

# 이루다 로드맵 생성 엔진 상세 분석

## 1. 로드맵 생성 프로세스 전체 흐름

데이터 수집 → 분석 → 생성 → 저장

사용자 프로필 조회 → 규칙 기반 매칭 → 로드맵 생성 → DB 저장 → 상세 계획 확장

## 2. 사용자 프로필 데이터 구조

회원가입 시 수집되는 데이터

```
python

# 기본 정보
name, email, password

# 핵심 프로필 정보
housing_status = "자립준비청소년" | "원룸 거주" | "쉐어하우스" | "기타"
income_level = "50만원 이하" | "50-100만원" | "100-150만원" | "150만원 이상"
support_needs = ["주거지원", "경제지원", "교육지원", "취업지원", "심리지원", "의료지원"]
```

## 프로필 데이터 활용 방식

```
python

# generate_roadmap() 함수에서 사용자 프로필 조회
cursor.execute("""
    SELECT housing_status, income_level, support_needs
    FROM user_profiles WHERE user_id = ?
""", (current_user.id,))
profile_data = cursor.fetchone()

support_needs = json.loads(profile_data[2]) if profile_data[2] else []
```

## 3. 현재 로드맵 생성 엔진 (규칙 기반)

### 3.1 하드코딩된 템플릿 구조

```
python
```

```
roadmap = {
    "title": f"{current_user.name}님의 맞춤형 자립 로드맵",
    "description": "체계적인 자립을 위한 단계별 계획입니다.",
    "priority_areas": ["주거 안정", "경제적 자립", "사회적 네트워크 구축"],
    "timeline": {
        "1개월": ["주거급여 신청", "구직활동 시작", "자립지원센터 상담"],
        "3개월": ["안정적 일자리 확보", "생활비 관리 시스템 구축", "멘토 찾기"],
        "6개월": ["주거 독립 준비", "비상자금 마련", "사회보험 가입"]
    }
}
```

## 3.2 현재 엔진의 한계점

- **개인화 부족**: 사용자 프로필을 조회하지만 실제로는 활용하지 않음
- **고정된 템플릿**: 모든 사용자에게 동일한 로드맵 제공
- **조건부 로직 없음**: housing\_status나 income\_level에 따른 차별화 없음

## 4. 상세 계획 생성 엔진

### 4.1 generate\_detail\_plan() 함수 분석

```
python

def generate_detail_plan(period, goals):
    detail_plan = []

    for i, goal in enumerate(goals):
        if '주거' in goal:
            detail_plan.append({
                'title': f"{goal} - 세부 계획",
                'tasks': [
                    "주거급여 신청 자격 요건 확인하기",
                    "필요 서류 준비 (소득증명서, 임대차계약서)",
                    "관할 주민센터에서 신청 접수하기",
                    "심사 결과 확인 및 후속 조치",
                    "월별 주거비 관리 시스템 구축하기"
                ],
                'estimated_time': '2-3주'
            })
```

### 4.2 도메인별 세부 계획 템플릿

#### 주거 관련 계획

- 주거급여 신청 프로세스

- 필요 서류 준비 가이드
- 관할 기관 연결
- 주거비 관리 시스템

## 취업 관련 계획

- 이력서/자기소개서 작성
- 취업지원 프로그램 신청
- 직업훈련 과정 탐색
- 면접 준비 및 연습

## 경제 관리 계획

- 수입/지출 현황 파악
- 생계급여 지원 신청
- 가계부 작성 습관
- 비상자금 적립 계획

# 5. 진행 상황 추적 시스템

## 5.1 TODO 변환 메커니즘

python

```
def convert_to_todos():  
    # 로드맵의 각 목표를 개별 할일로 변환  
    for goal in goals:  
        cursor.execute("""  
            INSERT INTO progress_tracking  
            (user_id, roadmap_id, task_name, task_category, status, priority)  
            VALUES (?, ?, ?, ?, 'pending', 3)  
            """, (current_user.id, roadmap_id, goal, period))
```

## 5.2 진행률 계산 알고리즘

python

```
# roadmap() 함수에서 진행률 계산
cursor.execute("""
    SELECT COUNT(*) as total,
           SUM(CASE WHEN status = 'completed' THEN 1 ELSE 0 END) as completed
    FROM progress_tracking WHERE user_id = ?
""", (current_user.id,))

progress_data = cursor.fetchone()
if progress_data[0] > 0:
    progress_percentage = int((progress_data[1] / progress_data[0]) * 100)
```

## 6. 정책 추천 엔진 분석

### 6.1 점수 기반 추천 알고리즘

```
python

def get_recommended_policies(policies):
    for policy in policies:
        score = 0
        policy_text = (policy.get('서비스명', '') + ' ' +
                       policy.get('지원내용', '') + ' ' +
                       policy.get('지원대상', '')).lower()

        # 지원 요구사항 매칭
        for need in support_needs:
            if need == '주거지원' and ('주거' in policy_text or '임대' in policy_text):
                score += 3
            elif need == '경제지원' and ('생계' in policy_text or '급여' in policy_text):
                score += 3
        # ... 추가 매칭 로직

        # 자립준비청년 대상 정책 우선
        if '자립' in policy_text or '청소년' in policy_text:
            score += 2
```

### 6.2 추천 가중치 체계

- **직접 매칭:** 3점 (주거, 경제, 취업, 교육지원)
- **간접 지원:** 2점 (심리지원, 자립준비청년 대상)
- **최종 정렬:** 점수 순으로 상위 20개 정책 추천

## 7. 데이터베이스 설계

### 7.1 로드맵 저장 구조

```
sql

-- roadmaps 테이블
roadmaps (
  id INTEGER PRIMARY KEY,
  user_id INTEGER,
  title TEXT,
  description TEXT,
  priority_areas TEXT, -- JSON 형태로 저장
  timeline TEXT,      -- JSON 형태로 저장
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)

-- progress_tracking 테이블
progress_tracking (
  id INTEGER PRIMARY KEY,
  user_id INTEGER,
  roadmap_id INTEGER,
  task_name TEXT,
  task_category TEXT,
  status TEXT,        -- 'pending', 'in_progress', 'completed'
  priority INTEGER,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

### 7.2 JSON 데이터 처리

```
python

# 저장 시
json.dumps(roadmap['priority_areas'])
json.dumps(roadmap['timeline'])

# 조회 시
json.loads(roadmap_data[2]) if roadmap_data[2] else []
json.loads(roadmap_data[3]) if roadmap_data[3] else {}
```

## 8. 챗봇 연동 및 페이지 제안

### 8.1 메시지 패턴 분석

```
python
```

```
def generate_mock_response(message):
    message_lower = message.lower()

    if '로드맵' in message_lower or '계획' in message_lower:
        return "로드맵 생성을 도와드릴게요! 개인 맞춤형 자립 계획을 세워보시겠어요?"
```

## 8.2 스마트 페이지 제안

```
python

def check_for_page_suggestion(message):
    if '로드맵' in message_lower:
        return {
            'type': 'redirect',
            'url': '/roadmap',
            'message': '로드맵 페이지로 이동해서 자세한 계획을 확인해보시겠어요?'
        }
```

## 9. 현재 시스템의 강점과 한계

### 9.1 강점

- 모듈화된 구조: 각 기능이 독립적으로 구현
- 확장 가능한 설계: JSON 기반 유연한 데이터 구조
- 실용적 접근: 실제 자립준비청년이 필요한 영역 포괄
- 진행 추적: TODO 변환 및 진행률 계산 시스템

### 9.2 한계점

- 규칙 기반 한계: 복잡한 개인 상황 고려 부족
- 정적 템플릿: 동적 계획 생성 부족
- AI 미활용: OpenAI API 비활성화 상태
- 데이터 부족: 실제 사용자 행동 데이터 없음

## 10. 향후 개선 방향

### 10.1 AI 엔진 도입 계획

```
python
```

```
# 향후 OpenAI 연동 시 예상 구조
def generate_ai_roadmap(user_profile):
    prompt = f"""
    자립준비청년을 위한 맞춤형 로드맵 생성:
    - 주거상황: {profile['housing_status']}
    - 소득수준: {profile['income_level']}
    - 필요지원: {profile['support_needs']}

    단계별 실행 가능한 계획을 JSON 형태로 생성해주세요.
    """

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "system", "content": system_prompt},
                  {"role": "user", "content": prompt}]
    )
```

## 10.2 머신러닝 기반 개선안

- 사용자 행동 패턴 분석: 성공적인 자립 경로 학습
- 협업 필터링: 유사한 프로필 사용자 경험 활용
- 강화학습: 피드백 기반 로드맵 최적화

## 11. Q&A 대비 핵심 포인트

### 기술적 질문

Q: "현재 로드맵 생성이 AI 기반인가요?" A: 현재는 규칙 기반 템플릿 시스템이며, OpenAI API 연동은 개발 단계에서 비용 절약을 위해 비활성화. 실제 배포 시 AI 기반으로 전환 예정.

Q: "개인화 수준이 어느 정도인가요?"

A: 현재는 사용자 프로필을 수집하지만 템플릿 기반 제공. 향후 프로필 데이터 기반 조건부 로직과 AI 개인화 도입 계획.

Q: "로드맵 검증은 어떻게 하나요?" A: 전문가 자문단 구성 및 RAGAS 프레임워크 도입으로 로드맵 품질 평가 시스템 구축 예정.

### 비즈니스 질문

Q: "실제 자립준비청년의 피드백은 어떻게 반영하나요?" A: 진행률 추적 시스템과 TODO 변환 기능으로 사용자 행동 데이터 수집, 주간 피드백 주기로 개선사항 반영.

Q: "다른 지원 기관과의 연계는 어떻게 되나요?" A: 정책 DB에 기관 정보 포함하여 자동 연계, 향후 API 연동으로 실시간 정보 업데이트 계획.