

꿀 밀러

(쿠팡 **최저가** 탐색기)

권용규 류경문 권도균 권승연
장상은 주지환 이지훈 이승훈



프로그램 개요

똑같은 물건이라도 가격대는 천차만별

정보가 넘쳐나는 시대에 따라
동일한 물건이라도 판매자에 따라
가격이 변동되는 경우가 적지 않다.
이러한 상황에서 좀 더 쉽고
간편하게 최저가를 찾아내서
조금이라도 돈을 절약하는 방법은 없을까?
이와 같은 의문에서 프로그램은 시작된다.

02



1

상품 검색.

원하는 상품의 이름과 가격대의 범위를 지정하여 조건에 부합하는 데이터를 웹스크래핑을 통해 가져와서 그러한 결과를 위젯내에서 바로 보여주고 엑셀 형식으로 저장한다.

2

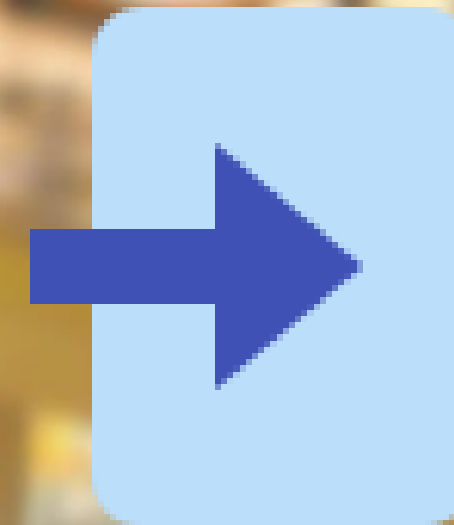
고객 관리.

검색을 원하는 고객의 정보를 관리하기 위해 이메일 아이디를 입력받아 저장한다. 이렇게 입력받은 고객의 이메일 정보는 엑셀 형식으로 저장되어 분류되어 추가, 삭제등의 관리가 가능하다.

3

메일 발송.

1단계에서 검색한 결과를 토대로 만들어진 데이터를 엑셀 파일을 포함하여 2단계에서 저장된 고객의 이메일 아이디를 통해 발송한다.



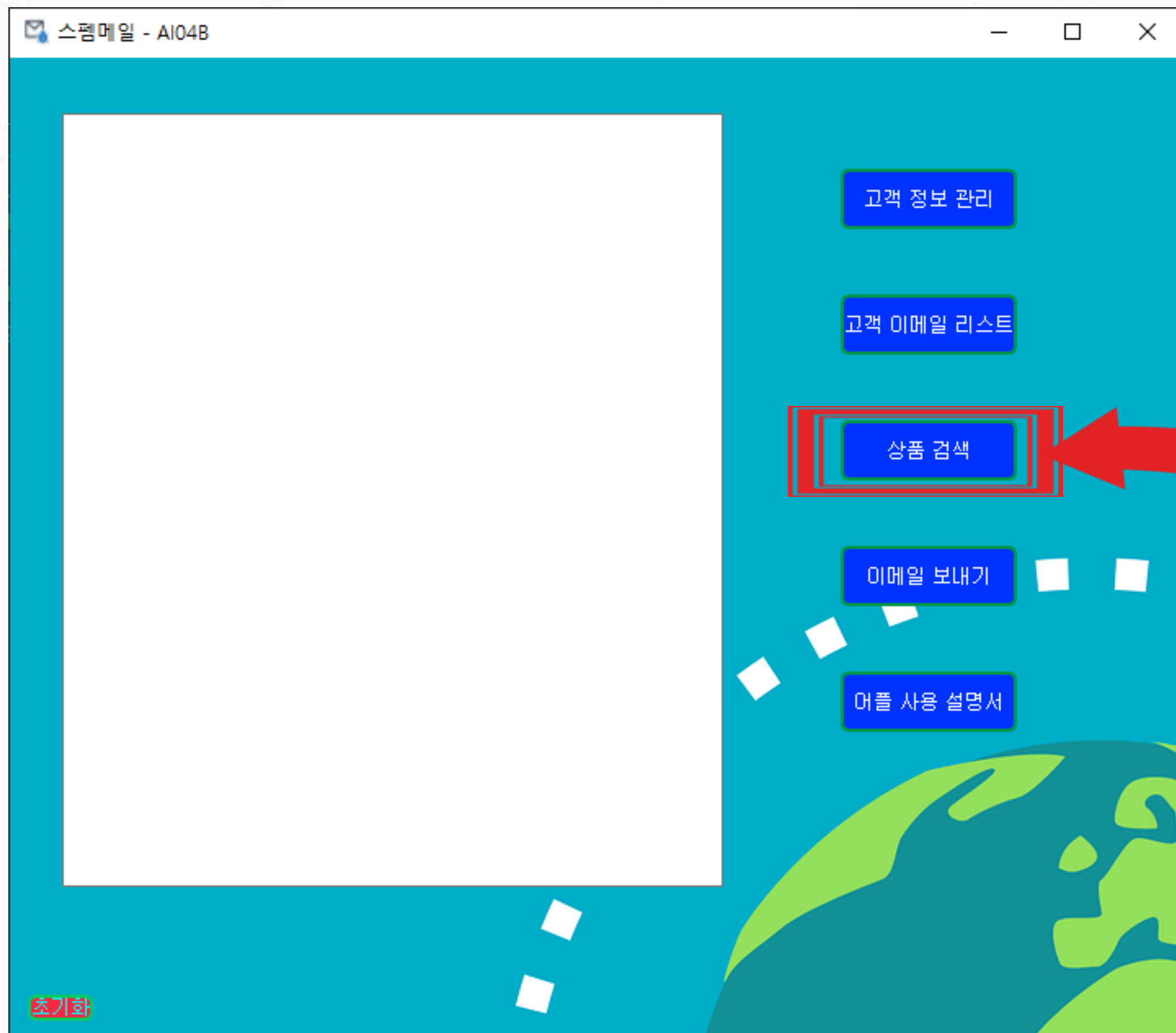
Welcome, Black Cows

이 사이트에 대한 권리는 제작자에게 있습니다.

Support by [경북산업직업전문학교](#)

주소: 대구 동구 동부로 121

전화번호: 0507-1482-7755



1단계 : 상품 검색

웹 스크래핑(Web Scrapping)

MainWindow를 열어
상품 검색 버튼을 누르면
검색창이 오픈이 되며
원하는 물품의
가격대와 범위를 지정해서
검색을 할 수 있다.

1단계 : 상품 검색

웹 스크래핑(Web Scrapping)

제품검색						
상품 이름		닌텐도	최저가격대	10000	최고가격대	200000
		검색		보기		
	제품명	가 격	평 점	평점 등록수	구매링크	제품 미리보기
1	닌텐도 스위치 슈퍼 마리...	56400	5	(1091)	바로가기	src = https://...
2	탐탁 NS Switch OLED ...	58000	평점 없음	평점 없음	바로가기	src = https://...
3	닌텐도 스위치 진 ...	60420	5	(18)	바로가기	src = https://...
4	닌텐도 스위치 마리오카...	60790	5	(2417)	바로가기	src = https://...
5	닌텐도스위치 링 피트 ...	78000	5	(2824)	바로가기	src = https://...

검색창이 나오면 상품 이름을 입력하고
원하는 가격의 범위를 입력한 뒤 검색을 누르고 보기를 누르면
결과가 표시되며 엑셀형태로 저장된다.

1단계 : 상품 검색

웹 스크래핑(Web Scraping)

```
# from PyQt5 import QtWidgets
import sys
import requests
import re
from PyQt5 import uic
from bs4 import BeautifulSoup
import math
from PyQt5.QtWidgets import *
from Main_Gui import Excelseat
# from alert_test import alert_test
```



```

class Dialog2(QDialog):
    def __init__(self, parent):
        super(Dialog2, self).__init__(parent)
        product_ui = './dialog2.ui'
        uic.loadUi(product_ui, self)
        self.search = None
        self.Lpricecut = None
        self.Upricecut = None
        self.Co = Excelseat()
        self.rowIndex = 0
        self.pushButton.clicked.connect(self.Search)
        self.pushButton_2.clicked.connect(self.connect)
        self.show()

    def alret1(self):
        QMessageBox.warning(self, '오류!', '이름을 확인해 주세요')

    def alret2(self):
        QMessageBox.warning(self, '오류!', '가격설정을 확인해 주세요')

```

1단계 : 상품 검색

웹 스크래핑(Web Scrapping)


```

def Search(self):
    try:
        self.search = self.lineEdit.text()
        if(self.search == ''):
            self.alret1()
        self.Lpricecut = int(self.lineEdit_2.text())
        self.Upricecut = int(self.lineEdit_3.text())
        if (self.Lpricecut < 0 or self.Upricecut < self.Lpricecut):
            self.alret2()
        url = "https://www.coupang.com/np/search?component=&q=" + self.search + "&channel=user"

        headers = {
            "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) "
                           "AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36"
        }
        res = requests.get(url, headers=headers)
        res.raise_for_status()
        soup = BeautifulSoup(res.text, "lxml")
        # print(res.text)

        items = soup.find_all("li", attrs={"class": re.compile("^search-product")})
        # price = soup.find_all("li", attrs={"class": re.compile("^search-product")})

```

1단계 : 상품 검색

웹 스크래핑(Web Scraping)

```

# 상품 불러오기
i = 0
information = []
for item in items:
    # 이름 값
    name = item.find("div", attrs={"class": "name"}).get_text()
    # 가격 값
    price = item.find("strong", attrs={"class": "price-value"}).get_text()
    price = int(price.replace(', ', ''))
    if (price > self.Upricecut or price < self.Lpricecut):
        continue
    # 평점 값
    rate = item.find("em", attrs={"class": "rating"})
    if rate:
        rate = rate.get_text()
        rate = float(rate)
        rate = math.ceil(int(rate))
    else:
        rate = "평점 없음"
    # 평점 수 값
    rate_cnt = item.find("span", attrs={"class": "rating-total-count"})
    if rate_cnt:
        rate_cnt = rate_cnt.get_text()
    else:
        rate_cnt = "평점 없음"
    # 링크 값
    link = item.find("a", attrs={"class": "search-product-link"})['href']
    link = "https://www.coupang.com" + link
    # 이미지 값
    imgsrc = item.find("img", attrs={"class": "search-product-wrap-img"})['src']
    imgsrc = "src = https:" + imgsrc
    # 정렬
    information.append([name, str(price), str(rate), rate_cnt, link, imgsrc])
information.sort(key=lambda x: int(x[1]))
self.Co.create(information)
except Exception as e:
    print(e)

```

1단계 : 상품 검색

웹 스크래핑(Web Scrapping)

1단계 : 상품 검색

웹 스크래핑(Web Scrapping)

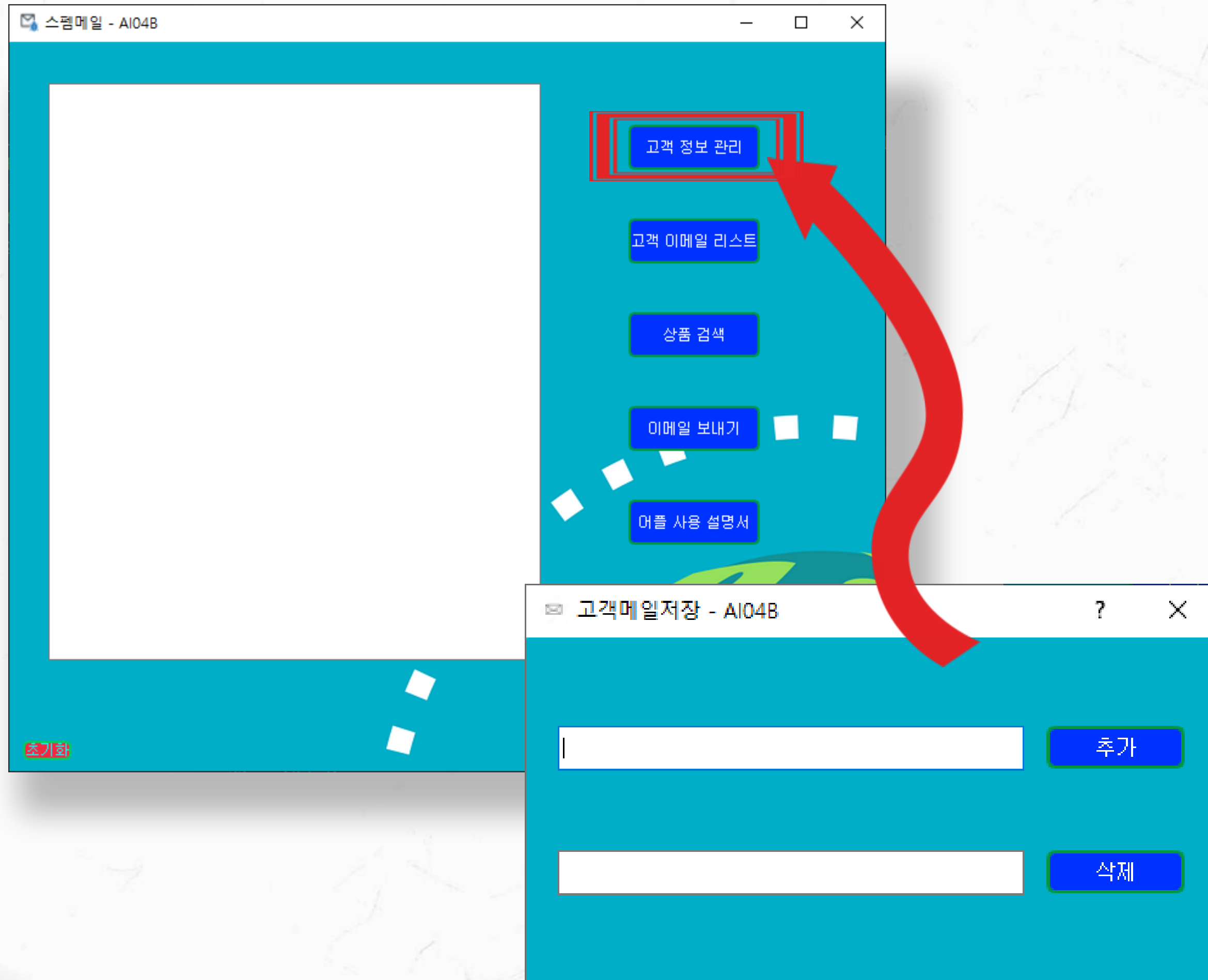
```
def connect(self):  
    self.tableWidget.clearContents()  
    rows = self.Co.loadrow()  
    self.rowIndex = 0  
    for row in rows:  
        self.tableWidget.setItem(self.rowIndex, 0, QTableWidgetItem(row[0]))  
        self.tableWidget.setItem(self.rowIndex, 1, QTableWidgetItem(row[1]))  
        self.tableWidget.setItem(self.rowIndex, 2, QTableWidgetItem(row[2]))  
        self.tableWidget.setItem(self.rowIndex, 3, QTableWidgetItem(row[3]))  
        self.tableWidget.setItem(self.rowIndex, 4, QTableWidgetItem(row[4]))  
        self.tableWidget.setItem(self.rowIndex, 5, QTableWidgetItem(row[5]))  
  
        self.rowIndex += 1
```

2단계 : 고객 관리

Client Management

'고객 정보 관리' 위젯 버튼으로
이메일 아이디를 입력받아
저장, 삭제 관리가 가능.

이렇게 저장되어진 고객의 정보는
'고객 이메일 리스트' 버튼을 통해 확인



2단계 : 고객 관리

Client Management

'고객 정보 관리' 위젯 버튼으로
이메일 아이디를 입력받아
엑셀 형식으로 저장, 삭제 관리가 가능.
이렇게 저장되어진 고객의 정보는
'고객 이메일 리스트' 버튼을 통해 확인

2단계 : 고객 관리

Client Management

```
from email.message import EmailMessage

class Customer_Email:

    def __init__(self):
        pass

    def Load_List(self):
        rows = []
        wb = load_workbook("Customer_Mail.xlsx")
        ws = wb.active
        for row in ws.iter_rows():
            rows.append(row[0].value)

        wb.close()
        return rows
```

2단계 : 고객 관리

Client Management

```
def RemoveEmail(self, email):  
    rows = []  
    count = 1  
    wb = load_workbook("Customer_Mail.xlsx")  
    ws = wb.active  
  
    for row in ws.iter_rows():  
        rows.append(row[0].value)  
  
    for e in rows:  
        if email != e:  
            count += 1  
        elif email == e:  
            ws.delete_rows(count)  
  
    wb.save("Customer_Mail.xlsx")  
    wb.close()
```

```
def SaveEmail(self, email):  
    rows = []  
    wb = load_workbook("Customer_Mail.xlsx")  
    ws = wb.active  
  
    for row in ws.iter_rows():  
        rows.append(row[0].value)  
    if len(rows) >= 1:  
        for i in range(0, len(rows)):  
            if rows[i] == email:  
                return  
    ws.append([email,1])  
    wb.save("Customer_Mail.xlsx")  
    wb.close()
```

2단계 : 고객 관리

Client Management

```
def CreateEmail_xl(self):
    wb = Workbook() # 새 워크북 생성
    ws = wb.active # 현재 활성화된 sheet 가져옴
    ws.append(['Email List', 'subscribe'])

    ws.title = "email" # sheet 의 이름을 변경
    wb.save("Customer_Mail.xlsx")
    wb.close()

def recive_mail(self):#류경문 - 이메일 받기
    wb=load_workbook("Customer_Mail.xlsx")
    ws=wb.active

    mailbox = MailBox("imap.gmail.com", 993)
    mailbox.login(EMAIL_ADDRESS, EMAIL_PASSWORD, initial_folder="INBOX")
    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
```



```

for msg in mailbox.fetch(('UNSEEN TEXT "AI04B"'), limit=5, reverse=False):

    if msg.from_ is not None: #이메일 내용에 등록 이란 단어가 포함되어있으면 엑셀에 1을 추가
        if "등록" in msg.text:

            ws.append([msg.from_, 1])
            print("새로 등록한 고객", msg.from_)

        if "해지" in msg.text: #내용에 해지란 단어가 있으면 0을 추가

            ws.append([msg.from_, 0])
            print("해지 고객", msg.from_)

            cu_address=msg.from_ #해지 안내 메일 발송
            msg = EmailMessage()
            msg["Subject"] = cu_address + "님 해지완료. "
            msg["From"] = EMAIL_ADDRESS
            msg["TO"] = cu_address
            msg.set_content(cu_address+ "고객님 정상 해지 완료! \n감사합니다.")

            with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
                smtp.ehlo()
                smtp.starttls()
                smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)

                smtp.send_message(msg)

        else:
            self.del_list() #이메일 중복 삭제, 해지 요청한사람 삭제 작업으로 넘김

```

2단계 : 고객 관리

Client Management

2단계 : 고객 관리

Client Management

```
wb.save("Customer_Mail.xlsx")
```

```
wb.close()
```

```
self.del_list()#이메일 중복 삭제, 해지 요청한사람 삭제 작업으로 넘김
```

```
def del_list(self):#류경문 - 중복제거 + 해지 고객 삭제
```

```
del_same = pd.read_excel("Customer_Mail.xlsx", sheet_name="email")
```

```
df = del_same.drop_duplicates(["Email List"], keep="last")
```

```
df.to_excel("Customer_Mail.xlsx", sheet_name="email", index=False)
```

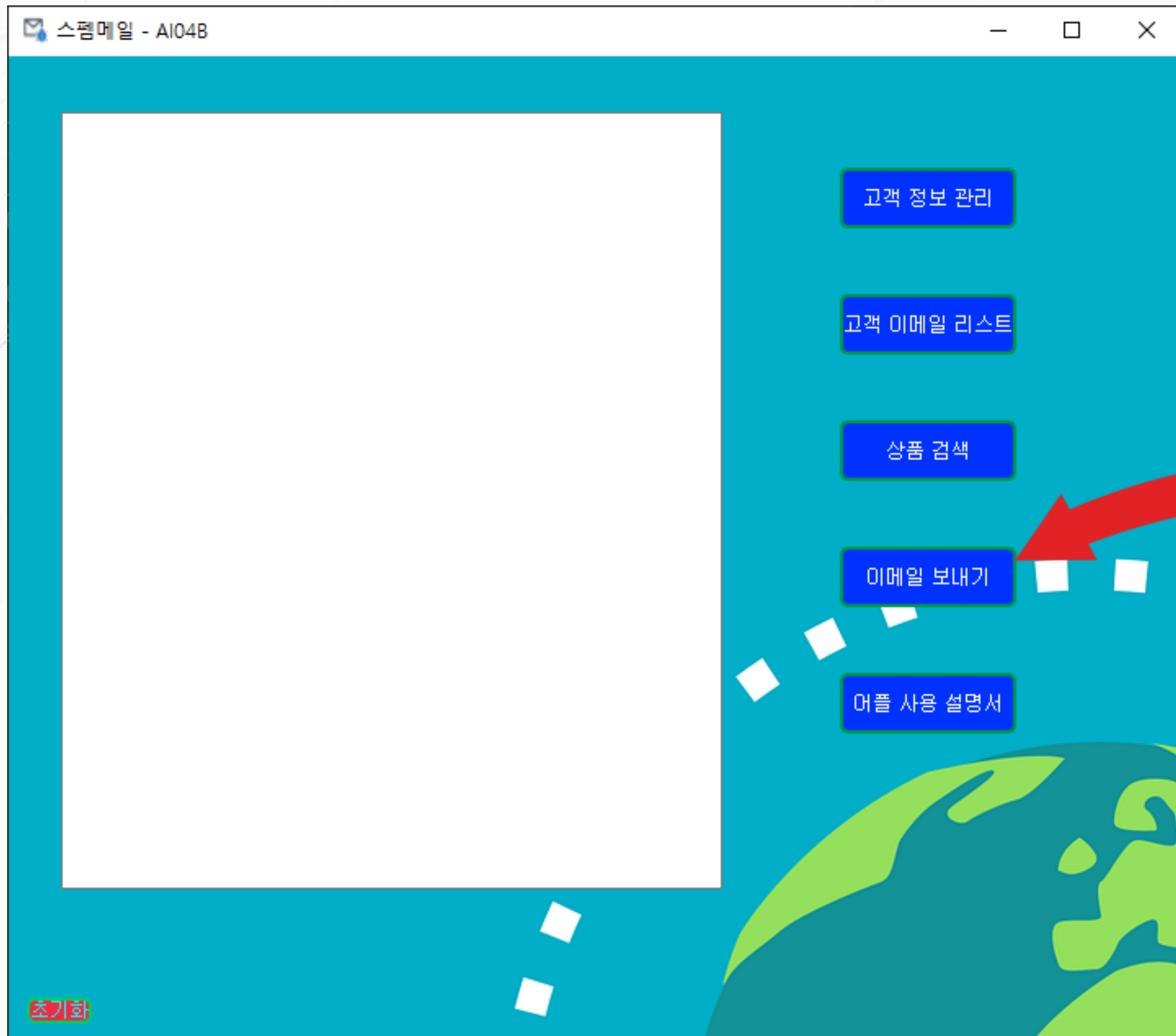
```
del_same = pd.read_excel("Customer_Mail.xlsx", sheet_name="email")
```

```
df1 = del_same[del_same['subscribe'] == 1] #B 열에 숫자 1이 없는 행은 전부 삭제
```

```
#해지 요청한 사용자는 0이 들어가서 이단계에서 삭제됨
```

```
df1.to_excel("Customer_Mail.xlsx", sheet_name="email", index=False)
```

```
print("중복, 해지고객 관리 완료")
```



3단계 : 메일 발송

Send Mail

'이메일 보내기 버튼'을 누르면
2단계에서 입력받아
엑셀에 저장되어 있는 이메일 정보로
1단계에서 크롤링한 상품의 검색 결과를
매크로 메일을 발송

```
def send_mail(self): #류경문 - 이메일 보내기
```

```
wb = load_workbook('Customer_Mail.xlsx') #현재 저장된 고객 명단  
ws = wb.active
```

```
wb1 = load_workbook("검색결과.xlsx", data_only=True) #쿠팡 검색 결과  
ws1 = wb1.active
```

```
for temp1 in ws.iter_rows(min_row=2): #고객 명단 불러오기  
    customer_addr = temp1[0].value
```

```
for i in ws1.iter_rows(min_row=2): #검색 결과 불러오기  
    img_tag = i[5].value
```

```
s = smtplib.SMTP('smtp.gmail.com', 587)  
s.starttls()  
s.login(EMAIL_ADDRESS, EMAIL_PASSWORD)  
msg = MIMEMultipart('alternative')  
msg['Subject'] = customer_addr+"고객님만을 위한 특가!"  
msg['From'] = EMAIL_ADDRESS  
msg['To'] = customer_addr
```

3단계 : 메일 발송

Send Mail

3단계 : 메일 발송

Send Mail

```
body = '''\n<h2>더이상 연락을 받기 싫으신분은 답장하기를 눌러서 보내기를 누르시면 해지 처리 해 드립니다. \n
      AI04B</h2>\n
      <a href="'''+i[4].value+'''">쿠팡 바로가기</a>
'''+ img_tag + '''

    ...

#body 안에 HTML 작성가능(img_tag에는 웹스크래핑으로 얻은 html 태그가 담겨져있음)

body = (body.replace('[', '').replace(',', ','))
text = MIMEText(body, 'html', 'utf-8')
msg.attach(text)

wb.close()
s.sendmail(EMAIL_ADDRESS, customer_addr, msg.as_string())
s.quit()
print(customer_addr + "님에게 발송 완료")
wb1.close()
wb.close()
```

3단계 : 메일 발송

Send Mail

☆ newruh@naver.com고객님만을 위한 특가! 

보낸사람  <newruh2@gmail.com>


받는사람 <newruh@naver.com>

더이상 메일을 받기 싫으신 고객님께서는, 답장하기를 눌러서 보내기를 누르시면 해지 처리 해 드립니다. - AI04B -

[쿠팡 바로가기](#)




Apple 2020 맥북 에어 13, 스페이스 그레이, M1, 256GB, 8GB, MAC OS, MGN63KH/A


12% 4,290,000-1,134,200원  로켓와우

내일(금) 12/10 도착 보장

새 상품, 중고 (3) 최저1,100,170원

5.0 (5431)

 최대 5% 카드 즉시할인

 최대 50,000원 적립





Q&A

THANK
YOU