



# 간편 장부 프로그램

OCR과 인공지능 기술로 회계 관리를 자동화하는 솔루션



영수증 자동 인식



AI 기반 분석



데이터 시각화

김동선 | 김미소



## 01

## 개발 진행 상황

## 간편 장부 프로그램

이름(아이디)

전화번호(비밀번호)

영수증 이미지 업로드

 선택된 파일 없음

## 간편 장부 프로그램

이름(아이디)

전화번호(비밀번호)

영수증 이미지 업로드

 example(3).jpg

농협			
주소: 경기 의정부시      전화: 031-***-****			
대표: 최영*      사업자번호: 127-82-****			
홈페이지: http://www.nonghyup.com/			
영수증 이미지합치 교환/환불 불가(30일내)			
교환/환불 구매점에서 가능(결제카드지급)			
-----			
김갑순    2015-11-03 16:31:53    0002-00085			
상품(코드)	단가	수량	금액
001 P국산농우유 900ML			[2,150]
*8801104210645	1,350	1	1,350
002 P왕파			
*231973	3,300	1	3,300
003 P우유			
*231913	500	1	500
004 P꽃얼			
*231308	750	1	750
005 P하선정 바로먹기올온장아찌 150g			
8801007265889	1,380	1	1,380
006 P브로커리			
*232285	1,280	1	1,280
관 대 총 액:			8,560
◆ 받 을 금 액:			8,560
인 용 액:			8,560
>> 부가세면세물품가액:			7,180
>> 부가세과세물품가액:			1,255
부 가 세:			125
-----			
바코드알 * 면세, # 영세, 상품명 P포인트			
회원: 2010190034*** 박*분 님			
오스 구매포인트: 40			


## 02

## Maria DB 연동

```
-- 👤 사용자 테이블
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  phone VARCHAR(20), receipts
  gender ENUM('M', 'F')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 📄 영수증 테이블 (사용자 FK)
CREATE TABLE receipts (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT,
  file_name VARCHAR(255),
  raw_text TEXT,
  summary_date DATETIME,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 🍽 메뉴 항목 테이블 (영수증 FK)
CREATE TABLE receipt_items (
  id INT AUTO_INCREMENT PRIMARY KEY,
  receipt_id INT,
  menu_name VARCHAR(255),
  price INT,
  FOREIGN KEY (receipt_id) REFERENCES receipts(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

#  MariaDB 연결 정보

```
db_url = "mysql+pymysql://root:wsuser!@localhost:3336/ledger"
engine = create_engine(db_url, echo=True)
```

 users 테이블

- 사용자 기본 정보 관리
- 사용자 ID(PK), 이름, 전화번호
- 필드 영수증 데이터와의 1:N 관계 형성

 receipts 테이블

- 영수증 메타데이터 저장
- 영수증 ID(PK), 사용자 ID(FK), 날짜, 상점
- 명 등 개별 상품 항목과 1:N 관계 형성

 receipts\_item 테이블

- 영수증 내 개별 상품 정보
- 항목 ID(PK), 영수증 ID(FK), 상품명, 가격,
- 수량 상세 분석 및 통계 기능 지원

## 03

## DB + Flask 연동

```
# ✅ 5) DB 저장: users → receipts → receipt_items
with engine.begin() as conn:
    # 5-1) 사용자 존재 여부 확인
    select_user = text("""
        SELECT id FROM users WHERE name = :name AND phone = :phone
    """)
    user_result = conn.execute(select_user, {
        "name": user_name,
        "phone": user_phone
    }).fetchone()

    if user_result:
        user_id = user_result[0]
        print(f"✅ 기존 사용자 ID: {user_id}")
    else:
        insert_user = text("""
            INSERT INTO users (name, phone, gender)
            VALUES (:name, :phone, :gender)
        """)
        result = conn.execute(insert_user, {
            "name": user_name,
            "phone": user_phone,
            "gender": None # 성별은 입력폼 없으면 NULL
        })
        user_id = result.lastrowid
        print(f"✅ 새 사용자 ID 생성: {user_id}")
```

```
# 5-2) receipts 저장
insert_receipt = text("""
    INSERT INTO receipts (user_id, file_name, raw_text, summary_date)
    VALUES (:user_id, :file_name, :raw_text, :summary_date)
""")
result = conn.execute(insert_receipt, {
    "user_id": user_id,
    "file_name": file.filename,
    "raw_text": ocr_text,
    "summary_date": date_str
})
receipt_id = result.lastrowid
print(f"✅ 영수증 저장 완료: receipt_id={receipt_id}")

# 5-3) receipt_items 저장
for menu in menus:
    insert_item = text("""
        INSERT INTO receipt_items (receipt_id, menu_name, price)
        VALUES (:receipt_id, :menu_name, :price)
    """)
    conn.execute(insert_item, {
        "receipt_id": receipt_id,
        "menu_name": menu["name"],
        "price": menu["price"]
    })

print(f"✅ 전체 DB 저장 완료")
```

## 04

## DB 화면

#	id	receipt_id	menu_name	price
1	1	1	크림 파스타	540,900,123
2	2	1	닭 한 마리 쌀국수	7,000
3	3	2	크림 파스타	5,409
4	4	2	닭 한 마리 쌀국수	7,000
5	5	2	카드 결제 일시불	22,000
6	6	3	크림 파스타	54,090
7	7	3	닭 한 마리 쌀국수	7,000
8	8	3	카드 결제 금액	22,000

#	id	name	phone	gender
1	1	김미소	01023555103	(NULL)
2	2	김동선	01023456789	(NULL)
3	3	홍길동	01011112222	(NULL)

#	id	user_id	file_name	raw_text	summary_date	created_at
1	1	(NULL)	example.jpg	「6-11주문」2024-09-12 18:59크림 파스타닭 한 마 리...	2024-09-12 18:59:00	2025-07-15 16:26:07
2	2	1	example.jpg	「6-11주문」2024-09-12 18:59크림 파스타닭 한 마 리...	2024-09-12 18:59:00	2025-07-15 16:56:55
3	3	1	example.jpg	「6-11주문」2024-09-12 18:59크림 파스타닭 한 마 리...	2024-09-12 18:59:00	2025-07-16 12:07:01

## 05

## SQL

```
1 SELECT *  
2 FROM receipts  
3 WHERE user_id = 1;
```

receipts (2r × 6c)							
#	id	user_id	file_name	raw_text	summary_date	created_at	
1	2	1	example.jpg	「6-11주문」2024-09-12 18:59크림 파스 타닭 한 마 리...	2024-09-12 18:59:00	2025-07-15 16:56:55	
2	3	1	example.jpg	「6-11주문」2024-09-12 18:59크림 파스 타닭 한 마 리...	2024-09-12 18:59:00	2025-07-16 12:07:01	

## 06

# 향후 개발 계획



### Spring 연동

- Spring Framework 기반 백엔드 확장 RESTful
- API 설계 및 구현
- Spring Security를 통한 인증 시스템 구축



### 파일 다운로드 기능

- 영수증 분석 데이터 엑셀 추출 기능
- JSON 형식 데이터 추출 API 개발 PDF
- 리포트 생성 및 저장 기능



### 질의응답



문의 및 피드백

[es4135@naver.com](mailto:es4135@naver.com)