쇼핑Iz - 통합 가격비교 & 쿠폰 거래 플랫폼 기획서

1. 서비스 개요

1.1 핵심 가치

One-Stop 최저가 쇼핑: 모든 쇼핑몰의 실시간 가격과 쿠폰을 한 곳에서

쿠폰 거래 마켓: 사용하지 않는 쿠폰을 다른 사용자와 교환/판매

AI 가격 예측: 최적 구매 시점 추천

1.2 주요 기능

실시간 가격 비교 (쿠팡, 네이버, 11번가, G마켓 등)

통합 쿠폰 수집 및 자동 적용

P2P 쿠폰 거래소

가격 추적 및 알림

AI 기반 추가 할인 예측

2. MySQL 데이터베이스 설계

sql

```sql
-- 1. 사용자 관련 테이블
CREATE TABLE users (
    user_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(100) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    nickname VARCHAR(50) UNIQUE NOT NULL,
    phone VARCHAR(20),
    point INT DEFAULT 0,
    trust_score DECIMAL(3,2) DEFAULT 5.00,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    INDEX idx_email (email),
    INDEX idx_nickname (nickname)
);

-- 2. 상품 정보 테이블
CREATE TABLE products (
    product_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    barcode VARCHAR(50),
    product_name VARCHAR(255) NOT NULL,
    category_id INT,
    brand VARCHAR(100),
    model_number VARCHAR(100),
    image_url VARCHAR(500),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_barcode (barcode),
    INDEX idx_name (product_name),
    FULLTEXT idx_fulltext_name (product_name)
);

-- 3. 카테고리 테이블
CREATE TABLE categories (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    parent_id INT,
    category_name VARCHAR(100) NOT NULL,
    depth TINYINT DEFAULT 0,
    FOREIGN KEY (parent_id) REFERENCES categories(category_id)
);

-- 4. 쇼핑몰 플랫폼 테이블
CREATE TABLE platforms (
    platform_id INT PRIMARY KEY AUTO_INCREMENT,
    platform_name VARCHAR(50) NOT NULL,
    platform_url VARCHAR(255),
    api_key VARCHAR(255),
    crawling_interval INT DEFAULT 3600,
    is_active BOOLEAN DEFAULT TRUE
);

-- 5. 가격 추적 테이블
CREATE TABLE price_tracking (
    tracking_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    product_id BIGINT,
    platform_id INT,
    current_price DECIMAL(12,2) NOT NULL,
    original_price DECIMAL(12,2),
    discount_rate DECIMAL(5,2),
    shipping_fee DECIMAL(8,2) DEFAULT 0,
    product_url VARCHAR(500),
    in_stock BOOLEAN DEFAULT TRUE,
    tracked_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    FOREIGN KEY (platform_id) REFERENCES platforms(platform_id),
    INDEX idx_product_platform (product_id, platform_id),
    INDEX idx_tracked_at (tracked_at)
);
```

```
-- 6. 쿠폰 마스터 테이블
CREATE TABLE coupons (
    coupon_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    platform_id INT,
    coupon_code VARCHAR(100),
    coupon_name VARCHAR(255) NOT NULL,
    discount_type ENUM('PERCENTAGE', 'FIXED', 'SHIPPING_FREE') NOT NULL,
    discount_value DECIMAL(10,2) NOT NULL,
    minimum_purchase DECIMAL(10,2) DEFAULT 0,
    max_discount DECIMAL(10,2),
    category_id INT,
    product_id BIGINT,
    valid_from DATETIME NOT NULL,
    valid_until DATETIME NOT NULL,
    usage_limit INT,
    is_public BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (platform_id) REFERENCES platforms(platform_id),
    FOREIGN KEY (category_id) REFERENCES categories(category_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    INDEX idx_valid_date (valid_from, valid_until),
    INDEX idx_platform (platform_id)
);

-- 7. 사용자 쿠폰 보유 테이블
CREATE TABLE user_coupons (
    user_coupon_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT NOT NULL,
    coupon_id BIGINT NOT NULL,
    status ENUM('AVAILABLE', 'USED', 'EXPIRED', 'TRADING', 'TRADED') DEFAULT 'AVAILABLE',
    acquired_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    used_at TIMESTAMP NULL,
    is_tradeable BOOLEAN DEFAULT TRUE,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (coupon_id) REFERENCES coupons(coupon_id),
    INDEX idx_user_status (user_id, status),
    INDEX idx_coupon (coupon_id)
);
```

```sql
-- 8. 쿠폰 거래소 테이블
CREATE TABLE coupon_trades (
    trade_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    seller_id BIGINT NOT NULL,
    buyer_id BIGINT,
    user_coupon_id BIGINT NOT NULL,
    trade_type ENUM('EXCHANGE', 'SELL', 'FREE') NOT NULL,
    asking_price INT DEFAULT 0,
    asking_coupon_category VARCHAR(100),
    status ENUM('OPEN', 'RESERVED', 'COMPLETED', 'CANCELLED') DEFAULT 'OPEN',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completed_at TIMESTAMP NULL,
    FOREIGN KEY (seller_id) REFERENCES users(user_id),
    FOREIGN KEY (buyer_id) REFERENCES users(user_id),
    FOREIGN KEY (user_coupon_id) REFERENCES user_coupons(user_coupon_id),
    INDEX idx_status (status),
    INDEX idx_seller (seller_id),
    INDEX idx_created (created_at)
);

-- 9. 쿠폰 교환 제안 테이블
CREATE TABLE trade_offers (
    offer_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    trade_id BIGINT NOT NULL,
    offerer_id BIGINT NOT NULL,
    offered_coupon_id BIGINT,
    offered_points INT DEFAULT 0,
    message TEXT,
    status ENUM('PENDING', 'ACCEPTED', 'REJECTED', 'CANCELLED') DEFAULT 'PENDING',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (trade_id) REFERENCES coupon_trades(trade_id),
    FOREIGN KEY (offerer_id) REFERENCES users(user_id),
    FOREIGN KEY (offered_coupon_id) REFERENCES user_coupons(user_coupon_id),
    INDEX idx_trade (trade_id),
    INDEX idx_status (status)
);

-- 10. 가격 알림 설정 테이블
CREATE TABLE price_alerts (
    alert_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT NOT NULL,
    product_id BIGINT NOT NULL,
    target_price DECIMAL(12,2),
    alert_type ENUM('PRICE_DROP', 'IN_STOCK', 'NEW_COUPON') NOT NULL,
    is_active BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    INDEX idx_user_active (user_id, is_active),
    INDEX idx_product (product_id)
);

-- 11. 사용자 검색 기록 테이블
CREATE TABLE search_history (
    history_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT NOT NULL,
    search_keyword VARCHAR(255) NOT NULL,
    result_count INT DEFAULT 0,
    searched_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    INDEX idx_user (user_id),
    INDEX idx_keyword (search_keyword),
    INDEX idx_searched_at (searched_at)
);

-- 12. 쿠폰 사용 기록 테이블
CREATE TABLE coupon_usage_log (
    log_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_coupon_id BIGINT NOT NULL,
    product_id BIGINT,
    platform_id INT,
    original_price DECIMAL(12,2),
    discount_amount DECIMAL(12,2),
    final_price DECIMAL(12,2),
    used_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_coupon_id) REFERENCES user_coupons(user_coupon_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    FOREIGN KEY (platform_id) REFERENCES platforms(platform_id),
    INDEX idx_used_at (used_at)
);
```

```
-- 13. 쿠폰 자동 수집 로그 테이블
CREATE TABLE coupon_crawl_log (
    log_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    platform_id INT NOT NULL,
    crawled_count INT DEFAULT 0,
    new_count INT DEFAULT 0,
    updated_count INT DEFAULT 0,
    error_count INT DEFAULT 0,
    crawl_status ENUM('SUCCESS', 'PARTIAL', 'FAILED') NOT NULL,
    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completed_at TIMESTAMP NULL,
    error_message TEXT,
    FOREIGN KEY (platform_id) REFERENCES platforms(platform_id),
    INDEX idx_platform_time (platform_id, started_at)
);

-- 14. 거래 평가 테이블
CREATE TABLE trade_reviews (
    review_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    trade_id BIGINT NOT NULL,
    reviewer_id BIGINT NOT NULL,
    reviewed_id BIGINT NOT NULL,
    rating TINYINT CHECK (rating >= 1 AND rating <= 5),
    comment TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (trade_id) REFERENCES coupon_trades(trade_id),
    FOREIGN KEY (reviewer_id) REFERENCES users(user_id),
    FOREIGN KEY (reviewed_id) REFERENCES users(user_id),
    UNIQUE KEY unique_trade_reviewer (trade_id, reviewer_id),
    INDEX idx_reviewed (reviewed_id)
);

-- 15. 포인트 거래 내역 테이블
CREATE TABLE point_transactions (
    transaction_id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT NOT NULL,
    amount INT NOT NULL,
    transaction_type ENUM('CHARGE', 'WITHDRAW', 'TRADE_BUY', 'TRADE_SELL',
'REWARD') NOT NULL,
    reference_id BIGINT,
    balance_after INT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    INDEX idx_user_time (user_id, created_at)
);
```

3. 핵심 기능 상세

3.1 실시간 가격 비교

python

```
# 가격 비교 API 예시
GET /api/products/{product_id}/prices
{
  "product": {
    "id": 12345,
    "name": "삼성 갤럭시 버즈3",
    "lowest_price": 179000,
    "highest_price": 219000
  },
  "prices": [
    {
      "platform": "쿠팡",
      "price": 179000,
      "shipping": 0,
      "coupon_available": true,
      "final_price": 169000
    }
  ]
}
```

3.2 쿠폰 자동 수집 시스템

각 플랫폼 API/크롤링을 통한 실시간 수집
중복 제거 및 유효성 검증
사용자별 맞춤 쿠폰 추천

3.3 P2P 쿠폰 거래소

쿠폰 ↔ 쿠폰 교환
쿠폰 ↔ 포인트 거래
에스크로 시스템으로 안전 거래
신뢰도 기반 거래 시스템

3.4 AI 가격 예측

python

```
# 가격 예측 모델 예시
```

- 과거 가격 변동 패턴 분석
- 행사 일정 (블랙프라이데이, 정기세일 등)
- 재고 수준 추정
- 경쟁사 가격 동향

사용언어 -> 파이썬 제외하고, ec2 호스팅 쉬운거