EasyCrawler 프로젝트 기획서

1. 프로젝트 개요

1.1 프로젝트명

EasyCrawler - 초보자도 쉽게 사용하는 웹 크롤링 도구

1.2 프로젝트 목표

코딩 지식이 없는 일반 사용자도 웹페이지에서 원하는 데이터를 쉽게 수집하고, 정기적으로 자동화할 수 있는 도구 개발

1.3 타겟 사용자

- 마케팅 담당자 (경쟁사 가격 모니터링)
- 부동산 관련자 (매물 정보 수집)
- 쇼핑몰 운영자 (상품 가격 비교)
- 연구자/학생 (데이터 수집)
- 일반인 (관심 정보 정기 수집)

1.4 핵심 가치 제안

- **Zero-Code**: 코딩 없이 클릭만으로 크롤링
- **Template-Based**: 한 번 설정하면 재사용 가능
- **Auto-Schedule**: 정기적 자동 실행
- **Excel Ready**: 바로 활용 가능한 결과물

2. 기능 명세서

2.1 Core Features (Phase 1)

2.1.1 사이트 분석 모듈

기능: 크롤링 가능성 자동 판단

- **F001**: robots.txt 자동 체크
- **F002**: JavaScript 렌더링 필요 여부 감지
- **F003**: 로그인 요구 사항 감지
- **F004**: Rate Limiting 정책 확인
- **F005**: 크롤링 난이도 점수 표시 (★☆☆ ~ ★★★)

User Story:

> 사용자가 URL을 입력하면, 시스템이 자동으로 해당 사이트의 크롤링 가능성을 분석하여 "쉬움/보통/어려움" 등급으로 표시해준다.

2.1.2 Visual Element Selector

기능: 브라우저에서 직접 요소 선택

- **F006**: 웹페이지 오버레이 표시
- **F007**: 마우스 호버시 요소 하이라이트
- **F008**: 클릭으로 데이터 요소 선택
- **F009**: 선택된 요소의 CSS 셀렉터 자동 생성
- **F010**: 유사한 패턴 요소 자동 감지

User Story:

> 사용자가 웹페이지에서 수집하고 싶은 제품명을 클릭하면, 시스템이 자동으로 해당 페이지의 모든 제품명을 인식하고 선택할 수 있게 해준다.

2.1.3 패턴 저장 시스템

기능: 크롤링 템플릿 생성 및 관리

- **F011**: 선택된 요소들을 템플릿으로 저장
- **F012**: 템플릿에 의미있는 이름 부여
- **F013**: 필드별 데이터 타입 설정 (텍스트/숫자/날짜/URL)
- **F014**: 템플릿 목록 관리 (추가/수정/삭제)
- **F015**: 템플릿 내보내기/가져오기 (JSON 형태)

User Story:

> 사용자가 쇼핑몰에서 "상품명, 가격, 평점"을 선택한 후, "쇼핑몰_상품정보"라는 이름으로 템플릿을 저장하면, 나중에 다른 쇼핑몰에서도 동일한 패턴으로 재사용할 수 있다.

2.1.4 데이터 추출 엔진

기능: 실제 데이터 크롤링 실행

- **F016**: 단일 페이지 크롤링
- **F017**: 다중 페이지 크롤링 (페이지네이션 지원)
- **F018**: 메타데이터 우선 추출 (Open Graph, JSON-LD)
- **F019**: CSS 셀렉터 기반 추출
- **F020**: 에러 처리 및 재시도 로직

User Story:

> 사용자가 생성한 템플릿으로 "실행" 버튼을 클릭하면, 시스템이 자동으로 해당 웹사이트에서 데이터를 수집하여 표 형태로 보여준다.

2.1.5 엑셀 출력 모듈

기능: 수집된 데이터를 Excel 파일로 변환

- **F021**: CSV 형태 기본 출력
- **F022**: Excel (.xlsx) 형태 출력
- **F023**: 컬럼 헤더 자동 생성
- **F024**: 데이터 타입별 셀 포맷팅
- **F025**: 수집 일시 자동 기록

User Story:

> 크롤링이 완료되면 사용자는 "엑셀 다운로드" 버튼을 클릭하여 수집된 데이터를 즉시 Excel 파일로 받을 수 있다.

2.2 Advanced Features (Phase 2)

2.2.1 스케줄링 시스템

- **F026**: 크롤링 주기 설정 (매일/매주/매월)
- **F027**: 특정 시간 지정 실행
- **F028**: 백그라운드 실행
- **F029**: 실행 결과 알림 (이메일/슬랙)
- **F030**: 실행 로그 관리

2.2.2 고급 데이터 처리

- **F031**: 중복 데이터 제거
- **F032**: 데이터 정제 (공백 제거, 형식 통일)
- **F033**: 계산 필드 추가 (가격 차이, 증감률 등)
- **F034**: 데이터 필터링 조건 설정
- **F035**: 변화 감지 및 알림

2.3 Enterprise Features (Phase 3)

- **F036**: 클라우드 저장소 연동
- **F037**: 팀 템플릿 공유
- **F038**: API 엔드포인트 제공
- **F039**: 대시보드 및 차트

- **F040**: 데이터베이스 직접 저장

3. 기술 스택

3.1 Frontend

- **Chrome Extension**: Manifest V3
- **Technologies**: HTML5, CSS3, JavaScript (ES6+)
- **UI Framework**: Bootstrap 5 또는 Tailwind CSS
- **Icons**: Font Awesome

3.2 Backend

- **Language**: Python 3.9+
- **Framework**: FastAPI
- **Web Scraping**:
  - BeautifulSoup4 (정적 페이지)
  - Selenium (동적 페이지)
  - Playwright (고성능 대안)
- **Data Processing**: Pandas
- **Excel Export**: openpyxl

3.3 Database

- **Primary**: SQLite (로컬) → PostgreSQL (운영)
- **Cache**: Redis (세션 및 임시 데이터)
- **File Storage**: 로컬 파일시스템 → AWS S3

3.4 Infrastructure

- **Development**: Local Python Server
- **Production**: Docker + Docker Compose
- **Deployment**: AWS EC2 또는 Railway
- **Monitoring**: 기본 로깅 → ELK Stack

4. 아키텍처 설계

4.1 시스템 구조도

```
[Chrome Extension]
      ↓ (API 호출)
[FastAPI Backend]
      ↓
[크롤링 엔진] ← [템플릿 DB]
      ↓
[데이터 처리] → [Excel 생성]
      ↓
[스케줄러] → [결과 저장]
```

4.2 폴더 구조

```
easycrawler/
├── extension/              # Chrome Extension
│   ├── manifest.json
│   ├── popup/
│   ├── content/
│   └── background/
├── backend/                # Python Backend
│   ├── app/
│   │   ├── api/            # API 엔드포인트
│   │   ├── core/           # 핵심 로직
│   │   ├── models/         # 데이터 모델
│   │   └── services/       # 비즈니스 로직
│   ├── crawler/            # 크롤링 엔진
│   ├── scheduler/          # 스케줄링
│   └── tests/              # 테스트
├── templates/              # 템플릿 저장소
├── docs/                   # 문서
└── deploy/                 # 배포 스크립트
```

4.3 데이터베이스 스키마

sql

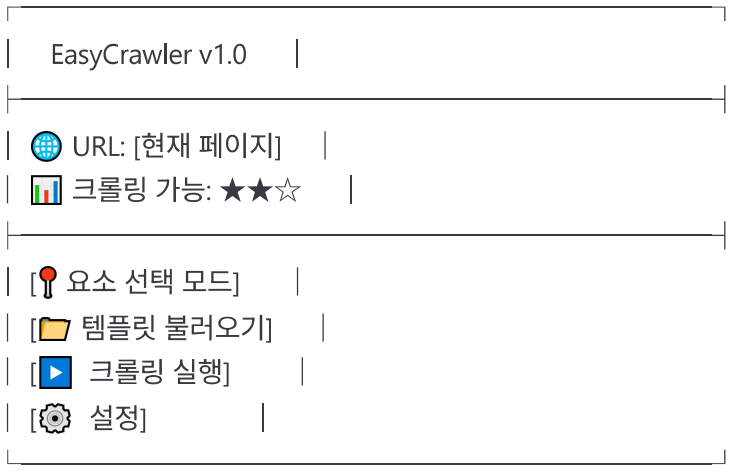
```sql
-- 템플릿 정보
CREATE TABLE templates (
    id INTEGER PRIMARY KEY,
    name VARCHAR(100),
    url_pattern VARCHAR(500),
    selectors JSON,
    created_at TIMESTAMP,
    updated_at TIMESTAMP
);

-- 크롤링 작업
CREATE TABLE crawl_jobs (
    id INTEGER PRIMARY KEY,
    template_id INTEGER,
    status VARCHAR(20),
    scheduled_at TIMESTAMP,
    completed_at TIMESTAMP,
    result_file VARCHAR(200)
);

-- 수집된 데이터
CREATE TABLE crawl_data (
    id INTEGER PRIMARY KEY,
    job_id INTEGER,
    data JSON,
    crawled_at TIMESTAMP
);
```

5. UI/UX 와이어프레임

5.1 Chrome Extension Popup



5.2 요소 선택 모드

웹페이지 오버레이:

```
┌──────────────────────────────┐
│ 🎯 클릭하여 요소 선택        │
│ 선택됨: [제품명] [가격]      │
│ [✅ 완료] [❌ 취소]          │
└──────────────────────────────┘
```

5.3 결과 화면

```
┌──────────────────────────────┐
│ 📊 크롤링 결과 (25개)        │
├──────────────────────────────┤
│ 제품명    | 가격   | 평점 |
│ 아이폰14  | 99만원 | 4.5 |
│ 갤럭시S23 | 89만원 | 4.3 |
├──────────────────────────────┤
│ [📥 Excel 다운로드]          │
│ [💾 템플릿 저장]             │
│ [🔄 다시 실행]               │
└──────────────────────────────┘
```

6. 개발 일정

Phase 1 (4주)

- **Week 1**: 프로젝트 설정 + 사이트 분석 모듈
- **Week 2**: Chrome Extension + Visual Selector
- **Week 3**: 크롤링 엔진 + 패턴 저장
- **Week 4**: Excel 출력 + 통합 테스트

Phase 2 (3주)

- **Week 5**: 스케줄링 시스템
- **Week 6**: 고급 데이터 처리
- **Week 7**: UI/UX 개선 + 버그 수정

Phase 3 (3주)

- **Week 8-10**: Enterprise 기능 + 배포

7. 성공 지표

7.1 기술적 지표

- 크롤링 성공률: 95% 이상

- 응답 시간: 평균 3초 이내
- 확장프로그램 크기: 5MB 이하
- 메모리 사용량: 100MB 이하

7.2 사용성 지표

- 첫 크롤링까지 소요 시간: 5분 이내
- 템플릿 재사용률: 80% 이상
- 사용자 만족도: 4.0/5.0 이상

8. 리스크 및 대응 방안

8.1 기술적 리스크

리스크	확률	영향도	대응 방안
웹사이트 구조 변경	높음	중간	자동 감지 + 알림 시스템
크롤링 차단	중간	높음	User-Agent 로테이션, 프록시
성능 이슈	중간	중간	비동기 처리, 캐싱

8.2 법적 리스크

리스크	대응 방안
저작권 침해	이용약관 명시, 개인용도 권장
개인정보 수집	데이터 암호화, 로컬 저장 우선

9. 라이센스 및 배포

9.1 오픈소스 라이센스

- **MIT License**: 상업적 이용 허용
- **개인 정보 보호**: 로컬 우선 처리
- **데이터 보안**: 사용자 책임 명시

9.2 배포 계획

- **Phase 1**: GitHub Release (개발자용)
- **Phase 2**: Chrome Web Store (일반용)
- **Phase 3**: 웹 서비스 형태 제공

다음 단계

1. **상세 작업 지침서** 작성

2. **개발 환경 설정**

3. **프로토타입 개발 시작**