

YMV Step 6 개발 현황 보고서 - 다음 개발자용

🔔 현재 상태 요약

프로젝트: YMV 관리 프로그램 v4.0
현재 위치: D:\ymv-business-system
진행 단계: Step 6 (견적서 관리 시스템) 중단됨
마지막 작업: main.py 업데이트 및 모듈 import 오류 발생

📋 미완성 작업 목록

Step 6-1: 데이터베이스 스키마 생성 (미실행)

상태: ❌ 실행되지 않음
파일: quotation_schema.sql 생성됨, 하지만 Supabase에서 실행 안됨
필요 작업: Supabase SQL Editor에서 스키마 실행

Step 6-2: quotation_management.py 컴포넌트 (미생성)

상태: ❌ 파일 생성 안됨
오류: ModuleNotFoundError: 'components.quotation_management'
필요 작업: 컴포넌트 파일 생성

Step 6-3: main.py 업데이트 (부분 완료)

상태: ⚠️ 업데이트됨, 하지만 import 오류로 실행 불가
문제: quotation_management 모듈 참조하지만 파일 없음

🗄️ 데이터베이스 현재 상태

기존 테이블 (정상 작동)

sql
✅ employees - 직원 관리 (로그인 작동)
✅ expenses - 지출 요청서 (business_purpose 컬럼 이슈 해결됨)
✅ customers - 고객 정보 (최소 1건 필요)
✅ products - 제품 정보 (최소 1건 필요)
✅ company_info - 회사 정보 (1건 필요)
✅ product_codes - 제품 코드 관리 (작동)
✅ purchases - 구매품 관리 (작동)

미생성 테이블 (Step 6용)

sql

- ✗ quotations - 견적서 마스터
- ✗ quotation_items - 견적서 상세
- ✗ inventory - 재고 관리
- ✗ quotation_history - 진행상황 추적
- ✗ quote_number_sequence - 번호 관리

해결된 데이터베이스 이슈

- expenses.business_purpose 컬럼: NOT NULL 제약조건 제거됨
- 컬럼명 통일: main.py에서 `purpose` → `business_purpose`로 수정됨

현재 파일 구조

```
D:\ymv-business-system/
├── app/
│   ├── main.py ✓ (업데이트됨, import 오류 있음)
│   └── components/
│       ├── __init__.py ✓
│       ├── code_management.py ✓ (정상 작동)
│       ├── multilingual_input.py ✓ (정상 작동)
│       └── quotation_management.py ✗ (미생성)
├── database/
│   ├── upgrade_v4.sql ✓ (실행됨)
│   ├── additional_schema_fix.sql ✓ (실행됨)
│   ├── fix_expenses_schema_safe.sql ✓ (실행됨)
│   └── quotation_schema.sql ✗ (생성됨, 미실행)
├── requirements.txt ✓
├── .streamlit/config.toml ✓
└── .env ✓ (Supabase 연결)
```

🔧 발생한 오류 및 해결책

1. ModuleNotFoundError

오류: `ModuleNotFoundError: 'components.quotation_management'`

원인: quotation_management.py 파일이 생성되지 않음

해결책: 컴포넌트 파일 생성 후 import 활성화

2. 데이터베이스 컬럼명 불일치

오류: `business_purpose column violates not-null constraint`

해결: `ALTER TABLE expenses ALTER COLUMN business_purpose DROP NOT NULL;`

현재 상태: 해결됨, main.py에서 business_purpose 사용

3. 긴 대화 맥락 손실

문제: 채팅 길이로 인한 컨텍스트 제한

영향: Step 6 중간에 중단

즉시 해야 할 작업

우선순위 1: 기본 실행 가능하게 만들기

1. main.py 임시 수정

```
python

# quotation_management import 주석 처리
# from components.quotation_management import show_quotation_management as quotation_component

# 임시 견적서 관리 함수 생성
def show_quotation_management():
    st.info("견적서 관리 시스템 준비 중...")
```

2. 기본 실행 테스트

```
bash

streamlit run app/main.py
```

우선순위 2: Step 6 완성

1. 데이터베이스 스키마 실행

- Supabase → SQL Editor
- quotation_schema.sql 전체 실행

2. quotation_management.py 생성

- 위치: `app/components/quotation_management.py`
- 완전한 CRUD 기능 포함

3. main.py import 복원

- quotation_management import 활성화
- 임시 함수 제거

컴포넌트 개발 주의사항

1. 파일 구조 준수

```
python
```

```
# components/quotation_management.py
def show_quotation_management(conn):
    """메인 함수"""
    pass

def generate_unique_key():
    """유니크 키 생성"""
    pass

# 기타 보조 함수들
```

2. 데이터베이스 연결 형식

```
python

# main.py에서 호출 방식
from components.quotation_management import show_quotation_management as quotation_component

# 사용법
quotation_component(st.connection("supabase"))
```

3. Streamlit 위젯 키 중복 방지

```
python

def generate_unique_key():
    return f"{int(time.time())}_{uuid.uuid4().hex[:8]}"

# 모든 위젯에 key 사용
st.text_input("label", key=generate_unique_key())
```

현재 작동하는 기능들

완전 작동

- ☒ 로그인/로그아웃 (Master/1023)
- ☒ 대시보드
- ☒ 지출 요청서 관리 (100% 완성)
- ☒ 코드 관리 시스템
- ☒ 구매품 관리 (부분)

메시지만 표시

- ☒ 견적서 관리 (Step 6에서 구현됩니다)

- ❌ 고객 관리 (Step 7에서 구현됩니다)
- ❌ 제품 관리 (Step 8에서 구현됩니다)
- ❌ 직원 관리 (향후 구현됩니다)

🔒 권한 시스템 현황

현재 구현 범위:

- 로그인/로그아웃 기본 기능
- employees.is_admin 필드 활용
- 지출 요청서에서만 권한 구분 적용

미구현 범위:

- 세분화된 권한 레벨
- 기능별 접근 권한 매트릭스

📄 견적서 시스템 설계 (Step 6)

데이터베이스 스키마

```
sql
-- 5개 테이블 필요
quotations (마스터)
quotation_items (상세)
inventory (재고)
quotation_history (이력)
quote_number_sequence (번호)
```

기능 요구사항

- 영어 기반 견적서 (베트남어 제품명 병기)
- 자동 견적서 번호 생성 (YMV-Q-2024-0001)
- 고객/제품 데이터 연동
- 프린트 최적화
- 통계 대시보드

견적서 포맷

고객정보(좌) | 회사정보(우)
견적서 정보 (중앙)
제품 목록 (테이블)
소계/VAT/합계

⚠ 다음 개발 시 주의사항

1. 개발 순서 엄수

1. 데이터베이스 스키마 먼저 실행
2. 컴포넌트 파일 생성
3. main.py import 수정
4. 테스트 실행

2. 데이터 의존성 확인

- customers 테이블: 최소 1개 고객 데이터
- products 테이블: 최소 1개 제품 데이터
- company_info 테이블: 회사 정보

3. 컴포넌트 독립성 유지

- 각 컴포넌트는 독립적으로 작동
- main.py에서 연결 객체만 전달
- 위젯 키 중복 방지 필수

4. 오류 처리 강화

- 데이터베이스 연결 실패 시 대응
- 필수 데이터 부족 시 안내 메시지
- import 오류 시 fallback 제공

🚀 Step 6 완료 후 로드맵

Step 7: 고객 관리 시스템 완성

Step 8: 제품 관리 시스템 완성

Step 9: 재고 관리 시스템

Step 10: 통합 테스트 및 배포

📞 개발 지원 정보

테스트 계정: Master / 1023

배포 URL: <https://ymv-business-system.streamlit.app>

로컬 실행: `streamlit run app/main.py`

Supabase: 연결 설정 완료

다음 개발자에게: 이 문서의 "즉시 해야 할 작업" 섹션부터 시작하세요. 기본 실행 가능 상태로 만든 후 Step 6을 완성하면 됩니다.