 과정에서 발생하는 검사 데이터를 기반으로 양품/불량 여부를 분석하고 품질 현황을 관리하는 프로그램입니다.
- 검사 결과 데이터를 날짜별로 집계하여 불량률과 품질 추이를 확인할 수 있도록 구성했으며, 결과를 그래프 및 표 형태로 시각화하여 관리자가 품질 상태를 빠르게 파악할 수 있도록 구현했습니다.
- 해당 프로젝트는 직업학교 교육 과정에서 팀 프로젝트로 수행한 경험이며, 데이터 처리 로직과 품질 분석 기능 구현을 중심으로 개발을 진행했습니다.

1) 프로젝트 개요

활용 데이터

제조AI데이터셋

전자부품(배터리팩) 예지보전 AI 데이터셋

전기차용 배터리모듈 용접 불량 분석을 위한 레이저 용접기 데이터

 제조AI데이터셋

 가이드북

업종	전자부품	유형	CSV
목적	예지보전	제조데이터 등록일	2022.12.23
사용조건	콘텐츠 변경허용	최종 수정일자	2023.08.14

제공기관
스마트제조혁신추진단
(수행기관 : (주)인터엑스/네스트필드(주))

태그
[#전기차](#), [#EV](#), [#전기차 배터리](#), [#배터리 용접](#), [#용접기](#), [#레이저 용접기](#), [#예지보전](#), [#N-HITS](#), [#AAS](#)

적용공정
배터리 용접 공정

내용
제조AI데이터셋 소개
전기차용 배터리모듈 용접에 사용되는 용접설비 데이터(용접출력, 작업속도, 출력설정, 작업상태)를 기준으로 설비의 이상 발생 징후를 예측하기 위한 제조 AI분석과정을 담은 데이터셋과 가이드북입니다. 용접설비로부터 운영 데이터를 수집하고, 장시간 높은 예측 정확도와 빠른 계산속도를 가지는 시계열 예측 알고리즘 N-HITS를 학습시켜 설비의 정상/비정상 상태 예측을 도모합니다.

생산 로그 데이터(CSV)를
기반으로 검사 결과를
정리하였습니다.

1) 프로젝트 개요

개발 기간:

24.04 ~ 24.04

개발 인원:

4명

담당 업무:

생산/검사 데이터(양품/불량)를 날짜 기준으로 집계 및 분석 로직 구현

품질 현황 시각화 기능 개발 (막대그래프/원형그래프/꺾은선그래프 적용)

불량 유형 및 발생 추이를 한눈에 확인할 수 있도록 UI 화면 구성

결과 데이터 필터링(기간 선택) 및 조회 기능 구현

팀 프로젝트 내 데이터 분석/시각화 파트 담당 및 기능 통합 지원

2) 프로젝트 수행절차

MSSQL 테이블 데이터 구조

ProductData Table			
논리이름	물리이름	데이터 타입	비고
용접시퀀스	PageNo	NVARCHAR(50)	(Count)
용접속도설정	Speed	NVARCHAR(50)	(mm/s)
용접길이설정	Length	NVARCHAR(50)	(mm)
용접출력	RealPower	NVARCHAR(50)	(W)
발광횟수설정	SetFrequency	NVARCHAR(50)	(Hz)
최대용접출력설정	SetDuty	NVARCHAR(50)	(%)
용접출력설정	SetPower	NVARCHAR(50)	(%)
용접시간	GateOnTime	NVARCHAR(50)	(s)
작업시간	WorkingTime	NVARCHAR(50)	-

2) 프로젝트 수행절차

CSV 데이터 유형/구조

	A	B	C	D	E	F	G	H	I
1	PageNo	Speed	Length	RealPower	SetFrequency	SetDuty	SetPower	GateOnTime	WorkingTime
2	1	250	241.1	1660	1000	100	82	1154	2022-02-11 09:12:28.065
3	2	250	241.2	1685	1000	100	83	1670	2022-02-11 09:12:31.355
4	3	250	241.1	1666	1000	100	82	1153	2022-02-11 09:12:34.187
5	4	250	241.2	1690	1000	100	83	1670	2022-02-11 09:12:37.429
6	5	250	241.1	1670	1000	100	82	1154	2022-02-11 09:12:40.189
7	6	250	241.2	1693	1000	100	83	1670	2022-02-11 09:12:43.509
8	7	250	241.1	1669	1000	100	82	1153	2022-02-11 09:12:49.277
9	8	250	241.2	1694	1000	100	83	1670	2022-02-11 09:12:52.547
10	9	250	241.1	1673	1000	100	82	1154	2022-02-11 09:12:55.389
11	10	250	241.2	1695	1000	100	83	1670	2022-02-11 09:12:58.755
12	11	250	241.1	1675	1000	100	82	1154	2022-02-11 09:13:01.615
13	12	250	241.2	1698	1000	100	83	1670	2022-02-11 09:13:04.827
14	13	30	19.4	682	1000	100	38	650	2022-02-11 09:13:09.307
15	14	30	19.4	680	1000	100	38	650	2022-02-11 09:13:11.019
16	15	30	19.4	682	1000	100	38	650	2022-02-11 09:13:13.949
17	16	30	19.4	680	1000	100	38	650	2022-02-11 09:13:17.335

현장에서 수집된
CSV 검사 로그 데이터를
파싱하여 DB에 저장

Winform 구성



Winform 구성

Winform은 Windows Forms의 약어로, Microsoft가 개발한 윈도우 환경에서 실행되는 GUI(그래픽 사용자 인터페이스) 애플리케이션입니다. 이 차트 예제는 Winform을 사용하여 데이터 분석과 시각화를 수행하는 방법을 보여주는 것입니다.

상단에는 데이터 분석 화면이 표시됩니다. 화면에는 ' pageNo'라는 텍스트박스와 사진이 있습니다. 사진에는 배터리 셀에 있는 용접 포인트가 강조되어 있습니다.

설명文: pageNo는 용접 공정 번호를 뜻합니다. 저희 자료의 배터리 셀에는 총 39개의 용접 포인트가 있고, pageNo는 총 39번까지 있습니다.

아래에는 차트가 표시되는 영역입니다. 차트들은 ' pageNo'라는 텍스트박스와 함께 배치되었습니다. 차트에는 'Legend1 - 비어 있음'이라는 제목이 포함되어 있습니다.

차트에는 LINE, BAR 차트가 포함되어 있습니다.

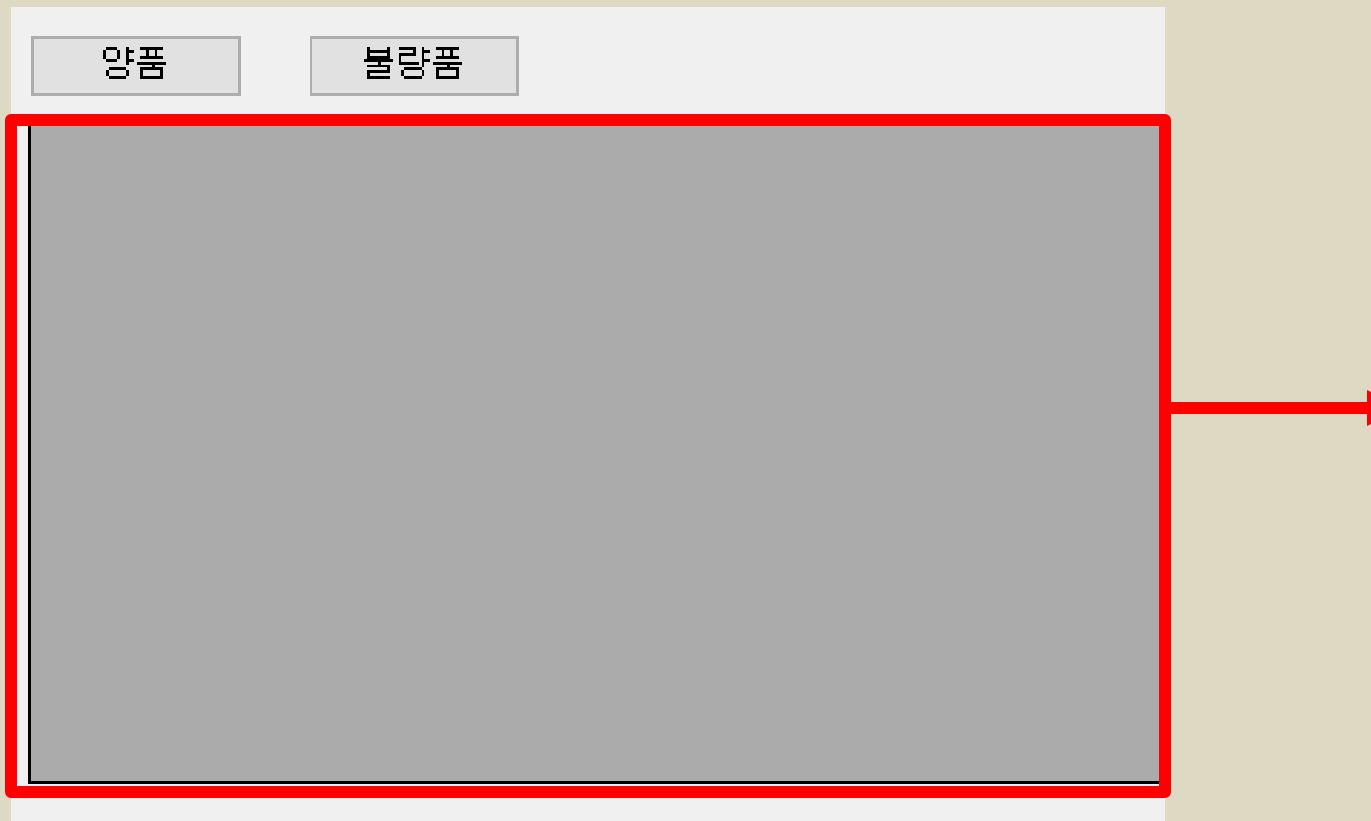
LINE 차트 데이터:

PageNo	Value
1	75
2	80
3	82
4	70
5	20
6	80
7	40

BAR 차트 데이터:

PageNo	Value
1	70
2	75
3	80
4	70
5	20
6	80
7	40

Winform 구성



그리드 뷰를 클릭하면 보고서
폼이 등장합니다. 문제가 있는
데이터에 대한 보고서를
작성하고 txt파일로 저장합니다.

A screenshot of a 'Report' form window titled '보고서 제목'. The window has a blue header bar with the title and standard window controls. The main area contains several text input fields and text boxes:

- '작성일 : 시간' (Creation Date : Time) with an input field.
- '작성자 :' (Creator) with an input field.
- '개요' (Summary) with a large text box.
- '품질 현황' (Quality Status) with a large text box.
- '문제점 및 개선 방향' (Problem Point and Improvement Direction) with a large text box.
- '조치 및 결과' (Measures and Results) with a large text box.

A '저장' (Save) button is located in the top right corner of the form.

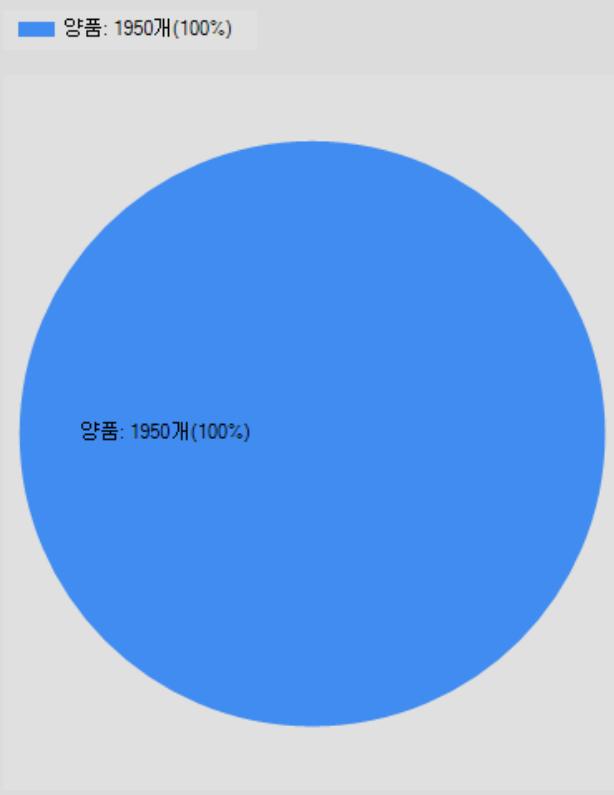
2) 프로젝트 수행 절차

```
// 불량품이 없을 때에는 양품만 보여주는 파이차트 코드  
참조 1개  
private void Chart_Click(object sender, EventArgs e)  
{  
    try  
    {  
        ConnectDB();  
        chart1.Series[0].Points.Clear();  
  
        //양품 데이터  
        SqlCommand cmdGood = new SqlCommand();  
        cmdGood.Connection = conn;  
  
        if (comboBox1.Text == "All")  
        {  
            cmdGood.CommandText = "SELECT COUNT(*) AS TotalGoodCount FROM ProductData WHERE (RealPower BETWEEN 1650 AND 1750 OR RealPower BETWEEN 670 AND 720)";  
        }  
        else  
        {  
            cmdGood.CommandText = "SELECT COUNT(*) AS GoodCount FROM ProductData " +  
                "WHERE CONVERT(date, WorkingTime) = @SelectedDate " +  
                "AND (RealPower BETWEEN 1650 AND 1750 OR RealPower BETWEEN 670 AND 720)";  
  
            cmdGood.Parameters.AddWithValue("@SelectedDate", comboBox1.SelectedItem.ToString());  
        }  
        int GoodCount = Convert.ToInt32(cmdGood.ExecuteScalar()); //ToInt32 - 문자형식을 숫자형식으로 변환  
  
        //불량품 데이터  
        SqlCommand cmdDefective = new SqlCommand();  
        cmdDefective.Connection = conn;  
  
        if (comboBox1.Text == "All")  
        {  
            cmdDefective.CommandText = "SELECT COUNT(*) AS TotalDefectiveCount FROM ProductData WHERE ([RealPower] < 1650 OR [RealPower] > 1750) AND ([RealPower] < 670 OR [RealPower] > 720)";  
        }  
        else  
        {  
            cmdDefective.CommandText = "SELECT COUNT(*) AS DefectiveCount FROM ProductData " +  
                "WHERE CONVERT(date, WorkingTime) = @SelectedDate " +  
                "AND ([RealPower] < 1650 OR [RealPower] > 1750) AND ([RealPower] < 670 OR [RealPower] > 720)";  
  
            cmdDefective.Parameters.AddWithValue("@SelectedDate", comboBox1.SelectedItem.ToString());  
        }  
        int DefectiveCount = Convert.ToInt32(cmdDefective.ExecuteScalar());
```

RealPower(출력)의
정상범위와 그렇지
않은것으로 SQL문 작성

2) 프로젝트 수행 절차

불량품이 없는 경우



```
//전체 데이터
SqlCommand cmdTotal = new SqlCommand();
cmdTotal.Connection = conn;

if (comboBox1.Text == "All")
{
    cmdTotal.CommandText = "SELECT COUNT(*) AS TotalAllCount FROM ProductData";
}
else
{
    cmdTotal.CommandText = "SELECT COUNT(*) AS TotalCount FROM ProductData " +
        "WHERE CONVERT(date, WorkingTime) = @SelectedDate";

    cmdTotal.Parameters.AddWithValue("@SelectedDate", comboBox1.SelectedItem.ToString());
}

int TotalCount = Convert.ToInt32(cmdTotal.ExecuteScalar());

//양품, 불량품 퍼센트
double GoodData = ((double)GoodCount / TotalCount) * 100.0;
GoodData = Math.Round(GoodData, 1);
double DefectiveData = ((double)DefectiveCount / TotalCount) * 100.0;
DefectiveData = Math.Round(DefectiveData, 1);

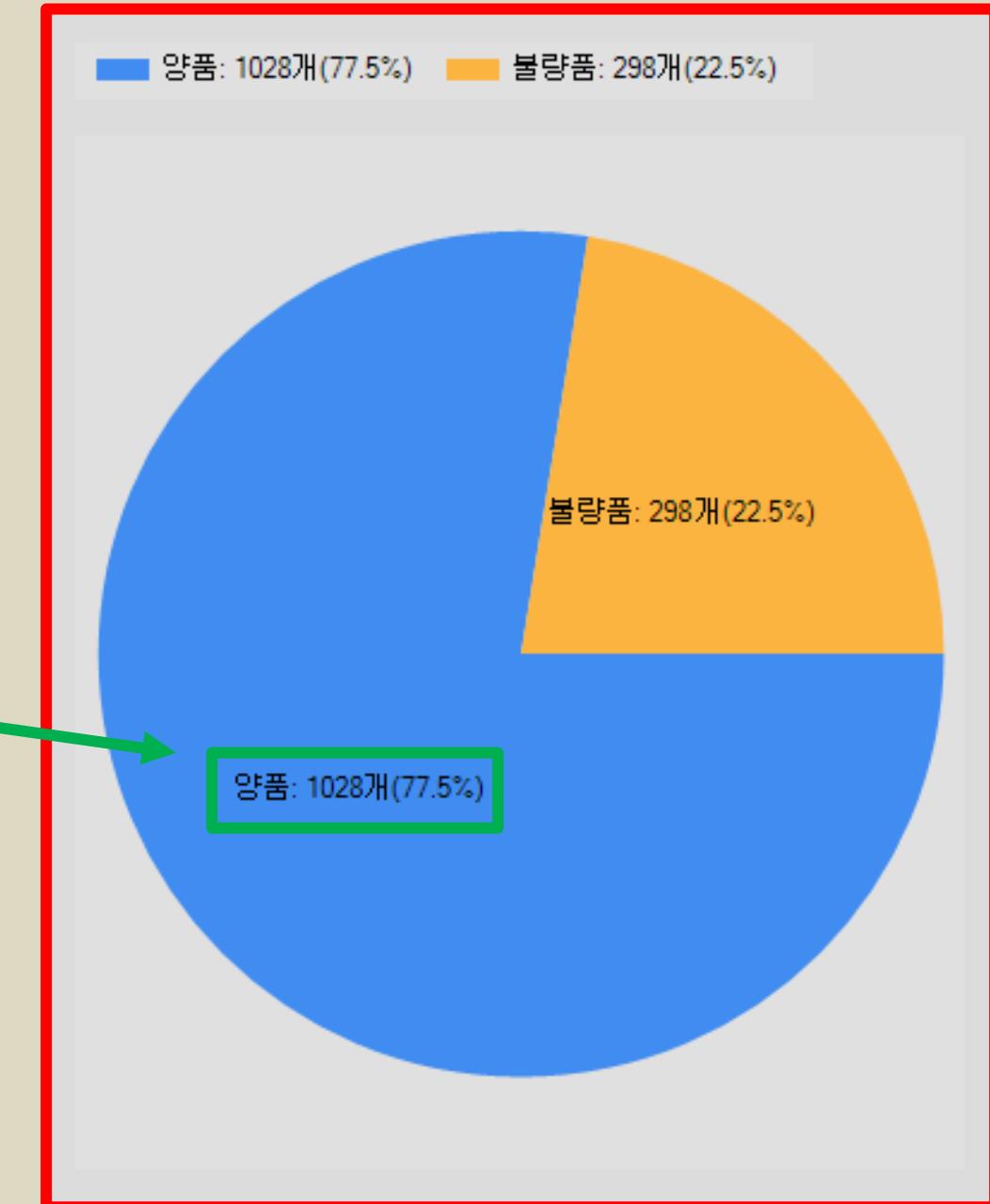
//양품, 불량품 개수
int GoodDataCount = Convert.ToInt32(cmdGood.ExecuteScalar());
int DefectiveDataCount = Convert.ToInt32(cmdDefective.ExecuteScalar());

chart1.Series[0].Points.AddXY("양품: " + GoodDataCount + "개 (" + GoodData + "%)", GoodCount);

// 불량품이 있는 경우에만 파이차트에 추가
if (DefectiveCount > 0)
{
    chart1.Series[0].Points.AddXY("불량품: " + DefectiveDataCount + "개 (" + DefectiveData + "%)", DefectiveCount);
}

catch (Exception ex)
{
    MessageBox.Show("날짜를 선택해 주세요");
}
finally
{
    conn.Close();
}
```

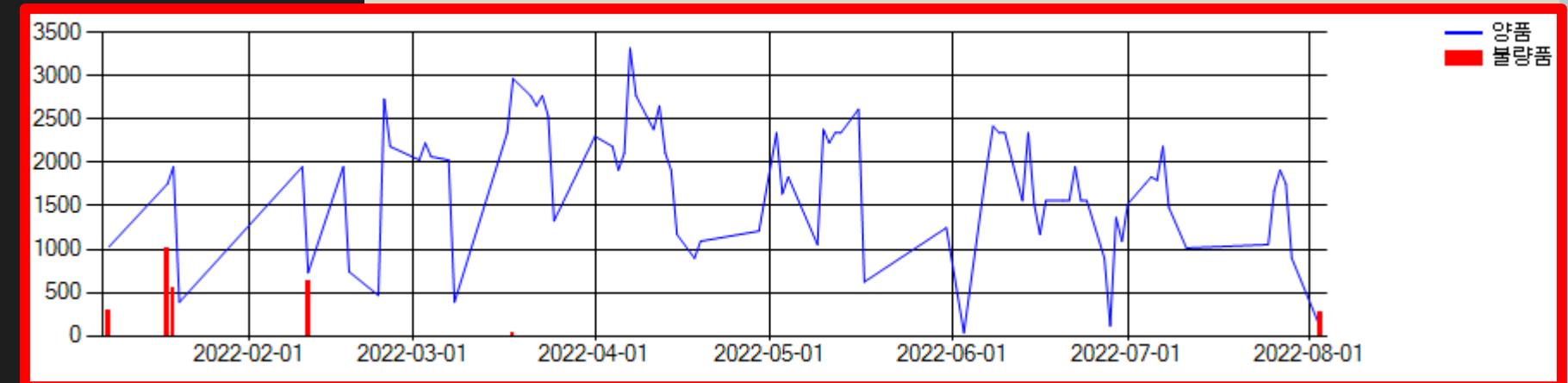
불량품이 있는 경우
파이차트에 표시되도록 함



양품과 불량품의 비율을
구하기 위해 GoodCount,
DefectiveCount, TotalCount
변수를 사용해 계산

2) 프로젝트 수행 절차

```
37 try
38 {
39     ConnectDB();
40     SqlCommand cmd = new SqlCommand();
41     cmd.Connection = conn;
42     string sql = "";
43     string sql2 = "";
44     List<string> date = new List<string>();
45
46     sql = "select count(*) as count, workingtime from productdata " +
47         "where(realpower between 1650 and 1750 or realpower between 670 and 720) " +
48         "group by workingtime order by workingtime"; // 양품
49     cmd.CommandText = sql;
50     SqlDataReader dr = cmd.ExecuteReader();
51
52     while (dr.Read())
53     {
54         date.Add(dr[1].ToString());
55         chart2.Series[0].Points.AddXY(DateTime.Parse(dr[1].ToString()), int.Parse(dr[0].ToString()));
56     }
57     dr.Close();
58
59     sql2 = "select count(*) as count, workingtime from productdata " +
60         "WHERE([RealPower] < 1650 OR[RealPower] > 1750) AND([RealPower] < 670 OR[RealPower] > 720) " +
61         "group by workingtime order by workingtime"; // 불량품
62     cmd.CommandText = sql2;
63     SqlDataReader dr2 = cmd.ExecuteReader();
64     while (dr2.Read())
65     {
66         chart2.Series[1].Points.AddXY(DateTime.Parse(dr2[1].ToString()), int.Parse(dr2[0].ToString()));
67     }
68     dr2.Close();
69 }
70 catch (Exception ex)
71 {
72     MessageBox.Show(ex.StackTrace);
73     MessageBox.Show(ex.Message);
74 }
75 finally { conn.Close(); }
```



양품은 선 그래프
불량품은 막대그래프로 표시

2) 프로젝트 수행 절차

```
//DB(데이터베이스)에 있는 양품 데이터만 보여주는 버튼  
참조 1개  
private void GoodData_button_Click(object sender, EventArgs e)  
{  
    try  
{  
        ConnectDB();  
        SqlCommand cmd = new SqlCommand();  
        cmd.Connection = conn;  
  
        if (comboBox1.Text == "All")  
        {  
            cmd.CommandText = "SELECT TRIM(PageNo) AS PageNo, TRIM(Speed) AS Speed, TRIM(Length) AS Length, TRIM(RealPower) AS RealPower, TRIM(SetFrequency) AS SetFrequency,"  
            + " TRIM(SetDuty) AS SetDuty, TRIM(SetPower) AS SetPower, TRIM(GateOnTime) AS GateOnTime, TRIM(WorkingTime) AS WorkingTime FROM ProductData"  
            + " WHERE (RealPower BETWEEN 1650 AND 1750 OR RealPower BETWEEN 670 AND 720) order by WorkingTime";  
        }  
        else  
        {  
            cmd.CommandText = "SELECT TRIM(PageNo) AS PageNo, TRIM(Speed) AS Speed, TRIM(Length) AS Length, TRIM(RealPower) AS RealPower, TRIM(SetFrequency) AS SetFrequency, " +  
                "TRIM(SetDuty) AS SetDuty, TRIM(SetPower) AS SetPower, TRIM(GateOnTime) AS GateOnTime, TRIM(WorkingTime) AS WorkingTime " +  
                "FROM ProductData " +  
                "WHERE CONVERT(date, WorkingTime) = @SelectedDate " +  
                "AND (RealPower BETWEEN 1650 AND 1750 OR RealPower BETWEEN 670 AND 720)";  
  
            cmd.Parameters.AddWithValue("@selectedDate", comboBox1.SelectedItem.ToString());  
        }  
  
        SqlDataAdapter da = new SqlDataAdapter(cmd);  
        DataSet ds = new DataSet();  
        da.Fill(ds, "mytest");  
  
        dataGridView2.DataSource = ds.Tables["mytest"];  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show("날짜를 선택해주세요");  
    }  
    finally  
    {  
        conn.Close();  
    }  
}
```

양품과 불량품을
DataGridView에 띄우기
위한 코드

	PageNo	Speed	Length	RealPower
▶	21	250	241,1	1721
	22	250	241,2	1700
	23	250	241,1	1723
	24	250	241,2	1703
	25	250	241,1	1721
	26	250	241,2	1702
	27	250	241,1	1724
	28	250	241,2	1704
	29	250	241,1	1726

	PageNo	Speed	Length	RealPower
▶	1	250	241,1	1810
	2	250	241,2	1900
	3	250	241,1	1810
	4	250	241,2	1900
	5	250	241,1	1810
	6	250	241,2	1900
	7	250	241,1	1800
	8	250	241,2	1900
	9	250	241,1	1810

2) 프로젝트 수행 절차

```
333     private void dataGridView2_CellClick(object sender, DataGridViewCellEventArgs e)
334     {
335         string selectedData = "";
336
337         Form fc = Application.OpenForms["Reports"];
338
339         if (report != null)
340             report.Dispose();
341
342         if (fc != null)
343         {
344             if (e.RowIndex >= 0)
345             {
346                 selectedData = "Wn";
347                 DataGridViewRow selectedRow = dataGridView2.Rows[e.RowIndex];
348                 for (int i = 0; i < 9; i++)
349                 {
350                     selectedData += selectedRow.Cells[i].Value.ToString() + "    ";
351                 }
352                 report.Show(selectedData, true);
353             }
354         }
355         if (fc == null)
356         {
357             if (e.RowIndex >= 0)
358             {
359                 selectedData = "PageNO    Speed    Length    RealPower    SetFrequency    SetDuty    SetPower    GateOnTime    WorkingTimeWn";
360                 DataGridViewRow selectedRow = dataGridView2.Rows[e.RowIndex];
361                 for (int i = 0; i < 9; i++)
362                 {
363                     selectedData += selectedRow.Cells[i].Value.ToString() + "    ";
364                 }
365                 report.Show(selectedData, false);
366             }
367         }
368     }
```

품질 현황

PageNO	Speed	Length	RealPower	SetFrequency	SetDuty	SetPower	GateOnTime
Working Time							
22	250	241,2	1699	1000	100	82	1154
							2022-01-08

문제점 및 개선 방향

```
public void Show(string selectedData, bool append)
{
    if (append)
        richTextBox2.Text += selectedData;
    else
        richTextBox2.Text = selectedData;
    Show();
}
```

보고서 창이 있을때와
없을때 그리드뷰의 셀
클릭시 불러오는 데이터를
다르게 함

2) 프로젝트 수행 절차

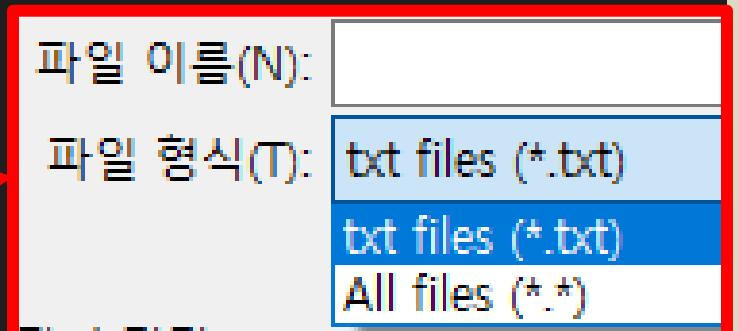
```
===== 파일 저장 =====
참조 1개
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        Stream myStream;
        SaveFileDialog saveFileDialog1 = new SaveFileDialog();

        saveFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
        saveFileDialog1.FilterIndex = 1;
        saveFileDialog1.RestoreDirectory = true;

        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {

            String txtSave = textBox1.Text + " 작성일 : " + label8.Text + " 작성자 : " + textBox2.Text + " 개요" + 
                richTextBox1.Text + " 품질 현황" + richTextBox2.Text + " 문제점 및 개선 방향" + 
                richTextBox3.Text + " 조치 및 결과" + richTextBox4.Text;

            this.Text = saveFileDialog1.FileName;
            using (StreamWriter sw = new StreamWriter(saveFileDialog1.FileName))
            {
                sw.WriteLine(txtSave);
            }
        }
    catch (Exception ex)
    {
        MessageBox.Show("오류 발생: " + ex.Message);
    }
    finally
    {
        Close();
    }
}
```



필터를 사용하여 파일형식 지정

저장 경로는 이전에 쓴 경로

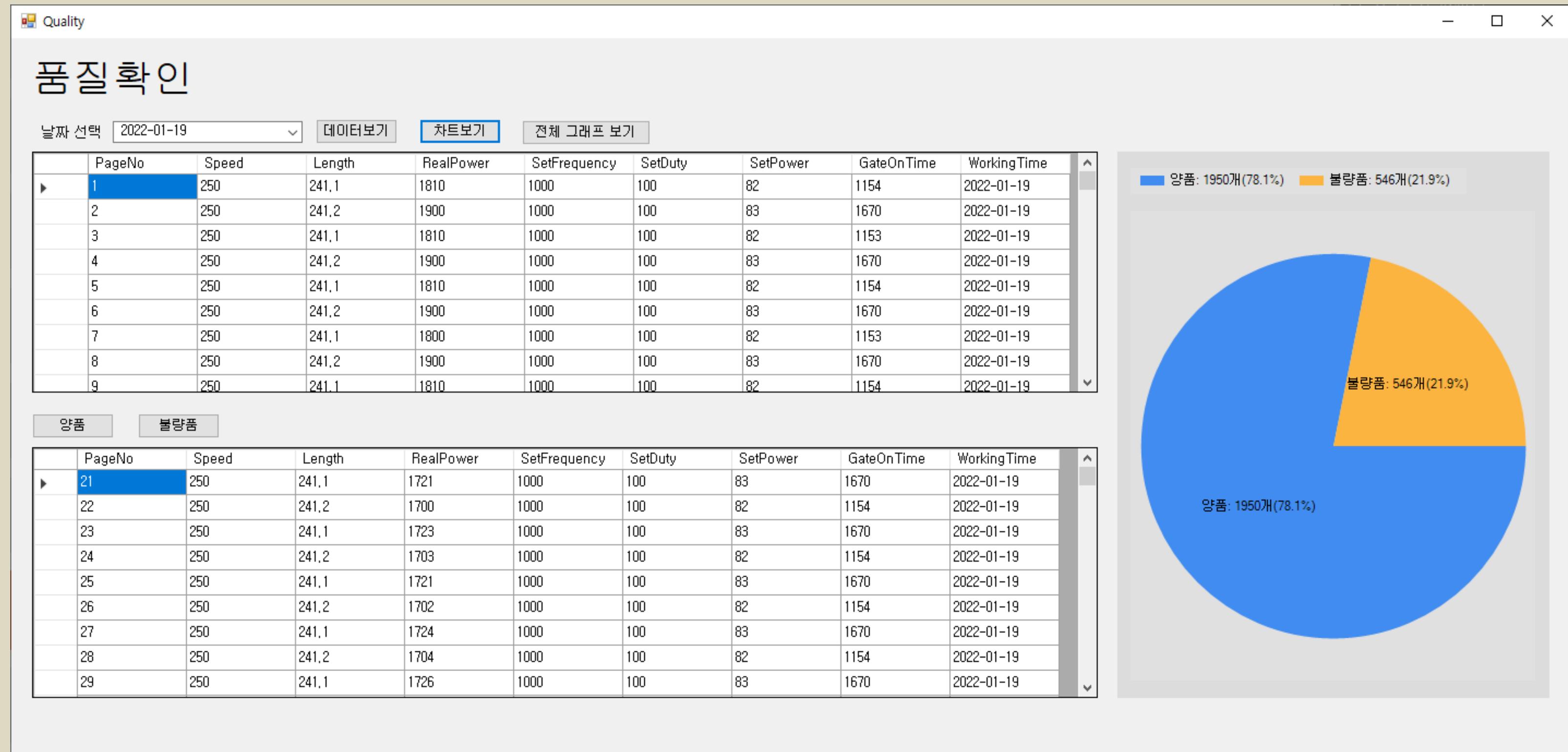
보고서에 저장된 Text를 저장

A screenshot of a Microsoft Word document titled 'test.txt - Windows 메모장'. The document contains the following text:

작성일 : 2024년 4월 11일 목요일 오전 11:57
작성자 :
개요
품질 현황
PageNO Speed Length RealPower SetFrequency SetDuty SetPower GateOnTime WorkingTime
2 250 241.2 1716 1000 100 83 1670 2022-02-18
문제점 및 개선 방향
조치 및 결과

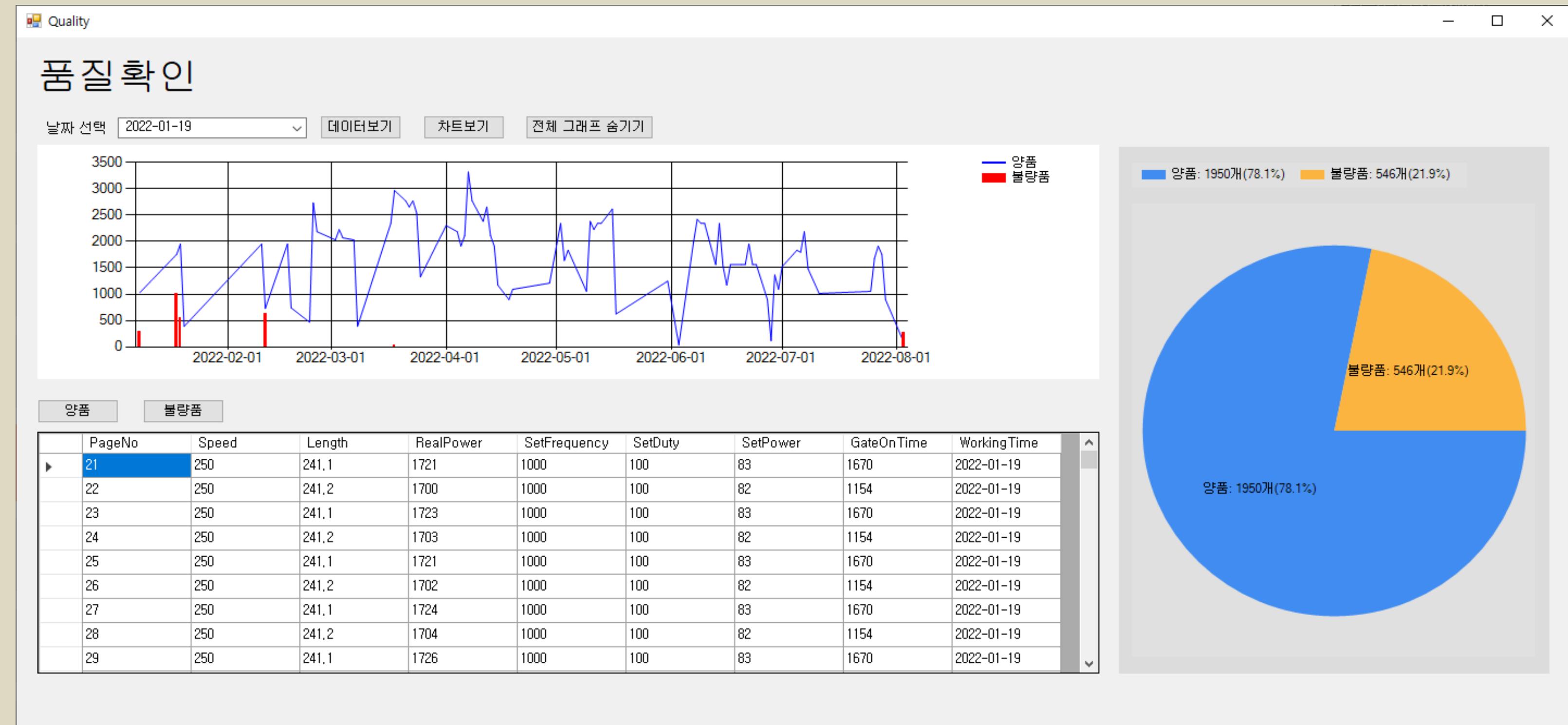
3) 프로젝트 수행 결과

실행 화면



3) 프로젝트 수행 결과

실행 화면



3) 프로젝트 수행 결과

실행 화면

Quality

품질 확인

날짜 선택: 2022-01-19 | 데이터보기 | 차트보기 | 전체 그래프 보기

PageNo	Speed	Length	RealPower	SetFrequency	SetDuty	SetPower	GateOnTime	WorkingTime
1	250	241.1	1810	1000	100	82	1154	2022-01-19
2	250	241.2	1900	1000	100	83	1670	2022-01-19
3	250	241.1	1810	1000	100	82	1153	2022-01-19
4	250	241.2	1900	1000	100	83	1670	2022-01-19
5	250	241.1	1810	1000	100	82	1154	2022-01-19
6	250	241.2	1900	1000	100	83	1670	2022-01-19
7	250	241.1	1800	1000	100	82	1153	2022-01-19
8	250	241.2	1900	1000	100	83	1670	2022-01-19
9	250	241.1	1810	1000	100	82	1154	2022-01-19

양품 불량품

PageNo	Speed	Length	RealPower	SetFrequency	SetDuty	SetPower	GateOnTime	WorkingTime
21	250	241.1	1721	1000	100	83	1670	2022-01-19
22	250	241.2	1700	1000	100	82	1154	2022-01-19
23	250	241.1	1723	1000	100	83	1670	2022-01-19
24	250	241.2	1703	1000	100	82	1154	2022-01-19
25	250	241.1	1721	1000	100	83	1670	2022-01-19
26	250	241.2	1702	1000	100	82	1154	2022-01-19
27	250	241.1	1724	1000	100	83	1670	2022-01-19
28	250	241.2	1704	1000	100	82	1154	2022-01-19
29	250	241.1	1726	1000	100	83	1670	2022-01-19

셀을 클릭하면 클릭한
데이터 행의 값들이
품질 현황에 뜸

보고서

보고서 제목

작성일: 2024년 4월 11일 목요일 오전 11:46

작성자: []

개요

품질 현황

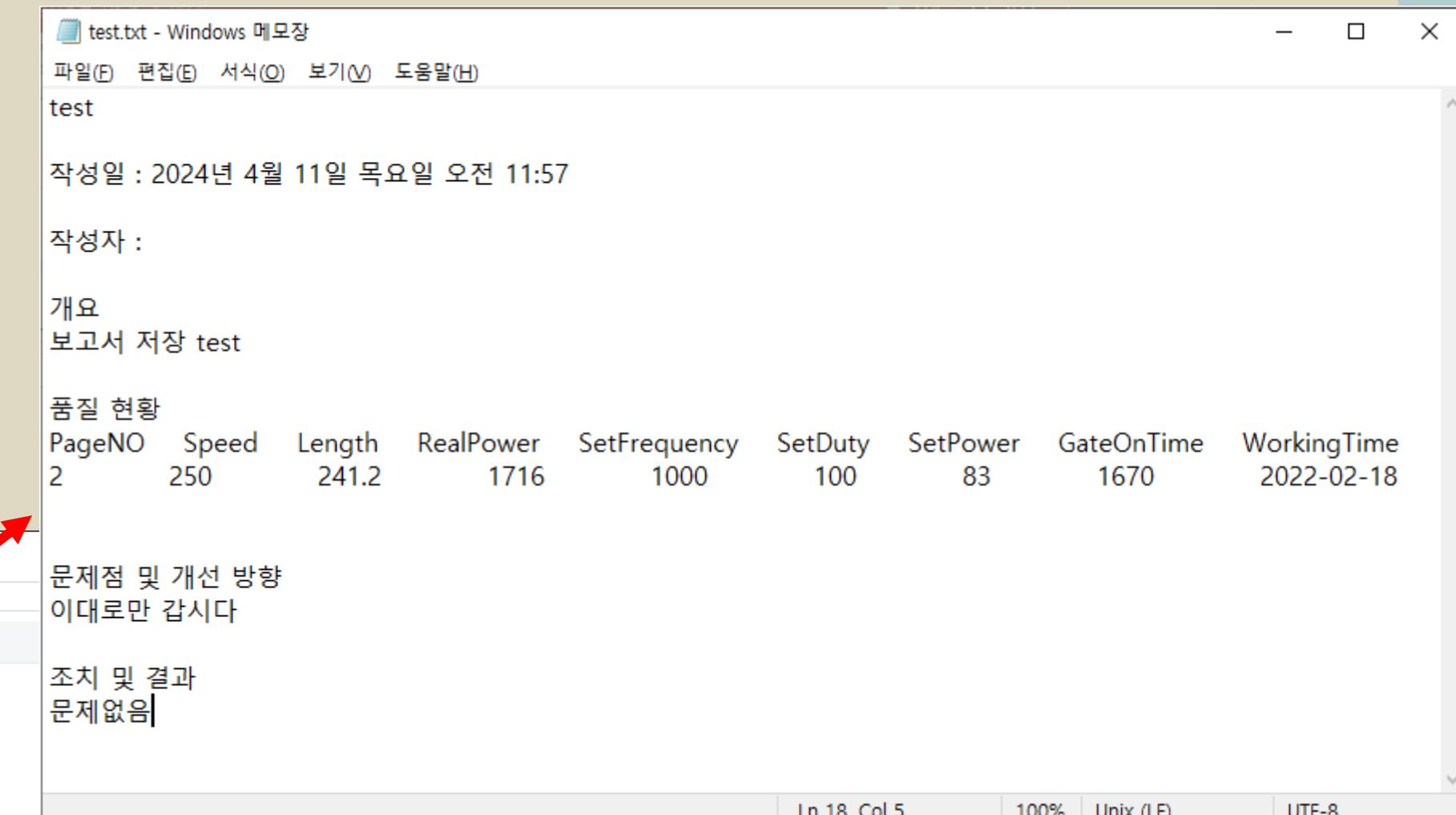
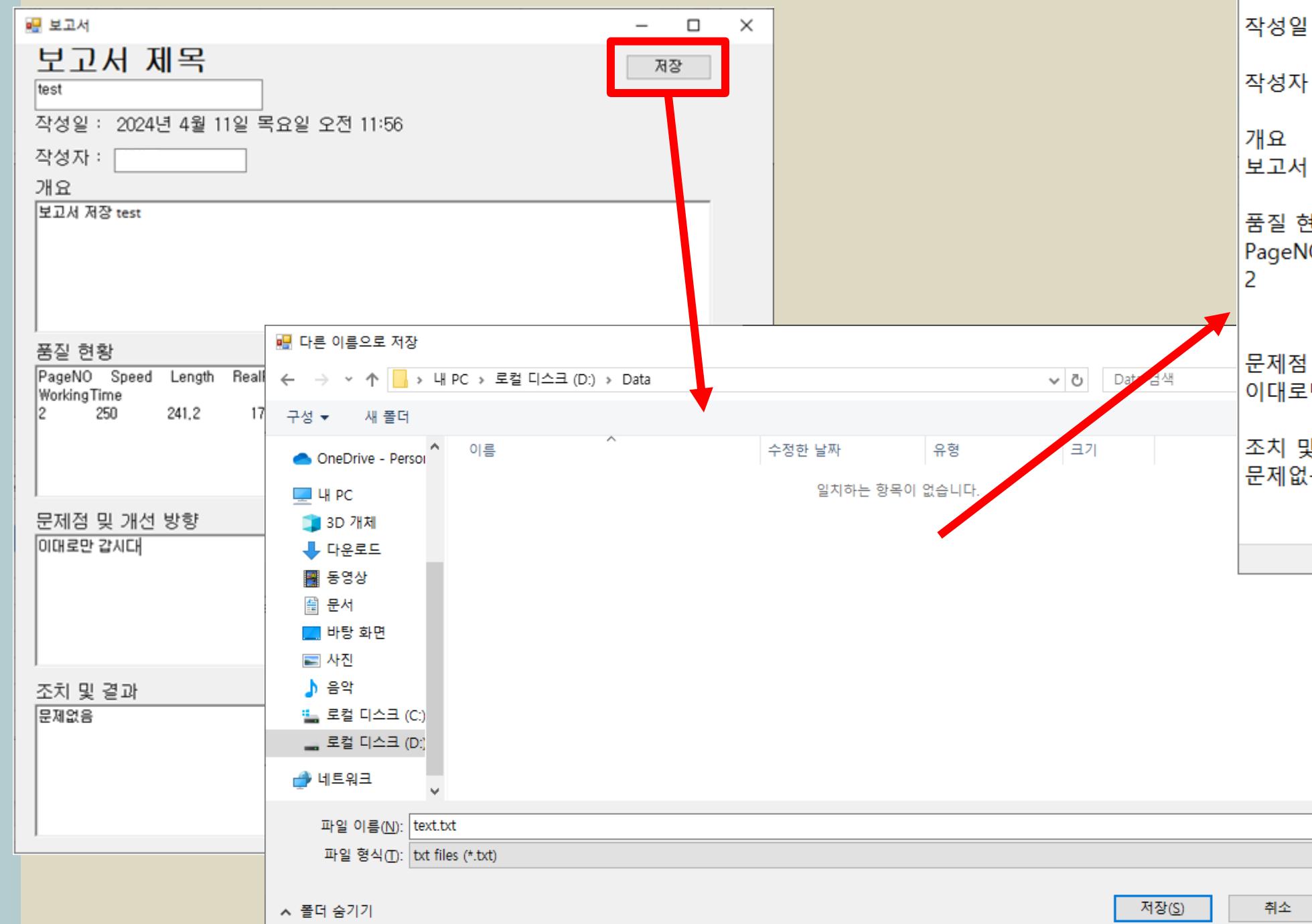
PageNO	Speed	Length	RealPower	SetFrequency	SetDuty	SetPower	GateOnTime	WorkingTime
22	250	241.2	1699	1000	100	82	1154	2022-01-08

문제점 및 개선 방향

조치 및 결과

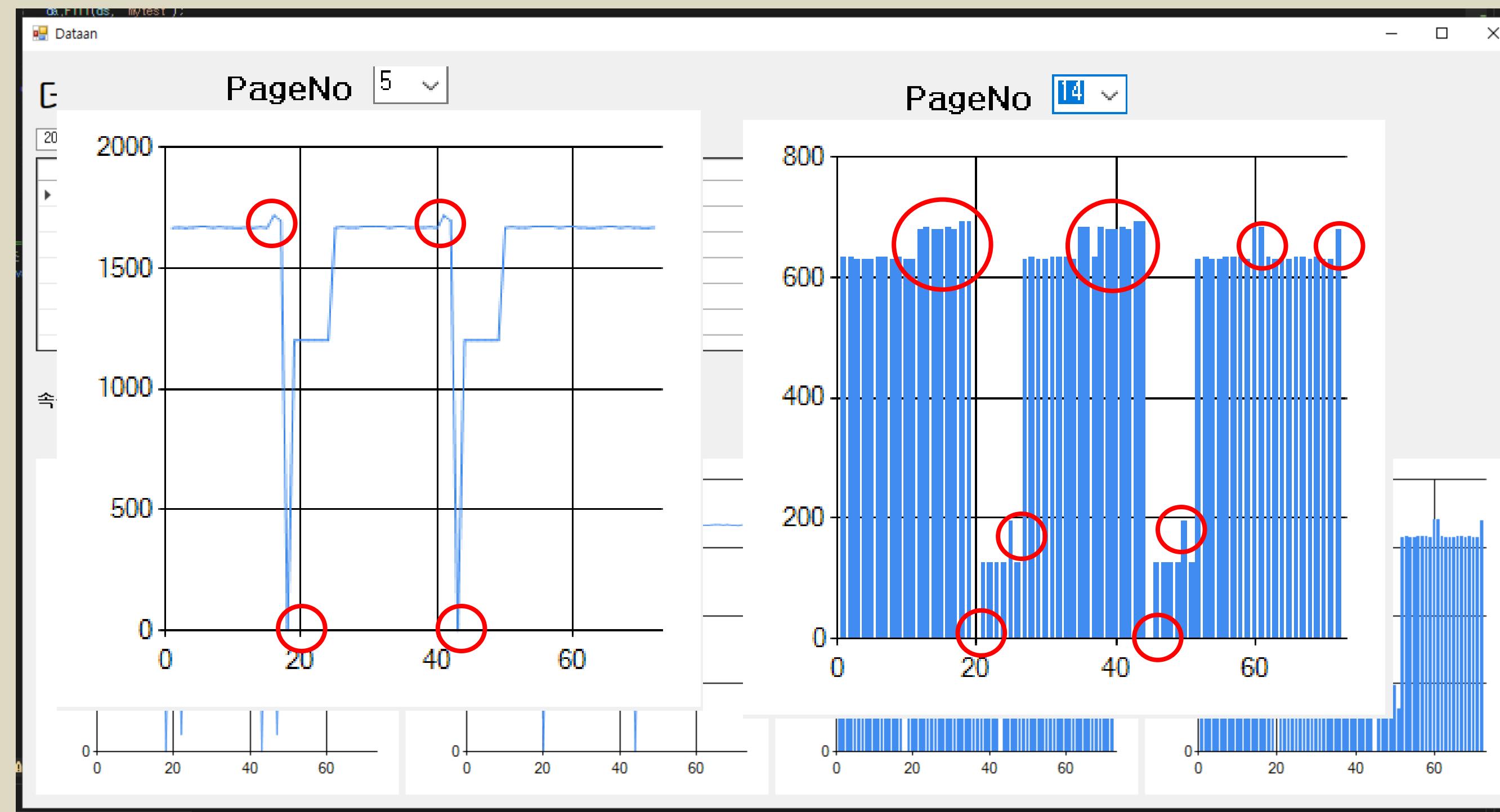
3) 프로젝트 수행 결과

실행 화면



3) 프로젝트 수행 결과

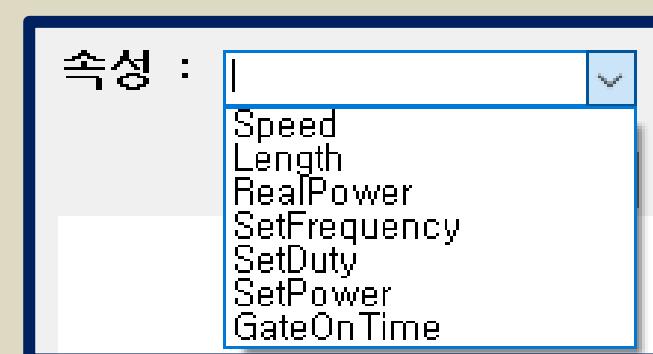
불량품이 생산될 때 제조 환경
데이터를 시각적으로 판단 할 수 있습니다.



3) 프로젝트 수행 결과

결과 분석

- 같은 날에 작업한 모든 공정 중 같은 용접 순서에서 문제가 발생 > 해당 공정에 대한 문제인식 및 조치 가능
- 전체적인 불량/양품의 비율을 쉽게 파악하여 품질 관리 가능
- 이상이 있는 항목에 대한 보고서를 작성하여 파일로 보관하여 문제가 있는 데이터들의 관리 가능
- 데이터 분석 품을 이용하여 이후 각 속성의 값을 다르게 주어 작업을 할 때에도 해당하는 값에 대해 분석이 가능하여 어떤 작업이 이뤄지는 품질 관리가 가능 Ex)



감사합니다

Thank you for Watching my presentation

데이터셋 출처 - 중소벤처기업부, Korea AI Manufacturing Platform(KAMP), 전자부품(배터리팩) 예지보전 AI 데이터셋,
스마트제조혁신추진단((주)인터엑스, 네스트필드(주)), 2022.12.23., www.kamp-ai.kr