

그룹 코드 관리 화면 (SYS-SCR-001)

사용 역할자: 시스템 관리자

관련 메뉴: 시스템 관리 > 코드 관리 > 그룹 코드 관리

기능 1: 그룹 코드 목록 조회

- 관련 요구사항: SYS-REQ-003 (코드 관리), SYS-REQ-034 (코드 관리 - 공통/그룹 코드 정리 및 통합)
- 입력 정보:
 - 검색 조건:
 - 그룹 코드 ID: 텍스트 입력 필드 (부분 일치 검색 가능)
 - 그룹 코드명: 텍스트 입력 필드 (부분 일치 검색 가능)
 - 사용 여부: Select 박스 (전체, 예, 아니오)
 - '검색' 버튼: 입력된 조건으로 목록 조회 요청.
 - 페이지네이션: 페이지 번호 클릭 또는 '다음/이전' 버튼 클릭.
 - 목록 정렬: 컬럼 헤더 클릭 (그룹 코드 ID, 그룹 코드명 등).
- 출력 정보:
 - 그룹 코드 목록 테이블:
 - 컬럼: 체크박스, 번호(No), 그룹 코드 ID, 그룹 코드명, 설명, 사용 여부, 등록일, 수정일.
 - 각 행은 하나의 그룹 코드를 나타냄.
 - 그룹 코드 ID 클릭 시 상세 조회/수정 팝업 또는 화면으로 이동.
 - 총 그룹 코드 수: 검색 결과에 해당하는 전체 항목 수 표시.
 - 페이지네이션 컴포넌트: 현재 페이지, 전체 페이지 수, 페이지 이동 버튼 표시.
- 프로세스 설명:
 1. 관리자가 그룹 코드 관리 화면에 진입하면, 기본적으로 모든 그룹 코드 목록(또는 첫 페이지)이 조회되어 표시된다.
 2. 관리자는 필요에 따라 검색 조건(그룹 코드 ID, 명칭, 사용 여부)을 입력하고 '검색' 버튼을 클릭한다.
 3. 프론트엔드는 검색 조건과 페이징 정보(페이지 번호, 페이지 당 항목 수)를 백엔드 API(/api/v1/admin/codes/groups)에 GET 방식으로 전송한다.
 4. 백엔드는 조건에 맞는 그룹 코드 목록을 DB에서 조회하여 페이징 처리 후 결과를 반환한다.
 5. 프론트엔드는 응답받은 데이터를 테이블 형태로 렌더링하고, 총 항목 수 및 페이지네이션 정보를 업데이트한다.
 6. 관리자는 페이지 번호를 클릭하거나 정렬 기준을 변경하여 목록을 탐색할 수 있다.
- 호출 API 정보:
 - HTTP 메서드: GET
 - API 엔드포인트: /api/v1/admin/codes/groups
 - API 목적: 그룹 코드 목록을 조건에 따라 조회 (페이징 포함).
 - 요청 파라미터 (Query Parameters): page, size, sort, groupId, groupName, isActive
 - 응답 데이터 (성공):

```
{
  "content": [
    {
      "groupId": "COURSE_TYPE",
```

```

        "groupName": "과정 유형",
        "description": "온라인, 오프라인, 혼합 등 과정 유형 분류",
        "isActive": true,
        "createdAt": "2024-01-10T10:00:00Z",
        "updatedAt": "2024-01-10T10:00:00Z"
    }
    // ... other group codes
],
"pageable": { "pageNumber": 0, "pageSize": 10, ... },
"totalElements": 25,
"totalPages": 3
}

```

- Mockup 데이터 예시: 생략 (위 응답 데이터 형식 참조)

기능 2: 그룹 코드 상세 조회/수정

- 관련 요구사항: SYS-REQ-003, SYS-REQ-034
- 입력 정보:
 - (목록에서 그룹 코드 ID 클릭 시) 해당 그룹 코드 정보가 채워진 폼 표시.
 - **그룹 코드 ID**: 텍스트 필드 (수정 불가, 읽기 전용).
 - **그룹 코드명**: 텍스트 필드 (수정 가능). 필수.
 - **설명**: 텍스트 영역 (수정 가능).
 - **사용 여부**: 라디오 버튼 또는 스위치 (예, 아니오). 필수.
 - **'저장' 버튼**: 수정된 내용 저장 요청.
 - **'취소' 버튼**: 변경 내용 취소 및 팝업/화면 닫기.
- 출력 정보:
 - 수정 성공 시: "그룹 코드 정보가 저장되었습니다." 메시지 표시 및 목록 새로고침 또는 팝업 닫기.
 - 수정 실패 시 (유효성 오류, 서버 오류 등): 오류 메시지 표시.
- 프로세스 설명:
 1. 관리자가 목록에서 특정 그룹 코드 ID를 클릭한다.
 2. 해당 그룹 코드의 상세 정보가 조회되어 수정 폼에 표시된다 (API: **GET** `/api/v1/admin/codes/groups/{groupId}`).
 3. 관리자가 필요한 정보를 수정(그룹 코드명, 설명, 사용 여부)하고 '저장' 버튼을 클릭한다.
 4. 프론트엔드는 수정된 정보를 JSON 형태로 백엔드 API(`/api/v1/admin/codes/groups/{groupId}`)에 PUT 방식으로 전송한다.
 5. 백엔드는 전달받은 정보로 해당 그룹 코드를 업데이트한다.
 6. 업데이트 성공 시 성공 응답 반환, 실패 시 실패 응답 반환.
 7. 프론트엔드는 결과에 따라 메시지를 표시하고, 목록을 새로고침하거나 팝업을 닫는다.
- 호출 API 정보:
 - **그룹 코드 상세 조회 API**:
 - HTTP 메서드: **GET**
 - API 엔드포인트: `/api/v1/admin/codes/groups/{groupId}`
 - **그룹 코드 수정 API**:
 - HTTP 메서드: **PUT**
 - API 엔드포인트: `/api/v1/admin/codes/groups/{groupId}`

- 요청 바디: {"groupName": "string", "description": "string", "isActive": boolean}
 - 응답: {"success": true/false, "message": "string"}
- Mockup 데이터 예시: 생략

기능 3: 그룹 코드 신규 등록

- 관련 요구사항: SYS-REQ-003, SYS-REQ-034
- 입력 정보:
 - '신규 등록' 버튼: 클릭 시 비어있는 등록 폼 팝업 또는 섹션 표시.
 - 그룹 코드 ID: 텍스트 필드. 필수. 영문 대문자, 숫자, 언더스코어(_) 조합 권장. 중복 불가.
 - 그룹 코드명: 텍스트 필드. 필수.
 - 설명: 텍스트 영역.
 - 사용 여부: 라디오 버튼 또는 스위치 (예, 아니오). 기본값 '예'. 필수.
 - '등록' 버튼: 입력된 정보로 신규 그룹 코드 등록 요청.
 - '취소' 버튼: 입력 내용 취소 및 팝업/화면 닫기.
- 출력 정보:
 - 그룹 코드 ID 중복 시: "이미 사용 중인 그룹 코드 ID입니다." 메시지 표시 (실시간 또는 등록 버튼 클릭 시).
 - 등록 성공 시: "그룹 코드가 등록되었습니다." 메시지 표시 및 목록 새로고침 또는 팝업 닫기.
 - 등록 실패 시: 오류 메시지 표시.
- 프로세스 설명:
 1. 관리자가 화면 상단의 '신규 등록' 버튼을 클릭한다.
 2. 비어있는 그룹 코드 등록 폼이 나타난다.
 3. 관리자가 그룹 코드 ID, 명칭, 설명, 사용 여부를 입력한다. (그룹 코드 ID 입력 시 중복 확인 API 호출 가능: `GET /api/v1/admin/codes/groups/check-id?groupId=...`)
 4. 관리자가 '등록' 버튼을 클릭한다.
 5. 프론트엔드는 입력된 정보를 JSON 형태로 백엔드 API(`/api/v1/admin/codes/groups`)에 POST 방식으로 전송한다.
 6. 백엔드는 ID 중복 여부 최종 확인 후, 유효하면 새로운 그룹 코드를 DB에 등록한다.
 7. 등록 성공 시 성공 응답 반환, 실패 시 실패 응답 반환.
 8. 프론트엔드는 결과에 따라 메시지를 표시하고, 목록을 새로고침하거나 팝업을 닫는다.
- 호출 API 정보:
 - 그룹 코드 ID 중복 확인 API:
 - HTTP 메서드: GET
 - API 엔드포인트: `/api/v1/admin/codes/groups/check-id?groupId={groupId}`
 - 응답: {"isActive": true/false}
 - 그룹 코드 등록 API:
 - HTTP 메서드: POST
 - API 엔드포인트: `/api/v1/admin/codes/groups`
 - 요청 바디: {"groupId": "string", "groupName": "string", "description": "string", "isActive": boolean}
 - 응답: {"success": true/false, "message": "string"}
- Mockup 데이터 예시: 생략

예외 처리 및 유의 사항

- **ID 규칙:** 그룹 코드 ID는 시스템 전반에서 사용되므로, 명명 규칙(영문 대문자, 숫자, _)을 정하고 일관되게 관리해야 함.
- **삭제 정책:** 그룹 코드는 하위 상세 코드를 가지므로, 일반적으로 물리적 삭제보다는 '사용 여부'를 '아니오'로 변경하여 비활성화하는 방식을 사용함. (삭제 기능 구현 시 하위 코드 존재 여부 체크 필요)
- **일괄 변경:** 필요시 CSV 업로드/다운로드를 통한 일괄 등록/수정 기능 고려 가능.

상세 코드 관리 화면 (SYS-SCR-002)

사용 역할자: 시스템 관리자

관련 메뉴: 시스템 관리 > 코드 관리 > 상세 코드 관리

기능 1: 상세 코드 목록 조회

- 관련 요구사항: SYS-REQ-003, SYS-REQ-034
- 입력 정보:
 - **그룹 코드 선택:** Select 박스 또는 자동 완성 검색. 필수. (SYS-SCR-001에서 관리되는 그룹 코드 목록 연동)
 - **검색 조건:**
 - 상세 코드 값: 텍스트 입력 필드.
 - 상세 코드명: 텍스트 입력 필드.
 - 사용 여부: Select 박스 (전체, 예, 아니오).
 - **'검색' 버튼:** 선택된 그룹 코드 및 검색 조건으로 목록 조회 요청.
 - **페이지네이션 및 정렬:** 그룹 코드 관리 화면과 동일.
- 출력 정보:
 - **상세 코드 목록 테이블:**
 - 컬럼: 체크박스, 번호(No), 상세 코드 값, 상세 코드명, 설명, 정렬 순서, 사용 여부, 등록일, 수정일.
 - 선택된 그룹 코드에 속한 상세 코드만 표시됨.
 - 상세 코드 값 클릭 시 상세 조회/수정 팝업 또는 화면 이동.
 - **총 상세 코드 수, 페이지네이션 컴포넌트:** 그룹 코드 관리 화면과 동일.
- 프로세스 설명:
 1. 관리자가 상세 코드 관리 화면에 진입하면, 먼저 그룹 코드를 선택하도록 유도한다.
 2. 관리자가 그룹 코드를 선택하면, 해당 그룹 코드에 속한 상세 코드 목록이 조회되어 표시된다. (API: [GET /api/v1/admin/codes/details?groupId=...](#))
 3. 관리자는 필요에 따라 추가 검색 조건(상세 코드 값/명, 사용 여부)을 입력하고 '검색' 버튼을 클릭하여 목록을 필터링할 수 있다.
 4. 페이지징 및 정렬 기능은 그룹 코드 관리 화면과 동일하게 동작한다.
- 호출 API 정보:
 - **그룹 코드 목록 조회 API (Select 박스용):**
 - HTTP 메서드: [GET](#)
 - API 엔드포인트: [/api/v1/admin/codes/groups?size=1000&isActive=true](#) (사용 중인 그룹 코드만 조회)
 - **상세 코드 목록 조회 API:**
 - HTTP 메서드: [GET](#)
 - API 엔드포인트: [/api/v1/admin/codes/details](#)

- 요청 파라미터: **groupId** (필수), **page**, **size**, **sort**, **codeValue**, **codeName**, **isActive**
- 응답 데이터: 그룹 코드 목록 조회 응답과 유사한 구조. 상세 코드 정보 포함.

```
{
  "content": [
    {
      "codeValue": "ONLINE",
      "codeName": "온라인 과정",
      "description": "전체 학습이 온라인으로 진행되는 과정",
      "sortOrder": 1,
      "isActive": true,
      "createdAt": "2024-01-11T10:00:00Z",
      "updatedAt": "2024-01-11T10:00:00Z"
    }
    // ... other detail codes for the selected group
  ],
  // ... pageable, totalElements, totalPages
}
```

- Mockup 데이터 예시: 생략

기능 2: 상세 코드 상세 조회/수정

- 관련 요구사항: SYS-REQ-003, SYS-REQ-034
- 입력 정보:
 - (목록에서 상세 코드 값 클릭 시) 해당 상세 코드 정보가 채워진 폼 표시.
 - **그룹 코드 ID**: 표시 (수정 불가).
 - **상세 코드 값**: 텍스트 필드 (수정 불가, 읽기 전용).
 - **상세 코드명**: 텍스트 필드 (수정 가능). 필수.
 - **설명**: 텍스트 영역 (수정 가능).
 - **정렬 순서**: 숫자 입력 필드 (수정 가능). 기본값 0. 목록 표시 순서 제어.
 - **사용 여부**: 라디오 버튼 또는 스위치. 필수.
 - **'저장' 버튼**: 수정된 내용 저장 요청.
 - **'취소' 버튼**: 변경 내용 취소.
- 출력 정보:
 - 수정 성공/실패 시 메시지 표시 및 목록 새로고침. (그룹 코드 관리 화면과 동일)
- 프로세스 설명:
 1. 관리자가 목록에서 특정 상세 코드 값을 클릭한다.
 2. 해당 상세 코드의 상세 정보가 조회되어 수정 폼에 표시된다 (API: **GET** `/api/v1/admin/codes/details/{codeValue}?groupId=...`).
 3. 관리자가 필요한 정보(상세 코드명, 설명, 정렬 순서, 사용 여부)를 수정하고 '저장' 버튼을 클릭한다.
 4. 프론트엔드는 수정된 정보를 백엔드 API(`/api/v1/admin/codes/details/{codeValue}?groupId=...`)에 PUT 방식으로 전송한다.
 5. 백엔드는 해당 상세 코드를 업데이트한다.
 6. 결과에 따라 프론트엔드에서 메시지 표시 및 후속 처리 수행.
- 호출 API 정보:

- 상세 코드 상세 조회 API:
 - HTTP 메서드: GET
 - API 엔드포인트: `/api/v1/admin/codes/details/{codeValue}`
 - 요청 파라미터: `groupId` (필수)
- 상세 코드 수정 API:
 - HTTP 메서드: PUT
 - API 엔드포인트: `/api/v1/admin/codes/details/{codeValue}`
 - 요청 파라미터: `groupId` (필수)
 - 요청 바디: `{"codeName": "string", "description": "string", "sortOrder": number, "isActive": boolean}`
 - 응답: `{"success": true/false, "message": "string"}`
- Mockup 데이터 예시: 생략

기능 3: 상세 코드 신규 등록

- 관련 요구사항: SYS-REQ-003, SYS-REQ-034
- 입력 정보:
 - (현재 선택된 그룹 코드 하위에 등록)
 - '신규 등록' 버튼: 클릭 시 비어있는 등록 폼 표시.
 - 그룹 코드 ID: 현재 선택된 값 표시 (수정 불가).
 - 상세 코드 값: 텍스트 필드. 필수. 영문 대문자, 숫자, 언더스코어 권장. 해당 그룹 내 중복 불가.
 - 상세 코드명: 텍스트 필드. 필수.
 - 설명: 텍스트 영역.
 - 정렬 순서: 숫자 입력 필드. 기본값 0.
 - 사용 여부: 라디오 버튼 또는 스위치. 기본값 '예'. 필수.
 - '등록' 버튼: 신규 상세 코드 등록 요청.
 - '취소' 버튼: 입력 내용 취소.
- 출력 정보:
 - 상세 코드 값 중복 시: "이미 사용 중인 상세 코드 값입니다." 메시지 표시.
 - 등록 성공/실패 시 메시지 표시 및 목록 새로고침. (그룹 코드 관리 화면과 동일)
- 프로세스 설명:
 1. 관리자가 특정 그룹 코드를 선택한 상태에서 '신규 등록' 버튼을 클릭한다.
 2. 비어있는 상세 코드 등록 폼이 나타난다 (그룹 코드 ID는 자동 설정).
 3. 관리자가 상세 코드 값, 명칭, 설명, 정렬 순서, 사용 여부를 입력한다. (상세 코드 값 중복 확인 API 호출 가능)
 4. 관리자가 '등록' 버튼을 클릭한다.
 5. 프론트엔드는 입력 정보를 백엔드 API(`/api/v1/admin/codes/details?groupId=...`)에 POST 방식으로 전송한다.
 6. 백엔드는 값 중복 확인 후, 유효하면 새로운 상세 코드를 DB에 등록한다.
 7. 결과에 따라 프론트엔드에서 메시지 표시 및 후속 처리 수행.
- 호출 API 정보:
 - 상세 코드 값 중복 확인 API:
 - HTTP 메서드: GET
 - API 엔드포인트: `/api/v1/admin/codes/details/check-value?groupId={groupId}&codeValue={codeValue}`
 - 응답: `{"isAvailable": true/false}`
 - 상세 코드 등록 API:

- HTTP 메서드: **POST**
 - API 엔드포인트: **/api/v1/admin/codes/details**
 - 요청 파라미터: **groupId** (필수)
 - 요청 바디: **{ "codeValue": "string", "codeName": "string", "description": "string", "sortOrder": number, "isActive": boolean }**
 - 응답: **{ "success": true/false, "message": "string" }**
- Mockup 데이터 예시: 생략

예외 처리 및 유의 사항

- **그룹 코드 의존성:** 상세 코드는 반드시 특정 그룹 코드 하위에 존재해야 함. 그룹 코드 선택이 선행되어야 함.
- **삭제 정책:** 상세 코드가 시스템 로직(예: 과정 유형 분류)에 사용 중일 수 있으므로, 물리적 삭제보다는 '사용 여부' 비활성화 권장.
- **정렬 순서:** 정렬 순서 값을 통해 Select 박스 등 UI에 표시되는 순서를 제어할 수 있음.

시스템 설정 관리 화면 (SYS-SCR-003)

사용 역할자 : 시스템 관리자

관련 메뉴 : 시스템 관리 > 시스템 설정

기능 1: 시스템 설정 목록 조회 및 수정

- 관련 요구사항: SYS-REQ-004 (시스템 설정 관리 - UI 기반)
- 입력 정보:
 - (화면 로드 시 현재 설정값 조회)
 - **설정 항목별 입력 필드:**
 - 예: 사이트 제목 (텍스트 필드)
 - 예: 관리자 이메일 (텍스트 필드, 이메일 형식)
 - 예: 파일 업로드 최대 용량(MB) (숫자 입력 필드)
 - 예: 비밀번호 정책 (복합 설정: 최소 길이, 특수문자 포함 여부 등 체크박스/숫자 입력)
 - 예: 자동 로그아웃 시간(분) (숫자 입력 필드)
 - (기타 시스템 운영에 필요한 설정 항목들...)
 - **'저장' 버튼:** 변경된 설정값 저장 요청.
- 출력 정보:
 - 각 설정 항목명과 설명(Tooltip 등) 표시.
 - 입력 필드 옆에 현재 설정값 표시.
 - 저장 성공 시: "시스템 설정이 저장되었습니다." 메시지 표시.
 - 저장 실패 시 (유효성 오류, 서버 오류 등): 오류 메시지 표시.
- 프로세스 설명:
 1. 관리자가 시스템 설정 관리 화면에 진입한다.
 2. 프론트엔드는 현재 시스템 설정값을 조회하는 API(**/api/v1/admin/settings**)를 호출하여 각 항목의 값을 폼에 채운다.

3. 관리자가 필요한 설정값을 수정한다. 프론트엔드는 입력값에 대한 실시간 유효성 검사(숫자 범위, 형식 등)를 수행할 수 있다.
4. 관리자가 '저장' 버튼을 클릭한다.
5. 프론트엔드는 변경된 설정값들을 Key-Value 형태의 JSON으로 조합하여 백엔드 API(`/api/v1/admin/settings`)에 PUT 방식으로 전송한다.
6. 백엔드는 전달받은 설정값들을 DB 또는 설정 파일에 업데이트한다.
7. 업데이트 성공 시 성공 응답 반환, 실패 시 실패 응답 반환.
8. 프론트엔드는 결과에 따라 메시지를 표시한다. 적용된 설정은 시스템 동작에 즉시 또는 재시작 후 반영될 수 있다.

- 호출 API 정보:

- 시스템 설정 조회 API:

- HTTP 메서드: GET
 - API 엔드포인트: `/api/v1/admin/settings`
 - 응답 데이터 (예시):

```
{
  "siteTitle": "스포츠 E러닝 포털",
  "adminEmail": "admin@sports.or.kr",
  "maxFileUploadSizeMB": 1024,
  "passwordPolicy": { "minLength": 8, "requireSpecialChars": true },
  "sessionTimeoutMinutes": 30
  // ... other settings
}
```

- 시스템 설정 수정 API:

- HTTP 메서드: PUT
 - API 엔드포인트: `/api/v1/admin/settings`
 - 요청 바디: 조회 API 응답과 동일한 구조. 변경된 값을 담아 전송.
 - 응답: `{"success": true/false, "message": "string"}`

- Mockup 데이터 예시: 생략

예외 처리 및 유의 사항

- **설정 항목 정의:** 시스템 운영에 필요한 설정 항목들을 사전에 명확히 정의해야 함.
- **유효성 검증:** 각 설정값의 타입(문자, 숫자, 불리언 등) 및 유효 범위(예: 용량은 양수)에 대한 검증 필수.
- **적용 시점:** 설정 변경 사항이 시스템에 반영되는 시점(즉시, 다음 로그인 시, 서버 재시작 시 등)을 명확히 해야 함.
- **보안:** 민감한 설정(예: 외부 API 키)은 UI에서 직접 수정하지 않거나, 추가 인증 절차를 거치도록 고려.

권한 그룹 관리 화면 (SYS-SCR-004)

사용 역할자: 시스템 관리자

관련 메뉴: 시스템 관리 > 권한 관리 > 권한 그룹 관리

기능 1: 권한 그룹 목록 조회

- 관련 요구사항: SYS-REQ-040 (권한 그룹 생성 관리)
 - 입력 정보:
 - 검색 조건 (필요시): 그룹명
 - '검색' 버튼
 - 페이지네이션, 정렬
 - 출력 정보:
 - **권한 그룹 목록 테이블:** 그룹 ID, 그룹명, 설명, 등록일 등
 - 총 그룹 수, 페이지네이션 컴포넌트
 - 프로세스 설명: 코드 관리 목록 조회와 유사. API 엔드포인트만 다름.
 - 호출 API 정보:
 - HTTP 메서드: **GET**
 - API 엔드포인트: **/api/v1/admin/permissions/groups**
 - 파라미터: **page, size, sort, groupName**
 - 응답 형식: 코드 목록 조회와 유사.
-

기능 2: 권한 그룹 상세 조회/수정

- 관련 요구사항: SYS-REQ-040
 - 입력 정보:
 - 그룹 ID (읽기 전용)
 - 그룹명 (수정 가능)
 - 설명 (수정 가능)
 - '저장', '취소' 버튼
 - 출력 정보:
 - 저장 성공/실패 메시지
 - 프로세스 설명: 코드 상세 조회/수정과 유사. API 엔드포인트만 다름.
 - 호출 API 정보:
 - 상세 조회: **GET /api/v1/admin/permissions/groups/{groupId}**
 - 수정: **PUT /api/v1/admin/permissions/groups/{groupId}** (요청 바디: **{"groupName": "string", "description": "string"}**)
-

기능 3: 권한 그룹 신규 등록

- 관련 요구사항: SYS-REQ-040
- 입력 정보:
 - '신규 등록' 버튼
 - 그룹명 (필수)
 - 설명
 - '등록', '취소' 버튼
- 출력 정보:
 - 등록 성공/실패 메시지
- 프로세스 설명: 코드 신규 등록과 유사. API 엔드포인트만 다름. (ID는 자동 생성될 수 있음)
- 호출 API 정보:

- 등록: `POST /api/v1/admin/permissions/groups` (요청 바디: `{"groupName": "string", "description": "string"}`)

예외 처리 및 유의 사항

- **권한 그룹 역할:** 권한 그룹은 개별 권한(퍼미션)들을 논리적으로 묶어 관리하기 위한 용도. 역할(Role)에 이 그룹 단위로 권한을 부여할 수 있음. (구현 방식에 따라 퍼미션 자체를 관리하는 화면이 별도로 필요할 수도 있음)

역할 관리 화면 (SYS-SCR-005)

사용 역할자 : 시스템 관리자

관련 메뉴 : 시스템 관리 > 권한 관리 > 역할 관리

기능 1: 역할 목록 조회

- 관련 요구사항: SYS-REQ-005 (권한 관리), SYS-REQ-007 (관리자 역할 기반 접근 제어)
- 입력 정보:
 - 검색 조건 (필요시): 역할명
 - '검색' 버튼
 - 페이지네이션, 정렬
- 출력 정보:
 - **역할 목록 테이블:** 역할 ID(코드), 역할명, 설명, 사용 여부, 등록일 등
 - 총 역할 수, 페이지네이션 컴포넌트
- 프로세스 설명: 코드 관리 목록 조회와 유사.
- 호출 API 정보:
 - HTTP 메서드: `GET`
 - API 엔드포인트: `/api/v1/admin/roles`
 - 파라미터: `page`, `size`, `sort`, `roleName`
 - 응답 형식: 코드 목록 조회와 유사.

기능 2: 역할 상세 조회/수정

- 관련 요구사항: SYS-REQ-005, SYS-REQ-008 (사용자 역할별 권한 부여)
- 입력 정보:
 - 역할 ID (읽기 전용)
 - 역할명 (수정 가능)
 - 설명 (수정 가능)
 - 사용 여부 (수정 가능)
 - **권한 매핑 섹션:**
 - 사용 가능한 권한 목록 (트리 구조 또는 그룹별 목록). 권한 그룹(SYS-SCR-004) 또는 개별 퍼미션(Permission) 단위.
 - 현재 역할에 부여된 권한 목록.
 - 권한 추가/제거 버튼 또는 체크박스.

- '저장', '취소' 버튼
- 출력 정보:
 - 저장 성공/실패 메시지.
- 프로세스 설명:
 1. 관리자가 목록에서 역할을 선택하여 상세 조회 API 호출 (`GET /api/v1/admin/roles/{roleId}`). 응답에는 역할 정보와 함께 매핑된 권한 목록 포함.
 2. 전체 사용 가능한 권한 목록도 별도 API로 조회 (`GET /api/v1/admin/permissions`).
 3. 관리자가 역할명, 설명, 사용 여부를 수정하고, 권한 매핑 섹션에서 필요한 권한을 추가하거나 제거한다.
 4. '저장' 버튼 클릭 시, 역할 정보와 변경된 권한 목록을 백엔드 API (`PUT /api/v1/admin/roles/{roleId}`)로 전송.
 5. 백엔드는 역할 정보 및 역할-권한 매핑 정보를 업데이트한다.
 6. 결과에 따라 프론트엔드 처리.
- 호출 API 정보:
 - 상세 조회: `GET /api/v1/admin/roles/{roleId}` (연결된 권한 ID 목록 포함)
 - 전체 권한 목록 조회: `GET /api/v1/admin/permissions`
 - 수정: `PUT /api/v1/admin/roles/{roleId}` (요청 바디: `{"roleName": "string", "description": "string", "isActive": boolean, "permissionIds": ["perm1", "perm2", ...]}`)

기능 3: 역할 신규 등록

- 관련 요구사항: SYS-REQ-005
- 입력 정보:
 - '신규 등록' 버튼
 - 역할 ID (코드, 예: `ROLE_ADMIN`, `ROLE_INSTRUCTOR`). 필수. 중복 불가.
 - 역할명 (필수)
 - 설명
 - 사용 여부 (기본값 '예')
 - 권한 매핑 섹션 (등록 시 권한 바로 매핑 가능)
 - '등록', '취소' 버튼
- 출력 정보:
 - ID 중복 시 오류 메시지.
 - 등록 성공/실패 메시지.
- 프로세스 설명: 상세 조회/수정과 유사하게 권한 매핑 포함하여 등록 처리.
- 호출 API 정보:
 - ID 중복 확인: `GET /api/v1/admin/roles/check-id?roleId=...`
 - 등록: `POST /api/v1/admin/roles` (요청 바디: `{"roleId": "string", "roleName": "string", ..., "permissionIds": [...]}`)

예외 처리 및 유의 사항

- **역할 ID:** `ROLE_` 접두사를 사용하는 등 규칙을 정하는 것이 Spring Security 등 프레임워크 연동에 유리할 수 있음.
- **기본 역할:** 시스템 운영에 필수적인 기본 역할(예: 최고 관리자, 일반 사용자)은 삭제/수정 제한 필요.
- **권한 세분화:** 권한(Permission)을 얼마나 세분화할지(예: 기능 단위, CRUD 단위) 설계가 중요.

메뉴 관리 화면 (SYS-SCR-006)

사용 역할자 : 시스템 관리자

관련 메뉴 : 시스템 관리 > 권한 관리 > 메뉴 관리 (또는 시스템 관리 > UI 관리 > 메뉴 관리)

기능 1: 메뉴 계층 구조 조회 및 관리

- 관련 요구사항: SYS-REQ-012 (사용자 역할별 메뉴 구성 관리), SYS-REQ-035 (메뉴 등록), SYS-REQ-036 (메뉴 삭제), SYS-REQ-038 (메뉴 계층 관리 - 대/중/소)
- 입력 정보:
 - 사용자/관리자 구분 선택: 라디오 버튼 또는 탭 (사용자 포털 메뉴 / 관리자 시스템 메뉴 구분).
 - 메뉴 트리 영역:
 - 현재 메뉴 구조를 계층적 트리(Tree View) 형태로 표시.
 - 각 메뉴 노드 옆에 '추가'(하위 메뉴), '수정', '삭제' 버튼 표시.
 - 드래그 앤 드롭(Drag & Drop)으로 메뉴 순서 및 계층 변경 기능 (선택적 고급 기능).
 - '전체 저장' 버튼: 트리 구조 변경 사항 일괄 저장.
- 출력 정보:
 - 선택된 구분(사용자/관리자)에 해당하는 메뉴 트리가 시각적으로 표시됨.
 - 메뉴 노드 정보: 메뉴명, 메뉴 ID, 연결 URL, 사용 여부 등 (툴팁 또는 상세 영역에 표시).
- 프로세스 설명:
 1. 관리자가 메뉴 관리 화면에 진입하여 관리 대상(사용자/관리자)을 선택한다.
 2. 해당 메뉴 구조 전체가 트리 형태로 조회되어 표시된다 (API: `GET /api/v1/admin/menus?type=USER/ADMIN`).
 3. 관리자는 트리 상에서 다음 작업을 수행할 수 있다:
 - 순서 변경: 특정 메뉴 노드를 드래그하여 같은 레벨 내 다른 위치로 이동하거나 다른 메뉴의 하위로 이동.
 - 추가: 특정 메뉴 노드의 '추가' 버튼을 클릭하여 하위 메뉴 등록 폼 열기.
 - 수정: 특정 메뉴 노드의 '수정' 버튼을 클릭하여 해당 메뉴 정보 수정 폼 열기 (기능 2).
 - 삭제: 특정 메뉴 노드의 '삭제' 버튼을 클릭하여 삭제 확인 후 삭제 처리 (기능 3).
 4. 트리 구조 변경(순서/계층 이동) 후 '전체 저장' 버튼을 클릭하면, 변경된 트리 구조 정보(부모-자식 관계, 순서)를 백엔드 API (`PUT /api/v1/admin/menus/structure?type=USER/ADMIN`)로 전송하여 일괄 업데이트한다.
- 호출 API 정보:
 - 메뉴 트리 조회 API:
 - HTTP 메서드: `GET`
 - API 엔드포인트: `/api/v1/admin/menus`
 - 파라미터: `type=USER/ADMIN` (필수)
 - 응답 형식: 계층 구조를 나타내는 JSON (예: 재귀적인 노드 배열)

```
[
  {
    "menuId": "M001", "menuName": "개발원 소개", "url":
"/about", "order": 1, "isActive": true, "permissionId":
null,
```

```

    "children": [
      { "menuId": "M00101", "menuName": "인사말", "url":
"/about/greeting", "order": 1, "isActive": true,
"permissionId": null, "children": [] },
      // ... other children
    ]
  },
  // ... other top-level menus
]

```

- 메뉴 구조 업데이트 API:

- HTTP 메서드: PUT
- API 엔드포인트: /api/v1/admin/menus/structure
- 파라미터: type=USER/ADMIN (필수)
- 요청 바디: 변경된 전체 트리 구조 또는 변경된 노드의 부모/순서 정보 목록.
- 응답: {"success": true/false, "message": "string"}

- Mockup 데이터 예시: 생략

기능 2: 메뉴 정보 수정

- 관련 요구사항: SYS-REQ-012, SYS-REQ-037 (메뉴 권한 연동 설정)
- 입력 정보:
 - (트리에서 메뉴 노드의 '수정' 버튼 클릭 시) 해당 메뉴 정보가 채워진 폼 표시 (팝업 또는 우측 패널).
 - **메뉴 ID**: 텍스트 필드 (읽기 전용).
 - **메뉴명**: 텍스트 필드 (수정 가능). 필수.
 - **연결 URL**: 텍스트 필드 (수정 가능). 프론트엔드 라우터 경로 또는 외부 URL.
 - **메뉴 아이콘**: 아이콘 선택 컴포넌트 (선택 사항).
 - **사용 여부**: 라디오 버튼 또는 스위치.
 - **권한 연동**:
 - Select 박스 또는 검색 가능한 목록 (시스템 내 역할(Role) 또는 개별 권한(Permission) 목록).
 - 해당 메뉴 접근에 필요한 역할/권한 선택. (SYS-REQ-005, 008 연계) 선택 안함 시 모든 로그인 사용자 접근 가능(또는 비로그인 접근 가능 - 정책).
 - **'저장' 버튼**: 수정된 메뉴 정보 저장 요청.
 - **'취소' 버튼**: 변경 내용 취소.
- 출력 정보:
 - 저장 성공/실패 메시지. 성공 시 트리 뷰 업데이트.
- 프로세스 설명:
 1. 관리자가 트리에서 특정 메뉴의 '수정' 버튼을 클릭한다.
 2. 해당 메뉴의 상세 정보가 조회되어 폼에 표시된다 (API: GET /api/v1/admin/menus/{menuId}).
 3. 관리자가 메뉴명, URL, 사용 여부, 권한 연동 설정을 수정하고 '저장' 버튼을 클릭한다.
 4. 프론트엔드는 수정된 정보를 백엔드 API (PUT /api/v1/admin/menus/{menuId})로 전송한다.
 5. 백엔드는 메뉴 정보 및 연결된 권한 정보를 업데이트한다.
 6. 결과에 따라 프론트엔드 처리.
- 호출 API 정보:
 - 메뉴 상세 조회 API: GET /api/v1/admin/menus/{menuId}
 - 메뉴 수정 API: PUT /api/v1/admin/menus/{menuId}

- 요청 바디: {"menuName": "string", "url": "string", "icon": "string", "isActive": boolean, "requiredPermissionId": "string (or roleId)", "order": number (순서는 구조 업데이트 API로 처리할 수도 있음)}
 - 응답: {"success": true/false, "message": "string"}
 - 역할/권한 목록 조회 API (Select 박스용): GET /api/v1/admin/roles 또는 GET /api/v1/admin/permissions
-

기능 3: 메뉴 삭제

- 관련 요구사항: SYS-REQ-036 (메뉴 삭제)
 - 입력 정보:
 - 트리에서 메뉴 노드의 '삭제' 버튼 클릭.
 - **삭제 확인 팝업**: "하위 메뉴가 있는 경우 함께 삭제됩니다. 정말 삭제하시겠습니까?" 메시지 및 '확인', '취소' 버튼.
 - 출력 정보:
 - 삭제 성공 시: 메뉴 노드가 트리에서 제거됨.
 - 삭제 실패 시: 오류 메시지 표시.
 - 프로세스 설명:
 1. 관리자가 트리에서 특정 메뉴의 '삭제' 버튼을 클릭한다.
 2. 삭제 확인 팝업이 표시된다.
 3. 관리자가 '확인'을 클릭하면, 프론트엔드는 해당 메뉴 ID를 백엔드 API (**DELETE /api/v1/admin/menus/{menuId}**)로 전송한다.
 4. 백엔드는 해당 메뉴 및 하위 메뉴(재귀적으로)를 DB에서 삭제하거나 비활성화 처리한다. 메뉴-권한 매핑 정보도 함께 처리.
 5. 삭제 성공 시 성공 응답 반환, 실패 시 실패 응답 반환.
 6. 프론트엔드는 성공 시 트리 뷰에서 해당 노드를 제거한다.
 - 호출 API 정보:
 - **메뉴 삭제 API**: **DELETE /api/v1/admin/menus/{menuId}**
 - 응답: {"success": true/false, "message": "string"}
-

기능 4: 메뉴 신규 등록

- 관련 요구사항: SYS-REQ-035 (메뉴 등록)
- 입력 정보:
 - (트리에서 특정 메뉴의 '추가' 버튼 클릭 시) 해당 메뉴 하위에 등록하기 위한 폼 표시.
 - **부모 메뉴 ID/명**: 표시 (읽기 전용).
 - **메뉴 ID**: 텍스트 필드 (자동 생성 또는 직접 입력 - 정책). 필수. 중복 불가.
 - **메뉴명**: 텍스트 필드. 필수.
 - **연결 URL**: 텍스트 필드.
 - **메뉴 아이콘**: 아이콘 선택 컴포넌트.
 - **사용 여부**: 라디오 버튼/스위치. 기본값 '예'.
 - **권한 연동**: 역할/권한 선택.
 - **'등록' 버튼**: 신규 메뉴 정보 등록 요청.
 - **'취소' 버튼**: 입력 내용 취소.
- 출력 정보:

- ID 중복 시 오류 메시지.
- 등록 성공 시: 새로운 메뉴 노드가 트리에 추가됨.
- 등록 실패 시: 오류 메시지.
- 프로세스 설명:
 1. 관리자가 트리에서 특정 메뉴(부모)의 '추가' 버튼을 클릭한다.
 2. 신규 메뉴 등록 폼이 나타난다 (부모 메뉴 정보 포함).
 3. 관리자가 새 메뉴 정보를 입력하고 '등록' 버튼을 클릭한다.
 4. 프론트엔드는 입력 정보와 부모 메뉴 ID를 백엔드 API (`POST /api/v1/admin/menus`)로 전송한다.
 5. 백엔드는 ID 중복 확인 후, 새 메뉴를 DB에 등록하고 부모-자식 관계 및 순서(기본적으로 마지막)를 설정한다.
 6. 결과에 따라 프론트엔드 처리 (트리 뷰 업데이트).
- 호출 API 정보:
 - 메뉴 등록 API: `POST /api/v1/admin/menus`
 - 요청 바디: `{"parentId": "string", "menuId": "string", "menuName": "string", "url": "string", ...}`
 - 응답: `{"success": true/false, "message": "string", "newMenuNode": {...}}` (등록된 메뉴 정보 반환)

예외 처리 및 유의 사항

- 메뉴 권한 연동: 메뉴에 설정된 권한(역할 또는 퍼미션)에 따라 사용자에게 해당 메뉴가 보이거나 접근이 제어되어야 함 (SYS-REQ-009, 039). 프론트엔드/백엔드 모두 권한 체크 로직 필요.
- 레이아웃 관리 연동: SYS-REQ-013(역할별 레이아웃 관리) 요구사항과 연계하여, 특정 역할에게 다른 메뉴 구조(레이아웃)를 제공하는 기능까지 확장 고려 가능. (복잡도 높음)
- 캐싱: 메뉴 구조는 자주 변경되지 않으므로, 조회 성능 향상을 위해 캐싱 전략 적용 고려.

교육장 시설 관리 화면 (SYS-SCR-007)

사용 역할자: 시스템 관리자

관련 메뉴: 시스템 관리 > 교육 과정 관리 > 교육장 시설 관리 (또는 별도 시설 관리 메뉴)

기능 1: 교육장 시설 목록 조회

- 관련 요구사항: SYS-REQ-017 (교육장 시설 기본 검색), SYS-REQ-018 (교육장 시설 상세 검색)
- 입력 정보:
 - 검색 조건:
 - 교육장명: 텍스트 필드.
 - 위치(주소): 텍스트 필드.
 - 사용 가능 여부: Select 박스 (전체, 예, 아니오).
 - '검색' 버튼: 조건에 맞는 시설 목록 조회 요청.
 - 페이지네이션, 정렬: 표준 목록 기능.
- 출력 정보:
 - 교육장 시설 목록 테이블:

- 컬럼: 체크박스, 번호(No), 교육장 ID, 교육장명, 위치(주소), 연락처, 사용 가능 여부, 등록일.
 - 교육장명 클릭 시 상세 조회/수정 화면 이동.
- 총 시설 수, 페이지네이션 컴포넌트.
- 프로세스 설명: 표준적인 관리자 목록 조회 기능.
- 호출 API 정보:
 - HTTP 메서드: **GET**
 - API 엔드포인트: **/api/v1/admin/facilities**
 - 파라미터: **page, size, sort, name, location, isAvailable**
 - 응답 형식: 표준 페이징 목록 응답.

기능 2: 교육장 시설 상세 조회/수정

- 관련 요구사항: SYS-REQ-014 (교육장 시설 정보 등록), SYS-REQ-015 (교육장별 교육 시설 소개 등록)
- 입력 정보:
 - (목록에서 교육장명 클릭 시) 해당 시설 정보가 채워진 폼 표시.
 - **교육장 ID**: 텍스트 필드 (읽기 전용 또는 자동 생성).
 - **교육장명**: 텍스트 필드. 필수.
 - **위치 (주소)**: 텍스트 필드 (주소 검색 API 연동 가능). 필수.
 - **상세 주소**: 텍스트 필드.
 - **연락처**: 텍스트 필드.
 - **수용 가능 인원**: 숫자 입력 필드.
 - **시설 소개**: 텍스트 영역 (Rich Text Editor 사용 가능). (SYS-REQ-015)
 - **찾아오시는 길 설명**: 텍스트 영역.
 - **지도 정보**: 위도/경도 입력 또는 지도 API 연동 마커 표시.
 - **시설 이미지**: 파일 업로드 컴포넌트 (다중 업로드 가능).
 - **사용 가능 여부**: 라디오 버튼/스위치.
 - '저장' 버튼, '취소' 버튼.
- 출력 정보:
 - 저장 성공/실패 메시지.
 - 업로드된 이미지 미리보기.
- 프로세스 설명:
 1. 관리자가 목록에서 시설명을 클릭하여 상세 조회 API 호출 (**GET /api/v1/admin/facilities/{facilityId}**).
 2. 폼에 정보가 표시되면 관리자가 수정하고 '저장' 버튼 클릭.
 3. 프론트엔드는 수정된 정보(이미지 파일 포함 시 FormData 사용)를 백엔드 API (**PUT /api/v1/admin/facilities/{facilityId}**)로 전송.
 4. 백엔드는 정보 업데이트 및 이미지 파일 저장 처리.
 5. 결과에 따라 프론트엔드 처리.
- 호출 API 정보:
 - 상세 조회: **GET /api/v1/admin/facilities/{facilityId}**
 - 수정: **PUT /api/v1/admin/facilities/{facilityId}** (Content-Type: **multipart/form-data** 또는 **application/json**)
 - 요청 바디: 시설 정보 필드 및 이미지 파일(들).
 - 응답: **{"success": true/false, "message": "string"}**

기능 3: 교육장 시설 신규 등록

- 관련 요구사항: SYS-REQ-014, SYS-REQ-015
- 입력 정보:
 - '신규 등록' 버튼.
 - 상세 조회/수정 폼과 동일한 입력 필드 (교육장 ID는 자동 생성될 수 있음).
 - '등록' 버튼, '취소' 버튼.
- 출력 정보:
 - 등록 성공/실패 메시지.
- 프로세스 설명: 상세 조회/수정과 유사한 플로우로 신규 등록 처리.
- 호출 API 정보:
 - 등록: `POST /api/v1/admin/facilities` (Content-Type: `multipart/form-data` 또는 `application/json`)
 - 요청 바디: 시설 정보 필드 및 이미지 파일(들).
 - 응답: `{"success": true/false, "message": "string", "facilityId": "..."} (생성된 ID 반환)`

예외 처리 및 유의 사항

- **과정 연계:** 등록된 교육장 정보는 오프라인/혼합 교육 과정 개설 시 장소 선택 항목으로 사용되어야 함 (SYS-REQ-016). COU 모듈과의 연계 필요.
- **시설 대관 연동:** 이 시설 정보는 사용자 포털의 시설 대관 기능(POR-REQ-041~045)과도 연동되어야 함. '대관 가능 여부' 등 추가 속성 필요 가능성.
- **이미지 처리:** 업로드된 이미지는 리사이징, 썸네일 생성 등 처리 필요.

결제/정산/환불 관리 화면 (SYS-SCR-008)

사용 역할자: 시스템 관리자, 재무 담당자

관련 메뉴: 시스템 관리 > 결제 관리 > 결제 내역 조회 (및 하위 메뉴: 취소 내역, 환불 내역, 정산 관리)

기능 1: 결제 내역 조회

- 관련 요구사항: SYS-REQ-023 (수강 신청 결제 내역 확인)
- 입력 정보:
 - 검색 조건:
 - 기간: 날짜 범위 선택 (결제일 기준).
 - 과정명: 텍스트 입력 필드.
 - 결제자명/ID: 텍스트 입력 필드.
 - 결제 상태: Select 박스 (전체, 결제 완료, 결제 실패, 부분 취소, 전체 취소 등).
 - 결제 수단: Select 박스 (전체, 카드, 계좌이체, 간편결제 등 - SYS-REQ-019).
 - '검색' 버튼, 페이지네이션, 정렬.
- 출력 정보:

- **결제 내역 목록 테이블:**
 - 컬럼: 결제 번호, 결제일시, 과정명, 결제자명(ID), 결제 금액, 결제 수단, 결제 상태, PG사 거래 ID 등.
 - 결제 번호 클릭 시 상세 정보 팝업 표시 (기능 2).
 - 총 결제 건수 및 금액 요약 정보.
 - 페이지네이션 컴포넌트.
 - 프로세스 설명: 표준적인 관리자 목록 조회 기능. PG사 연동 데이터 포함 조회.
 - 호출 API 정보:
 - HTTP 메서드: **GET**
 - API 엔드포인트: **/api/v1/admin/payments**
 - 파라미터: **page, size, sort, startDate, endDate, courseName, userName, status, paymentMethod**
 - 응답 형식: 표준 페이징 목록 응답 (결제 상세 정보 포함).
-

기능 2: 결제 상세 정보 조회

- 관련 요구사항: SYS-REQ-023
 - 입력 정보: (목록에서 결제 번호 클릭 시)
 - 출력 정보:
 - **결제 상세 팝업:**
 - 기본 결제 정보 (결제 번호, 일시, 금액, 수단, 상태 등).
 - 구매자 정보 (이름, ID, 연락처).
 - 과정 정보 (과정명, 기수).
 - PG사 정보 (거래 ID, 승인 번호 등).
 - 결제/취소/환불 이력 타임라인.
 - (필요시) '영수증 보기' 버튼 (PG사 제공 페이지 링크).
 - (필요시) '결제 취소' 버튼 (기능 3).
 - 프로세스 설명: 특정 결제 건의 상세 정보를 조회하여 팝업으로 표시.
 - 호출 API 정보:
 - HTTP 메서드: **GET**
 - API 엔드포인트: **/api/v1/admin/payments/{paymentId}**
 - 응답 형식: 상세 결제 정보를 담은 JSON 객체.
-

기능 3: 결제 취소 (환불) 처리

- 관련 요구사항: SYS-REQ-021 (조건 기반 자동 환불 처리 - 여기서는 관리자 수동 처리), SYS-REQ-022 (관리자 환불 처리 조건 수정 - 별도 설정 화면 필요 가능), SYS-REQ-024 (수강 신청 취소 내역 확인), SYS-REQ-025 (수강 신청 환불 내역 확인)
- 입력 정보:
 - (결제 상세 팝업에서 '결제 취소' 버튼 클릭 시)
 - **취소/환불 금액:** 숫자 입력 필드 (전체 취소 시 자동 계산, 부분 취소 시 직접 입력).
 - **취소 사유:** 텍스트 영역.
 - **'취소 실행' 버튼.**
- 출력 정보:
 - 취소 처리 성공 시: "결제 취소(환불) 처리가 완료되었습니다." 메시지 표시 및 상세 정보/목록 업데이트.
 - 취소 처리 실패 시 (PG사 통신 오류, 이미 취소된 건 등): 오류 메시지 표시.

- 프로세스 설명:
 1. 관리자가 결제 상세 팝업에서 '결제 취소' 버튼을 클릭한다.
 2. 취소 금액(부분/전체) 및 사유를 입력하고 '취소 실행' 버튼을 클릭한다.
 3. 프론트엔드는 취소 요청 정보를 백엔드 API (**POST** `/api/v1/admin/payments/{paymentId}/cancel`)로 전송한다.
 4. 백엔드는 해당 결제 건 정보를 확인하고, PG사 결제 취소 API를 호출한다.
 5. PG사로부터 취소 성공/실패 결과를 받아 시스템 DB의 결제 상태 및 환불 정보를 업데이트한다. (자동 환불 조건(SYS-REQ-021) 로직은 시스템 설정 및 과정 정보와 연계되어 백엔드에서 처리)
 6. 결과에 따라 프론트엔드에 응답을 반환한다.
 7. 프론트엔드는 메시지를 표시하고 화면을 갱신한다. (취소/환불 내역은 별도 메뉴 또는 동일 화면 필터링으로 조회 가능 - SYS-REQ-024, 025)
- 호출 API 정보:
 - **결제 취소/환불 API:**
 - HTTP 메서드: **POST**
 - API 엔드포인트: `/api/v1/admin/payments/{paymentId}/cancel`
 - 요청 바디: `{"cancelAmount": number, "reason": "string"}`
 - 응답: `{"success": true/false, "message": "string"}`

기능 4: 정산 관리 (개념)

- 관련 요구사항: SYS-REQ-020 (비용 정산 처리 관리 - 실시 기관/PG사 정산 포함)
- 화면 구성 아이디어:
 - 기간별/PG사별/과정별 정산 내역 조회 화면.
 - PG사로부터 받은 정산 데이터 업로드 또는 API 연동 조회 기능.
 - 시스템 내 결제/환불 데이터와 대사 기능.
 - 정산 완료 처리 기능.
- 참고: 이 기능은 복잡도가 높고 외부 시스템 연동이 필수적이므로, 별도 상세 설계 및 전문 솔루션 연동 고려 필요. 화면 설계는 기본 조회 기능 위주로 우선 고려.

기능 5: 계산서 발행/관리 (개념)

- 관련 요구사항: SYS-REQ-026 (계산서 발행 처리), SYS-REQ-027 (계산서 발행 내역 관리)
- 화면 구성 아이디어:
 - 계산서 발행 요청 목록 조회 화면 (사용자 요청 또는 특정 조건).
 - 계산서 정보 입력 및 발행 실행 기능 (외부 발행 서비스 연동).
 - 발행된 계산서 내역 조회 및 상태 관리 화면.
- 참고: 국세청 연동 등 외부 서비스 연동이 필요하며, 별도 상세 설계 필요.

예외 처리 및 유의 사항

- **PG사 연동:** 모든 결제/취소 처리는 PG사 API 연동이 필수적. 연동 오류, 통신 실패, 타임아웃 등에 대한 예외 처리 및 로깅 중요.
- **데이터 정합성:** 시스템 DB의 결제 상태와 PG사의 실제 처리 상태 간의 정합성 유지 방안 필요 (Webhook 수신 등).
- **환불 정책:** 시스템 설정(SYS-REQ-022) 및 과정별 정책에 따른 환불 가능 여부 및 금액 계산 로직 명확화 필요.
- **보안:** 민감한 결제 정보 접근 및 처리 권한 관리 철저.

다음으로 커뮤니티/콘텐츠(게시판, 팝업, 약관 등) 관리 화면들을 설계하겠습니다.

게시판 관리 화면 (SYS-SCR-009)

사용 역할자 : 시스템 관리자, 운영자

관련 메뉴 : 시스템 관리 > 콘텐츠 관리 > 게시판 관리 (또는 커뮤니티 관리 > 게시판 관리)

기능 1: 게시판 마스터 관리 (게시판 생성/수정)

- 관련 요구사항: SYS-REQ-028 (게시판 관리 - 갤러리형, 리스트형 등), POR-REQ-031 (포털 게시판 분류 관리), POR-REQ-032 (포털 게시판 마스터 관리 - 금치어 설정 등)
- 입력 정보:
 - '게시판 생성' 버튼.
 - 게시판 목록: 시스템 내 생성된 게시판 리스트 (ID, 이름, 유형, 사용 여부 등). 클릭 시 수정 폼 로드.
 - 게시판 설정 폼:
 - 게시판 ID (자동 생성 또는 직접 입력).
 - 게시판명 (필수).
 - 게시판 설명.
 - 게시판 유형: Select 박스 (리스트형, 갤러리형, FAQ형 등).
 - 사용 여부: 스위치.
 - 접근 권한: Select 박스 (전체, 로그인 사용자, 특정 역할 등).
 - 쓰기 권한: Select 박스 (관리자만, 특정 역할 이상, 로그인 사용자 등).
 - 댓글 사용 여부: 스위치.
 - 파일 첨부 사용 여부: 스위치. (최대 용량/개수 설정 추가 가능).
 - 금치어 설정: 텍스트 영역 (coma 등으로 구분 입력). (POR-REQ-032)
 - '저장' 버튼.
- 출력 정보:
 - 생성/수정 성공 또는 실패 메시지.
 - 게시판 목록 업데이트.
- 프로세스 설명:
 1. 관리자가 게시판 관리 화면에 진입하여 기존 게시판 목록을 확인하거나 '게시판 생성' 버튼을 클릭한다.
 2. 생성 또는 수정을 위해 게시판 설정 폼에 정보를 입력/수정한다.
 3. '저장' 버튼을 클릭하면 백엔드 API (`POST /api/v1/admin/boards` 또는 `PUT /api/v1/admin/boards/{boardId}`)를 호출하여 정보를 저장한다.
 4. 백엔드는 게시판 마스터 정보를 생성하거나 업데이트한다.
 5. 결과에 따라 프론트엔드 처리.
- 호출 API 정보:
 - 목록 조회: `GET /api/v1/admin/boards`
 - 상세 조회: `GET /api/v1/admin/boards/{boardId}`
 - 생성: `POST /api/v1/admin/boards`
 - 수정: `PUT /api/v1/admin/boards/{boardId}`
 - (삭제 API는 일반적으로 비활성화 처리로 대체)

기능 2: 게시물 관리 (게시판별)

- 관련 요구사항: SYS-REQ-028, SYS-REQ-030 (공지 사항 관리), SYS-REQ-031 (FAQ 관리) 등 개별 게시판 관리 요구사항 포괄.
- 입력 정보:
 - **게시판 선택**: Select 박스 (기능 1에서 관리되는 게시판 목록). 필수.
 - **검색 조건**: 제목, 내용, 작성자, 기간 등.
 - **'검색' 버튼, 페이지네이션, 정렬**.
 - **'글쓰기' 버튼** (선택된 게시판으로 이동 또는 팝업).
 - 목록 내 **'삭제', '수정', '답글' (FAQ 등) 버튼**.
- 출력 정보:
 - **게시물 목록 테이블**:
 - 컬럼: 체크박스, 번호(No), 제목(클릭 시 상세 보기), 작성자, 작성일, 조회수, 첨부파일 여부, 공지 여부 등 (게시판 유형 따라 가변적).
 - 총 게시물 수, 페이지네이션 컴포넌트.
- 프로세스 설명:
 1. 관리자가 관리할 게시판을 선택한다.
 2. 해당 게시판의 게시물 목록이 조회된다 (`GET /api/v1/admin/posts?boardId=...`).
 3. 관리자는 목록에서 게시물을 검색, 조회, 수정, 삭제, 또는 새로 작성할 수 있다.
 4. 게시물 수정/삭제/등록은 별도 화면(SYS-SCR-010) 또는 팝업에서 이루어진다.
- 호출 API 정보:
 - 게시물 목록 조회: `GET /api/v1/admin/posts` (boardId 필수, 검색 파라미터 포함)
 - 게시물 삭제: `DELETE /api/v1/admin/posts/{postId}` (다중 선택 삭제 가능)

예외 처리 및 유의 사항

- **게시판 유형별 기능**: 게시판 유형(리스트, 갤러리, FAQ)에 따라 목록 표시 방식 및 관리 기능이 일부 달라질 수 있음 (예: FAQ는 질문/답변 쌍 관리).
- **권한**: 관리자/운영자는 모든 게시물에 대한 관리 권한을 가짐.
- **콘텐츠 보안**: 게시물 내용 중 개인 정보, 비속어 등 필터링 기능 고려.

게시물 작성/수정 화면 (SYS-SCR-010)

사용 역할자 : 시스템 관리자, 운영자 (또는 글쓰기 권한 있는 사용자)

관련 메뉴 : 게시판 관리 화면에서 '글쓰기' 또는 '수정' 클릭 시

기능 1: 게시물 내용 작성/수정

- 관련 요구사항: SYS-REQ-028 등 게시판 관리 관련
- 입력 정보:
 - **게시판명**: 표시 (읽기 전용).
 - **제목**: 텍스트 필드. 필수.

- **내용:** Rich Text Editor 컴포넌트 (CKEditor, Quill 등). 필수. 이미지 업로드 지원.
- **작성자:** 표시 (관리자는 수정 가능할 수도 있음).
- **공지글 설정:** 체크박스 (관리자만). 최상단 고정 기능.
- **비밀글 설정:** 체크박스 (필요시).
- **첨부 파일:** 파일 업로드 컴포넌트 (다중 첨부, 용량/개수 제한).
- **'저장' 버튼, '취소' 버튼.**
- **출력 정보:**
 - 에디터 UI, 파일 목록 등.
 - 저장 성공/실패 메시지. 성공 시 목록 화면으로 이동.
- **프로세스 설명:**
 1. 관리자가 '글쓰기' 또는 '수정'(이 경우 기존 내용 로드)을 통해 해당 화면 진입.
 2. 제목, 내용, 첨부 파일 등 필요한 정보를 입력/수정한다.
 3. '저장' 버튼 클릭 시, 입력 데이터를 백엔드 API (**POST /api/v1/admin/posts** 또는 **PUT /api/v1/admin/posts/{postId}**)로 전송 (파일 포함 시 FormData).
 4. 백엔드는 게시물 정보를 저장/업데이트하고 첨부 파일을 처리한다.
 5. 결과에 따라 프론트엔드 처리.
- **호출 API 정보:**
 - 상세 조회 (수정 시): **GET /api/v1/admin/posts/{postId}**
 - 등록: **POST /api/v1/admin/posts**
 - 수정: **PUT /api/v1/admin/posts/{postId}**
 - (파일 업로드는 별도 API 또는 통합 처리 가능)

예외 처리 및 유의 사항

- **에디터 설정:** Rich Text Editor의 기능(툴바 옵션, 이미지 업로드 방식 등) 설정 필요.
- **파일 관리:** 첨부 파일 저장 정책(DB 또는 파일 시스템), 보안(악성 파일 검사) 고려.
- **XSS 방지:** 사용자 입력(특히 HTML 에디터 내용)에 대한 XSS(Cross-Site Scripting) 방지 처리 필수.

SYS 모듈의 나머지 관리 기능(팝업, 약관, SMS 등) 화면들도 위 게시판 관리와 유사한 CRUD 형태로 설계될 수 있습니다. 각 기능의 특성에 맞는 필드와 API를 정의하면 됩니다.

예시:

- **팝업 관리(SYS-SCR-011):** 목록(제목, 게시 기간, 사용 여부), 등록/수정(제목, 내용(이미지/텍스트), 링크 URL, 게시 시작/종료일, 크기/위치, 사용 여부).
- **약관 관리(SYS-SCR-012):** 목록(약관 종류, 버전, 시행일), 등록/수정(약관 종류 선택, 내용(에디터), 버전, 시행일).
- **SMS 발송 관리(SYS-SCR-013):** 발송 이력 조회(기간, 수신자, 내용, 상태), 수동 발송(수신자 선택(그룹/개별), 메시지 입력(템플릿 사용 가능), 예약 발송 여부), 자동 발송 설정(상황별(예: 회원가입 완료, 비밀번호 변경) 템플릿 및 발송 여부 설정).

대용량 콘텐츠 업로드 화면 (SYS-SCR-017)

사용 역할자: 시스템 관리자, 콘텐츠 관리자

관련 메뉴 : 시스템 관리 > 콘텐츠 관리 > 콘텐츠 업로드 (또는 과정 관리 내 콘텐츠 업로드 연계)

기능 1: 콘텐츠 파일 선택 및 업로드 실행

- 관련 요구사항: SYS-REQ-001 (대용량 학습 콘텐츠 업로드)
- 입력 정보:
 - **업로드 방식 선택:** 라디오 버튼 (개별 파일 업로드 / 폴더 업로드 / FTP 등 다른 방식 - 초기엔 파일/폴더 위주).
 - **파일 선택 영역:**
 - '파일 선택' 버튼 클릭 또는 드래그 앤 드롭 영역.
 - 선택된 파일 목록 표시 (파일명, 크기).
 - **(선택) 업로드 경로 지정:** 콘텐츠가 저장될 서버 내 경로 또는 카테고리 지정 (과정 연계 시 자동 지정될 수도 있음).
 - **'업로드 시작' 버튼.**
- 출력 정보:
 - 선택된 파일 목록 및 총 개수, 총 크기 표시.
 - 업로드 진행률 표시 (전체 및 개별 파일 - 기능 2 연계).
 - 업로드 성공/실패 메시지.
- 프로세스 설명:
 1. 관리자가 업로드할 콘텐츠 파일(동영상, 문서 등)을 선택한다 (개별 또는 폴더 단위).
 2. 선택된 파일 목록이 화면에 표시된다.
 3. 관리자가 '업로드 시작' 버튼을 클릭한다.
 4. 프론트엔드는 대용량 파일 업로드 라이브러리(tus.io, resumable.js 등 고려)를 사용하여 파일을 청크 (Chunk) 단위로 분할하여 백엔드 업로드 API (**POST /api/v1/admin/contents/upload**)로 전송 시작한다. (이어 올리기, 병렬 처리 지원)
 5. 백엔드는 수신된 청크를 임시 저장하고, 전체 파일 수신 완료 시 최종 파일로 병합하여 지정된 스토리지(S3, NAS 등)에 저장한다. 파일 메타데이터(파일명, 크기, 경로, 업로더 등)는 DB에 기록.
 6. 업로드 과정 중 진행 상태(%)를 프론트엔드로 주기적으로 전달 (기능 2).
 7. 전체 업로드 완료 또는 실패 시 최종 결과를 프론트엔드에 알린다.
 8. 프론트엔드는 결과 메시지를 표시하고 업로드 목록 상태를 갱신한다.
- 호출 API 정보:
 - **콘텐츠 업로드 API:** (대용량 업로드 프로토콜 사용)
 - HTTP 메서드: **POST, PATCH** 등 (프로토콜에 따라 다름)
 - API 엔드포인트: **/api/v1/admin/contents/upload**
 - 요청 데이터: 파일 청크 데이터 및 관련 메타데이터 (파일명, 전체 크기, 청크 번호 등).
 - 응답 데이터: 업로드된 바이트 수, 다음 청크 요청 정보, 최종 완료 시 파일 정보 등.

기능 2: 업로드 진행 상태 확인

- 관련 요구사항: SYS-REQ-002 (콘텐츠 파일 업로드 진행 상태 확인)
- 입력 정보: (기능 1의 업로드 실행 중)
- 출력 정보:
 - **업로드 목록 영역:**
 - 각 파일별 진행률 바(Progress bar) 및 상태 표시 (대기중, 업로드중, 처리중, 완료, 실패).
 - 전체 업로드 진행률 바 및 예상 남은 시간 표시.

- **(선택) 업로드 중단/재개 버튼:** 개별 파일 또는 전체 업로드 제어.
- **오류 발생 시:** 해당 파일에 오류 아이콘 및 메시지 표시 (예: '용량 초과', '네트워크 오류').
- 프로세스 설명:
 1. 파일 업로드가 시작되면, 프론트엔드는 업로드 라이브러리 또는 백엔드 API(상태 조회용)를 통해 각 파일 및 전체의 진행 상태 정보를 주기적으로 받는다.
 2. 수신된 정보를 바탕으로 화면의 진행률 바와 상태 텍스트를 실시간으로 업데이트한다.
 3. 업로드 중 오류가 발생하면 해당 파일의 상태를 '실패'로 변경하고 오류 메시지를 표시한다.
 4. 관리자는 진행 상황을 시각적으로 확인하고, 필요시 업로드를 중단하거나 실패한 항목을 재시도할 수 있다 (재시도 기능은 프로토콜 지원 시).
- 호출 API 정보: (업로드 API 응답 또는 별도 상태 조회 API)
 - 예: `GET /api/v1/admin/contents/upload/status?uploadId=...` (필요시)

예외 처리 및 유의 사항

- **대용량 처리:** 웹서버 타임아웃 방지, 안정적인 전송(이어 올리기), 서버 자원 관리(메모리, 디스크) 등 대용량 파일 처리에 특화된 기술 및 아키텍처 고려 필수.
- **보안:** 업로드 파일에 대한 악성코드 검사, 허용된 파일 확장자/MIME 타입 검증 필요.
- **후처리:** 업로드 완료 후 동영상 인코딩, 썸네일 추출 등 필요한 후처리 작업은 비동기적으로 수행되어야 함 (별도 작업 큐 및 워커 필요).
- **과정 연계:** 업로드된 콘텐츠를 특정 교육 과정의 학습 자료로 연결하는 기능 필요 (COU 또는 LRN 모듈 화면에서 처리).

환불 정책/조건 관리 화면 (SYS-SCR-018)

사용 역할자 : 시스템 관리자, 재무 담당자

관련 메뉴 : 시스템 관리 > 결제 관리 > 환불 정책 관리

기능 1: 환불 조건 설정 조회 및 수정

- 관련 요구사항: SYS-REQ-021 (조건 기반 자동 환불 처리), SYS-REQ-022 (관리자 환불 처리 조건 수정)
- 입력 정보:
 - (화면 로드 시 현재 설정된 환불 조건 조회)
 - **환불 조건 설정 영역:**
 - **기간별 환불율 설정:** (예: 표 형태)
 - 조건 기준: Select 박스 (학습 시작일 기준 / 결제일 기준).
 - 기간 설정: 숫자 입력 필드 (예: 시작 전, 시작 후 1일 이내, 시작 후 7일 이내 등).
 - 환불율(%): 숫자 입력 필드 (0~100).
 - '+' 버튼으로 기간 조건 추가 가능.
 - **진도율 기준 환불 설정:** (예: 혼합 사용 가능)
 - 조건 기준: 학습 진도율 (%).
 - 진도율 구간 설정: 숫자 입력 필드 (예: 0% 초과 30% 이하).
 - 환불율(%): 숫자 입력 필드.
 - '+' 버튼으로 구간 추가 가능.

- 기타 조건: (예: 특정 과정 유형별 예외 조건 설정 - 선택적).
 - 환불 불가 조건 명시: 텍스트 영역 (예: 수료증 발급 후 환불 불가).
- '저장' 버튼.
- 출력 정보:
 - 현재 설정된 환불 조건 테이블 또는 리스트 표시.
 - 입력 오류 시 (예: 기간 중첩, 환불율 범위 초과) 메시지 표시.
 - 저장 성공/실패 메시지.
- 프로세스 설명:
 1. 관리자가 환불 정책 관리 화면에 진입하여 현재 설정된 조건을 확인한다 (GET `/api/v1/admin/refund-policies`).
 2. 관리자가 기간별/진도율별 환불율 등 조건을 수정하거나 추가/삭제한다.
 3. '저장' 버튼 클릭 시, 설정된 조건 데이터를 백엔드 API (PUT `/api/v1/admin/refund-policies`)로 전송한다.
 4. 백엔드는 환불 정책 정보를 업데이트한다. 이 정보는 실제 환불 처리 로직(자동/수동)에서 참조된다.
 5. 결과에 따라 프론트엔드 처리.
- 호출 API 정보:
 - 조회: GET `/api/v1/admin/refund-policies`
 - 수정: PUT `/api/v1/admin/refund-policies`
 - 요청 바디: 환불 조건 구조를 담은 JSON (예: 기간 조건 배열, 진도율 조건 배열 등)
 - 응답: `{"success": true/false, "message": "string"}`

예외 처리 및 유의 사항

- **정책 복잡성:** 환불 정책은 법규(평생교육법 등)를 준수해야 하며, 조건이 복잡해질 수 있으므로 명확하고 유연한 설정 방식 제공 필요.
- **적용 시점:** 정책 변경 시 기존 신청 건에 소급 적용할지, 변경 이후 신청 건부터 적용할지 정책 정의 필요.
- **자동 환불 연계:** 여기서 설정된 조건은 백엔드의 자동 환불 처리 로직(SYS-REQ-021)에 직접 반영되어야 함.

정산 관리 화면 (SYS-SCR-019)

사용 역할자 : 시스템 관리자, 재무 담당자

관련 메뉴 : 시스템 관리 > 결제 관리 > 정산 관리

기능 1: 정산 내역 조회 (PG사 기준)

- 관련 요구사항: SYS-REQ-020 (비용 정산 처리 관리 - PG사 정산 포함)
- 입력 정보:
 - 검색 조건:
 - 기간: 날짜 범위 선택 (정산일 또는 거래일 기준).
 - PG사: Select 박스 (연동된 PG사 목록).
 - 정산 상태: Select 박스 (전체, 정산 완료, 대사 필요 등).
 - '검색' 버튼, 페이지네이션, 정렬.
 - (선택) 정산 데이터 업로드 버튼: PG사에서 받은 정산 파일(CSV, Excel) 업로드 기능.

- 출력 정보:
 - 정산 내역 목록 테이블:
 - 컬럼: 정산일, PG사, 거래 건수, 총 거래 금액, 수수료, 정산 금액, 상태, 관련 내부 데이터(결제 내역) 링크 등 (PG사 제공 데이터 기준).
 - 기간별 총 정산 금액 요약.
 - 페이지네이션 컴포넌트.
 - 프로세스 설명: PG사로부터 받은 정산 내역을 조회하거나 업로드하여 관리하는 기능.
 - 관리자가 기간, PG사 등 조건을 선택하고 '검색' 버튼을 클릭한다.
 - 백엔드는 저장된 정산 데이터 또는 PG사 API 연동을 통해(구현 방식에 따라) 해당 내역을 조회하여 반환 (`GET /api/v1/admin/settlements/pg`).
 - (파일 업로드 시) 관리자가 정산 파일을 업로드하면, 백엔드는 파일을 파싱하여 정산 데이터를 DB에 저장 (`POST /api/v1/admin/settlements/pg/upload`).
 - 조회된 정산 내역이 목록에 표시된다.
 - 호출 API 정보:
 - 조회: `GET /api/v1/admin/settlements/pg`
 - 파일 업로드: `POST /api/v1/admin/settlements/pg/upload`
-

기능 2: 내부 데이터와 대사 (개념)

- 관련 요구사항: SYS-REQ-020
 - 화면 구성 아이디어:
 - 정산 내역 목록에서 특정 건 선택 시, 해당 기간/PG사의 내부 결제/취소 내역과 비교하는 상세 화면 또는 팝업 표시.
 - 불일치 내역(누락, 금액 차이 등) 하이라이트 기능.
 - 대사 완료 또는 확인 필요 상태 변경 기능.
 - 참고: 상세한 대사 로직 및 화면 구성은 추가 협의 및 설계 필요.
-

기능 3: 실시 기관 정산 관리 (개념)

- 관련 요구사항: SYS-REQ-020 (실시 기관 정산 포함)
 - 화면 구성 아이디어:
 - 과정별/기관별 수익 분배 정책 설정 화면 (필요시).
 - 기간별/기관별 정산 대상 금액 계산 및 조회 화면.
 - 정산금 지급 처리 및 내역 관리 화면.
 - 참고: 수익 분배 모델 및 정산 프로세스가 복잡할 수 있어 상세 설계 필요.
-

예외 처리 및 유의 사항

- 데이터 표준:** PG사별 정산 데이터 형식이 다를 수 있으므로, 데이터 파싱 및 표준화 방안 필요.
 - 자동화:** PG사 API 연동을 통해 정산 데이터 자동 수신 및 대사 자동화 고려.
-

계산서 관리 화면 (SYS-SCR-020)

사용 역할자: 시스템 관리자, 재무 담당자

관련 메뉴 : 시스템 관리 > 결제 관리 > 계산서 관리

기능 1: 계산서 발행 내역 조회

- 관련 요구사항: SYS-REQ-027 (계산서 발행 내역 관리)
 - 입력 정보:
 - 검색 조건: 기간(발행일), 신청자 ID/이름, 사업자 정보, 발행 상태(발행 완료, 발행 실패, 요청).
 - '검색' 버튼, 페이지네이션, 정렬.
 - 출력 정보:
 - **계산서 발행 내역 목록 테이블:** 번호(No), 발행일, 신청자, 공급받는 자 정보(상호, 사업자번호), 공급가액, 세액, 합계 금액, 발행 상태, 국세청 전송 상태.
 - 각 행 클릭 시 상세 정보 및 재발행/취소 등 버튼 표시.
 - 프로세스 설명: 발행된 (또는 발행 요청된) 계산서 내역을 조회. 외부 전자 계산서 서비스 연동 결과 반영.
 - 호출 API 정보: **GET /api/v1/admin/tax-invoices**
-

기능 2: 계산서 발행 처리 (수동)

- 관련 요구사항: SYS-REQ-026 (계산서 발행 처리)
 - 입력 정보:
 - '신규 발행' 버튼 또는 발행 요청 건 선택.
 - **공급받는 자 정보:** 사업자 번호, 상호, 대표자명, 주소, 업태, 종목, 이메일. (사업자 번호 입력 시 정보 자동 조회 기능 연동 가능)
 - **공급자 정보:** 시스템 설정 또는 자동 표시.
 - **작성일자:** Date Picker.
 - **품목 정보:** (수강 과정 정보 연동 또는 직접 입력) 과정명, 수량, 단가, 공급가액, 세액. (면세 여부 등 고려)
 - **결제 정보 연동:** (필요시) 관련 결제 내역 선택.
 - '발행' 버튼.
 - 출력 정보:
 - 입력 정보 유효성 검증 결과.
 - 발행 요청 성공/실패 메시지. (실제 발행 및 국세청 전송은 비동기 처리될 수 있음)
 - 프로세스 설명:
 1. 관리자가 계산서 발행에 필요한 정보를 입력한다.
 2. '발행' 버튼 클릭 시, 백엔드 API (**POST /api/v1/admin/tax-invoices**) 호출.
 3. 백엔드는 외부 전자 계산서 발행 서비스 API를 호출하여 발행을 요청한다.
 4. 발행 서비스 처리 결과(성공, 실패, 국세청 전송 상태)를 받아 시스템 DB에 업데이트하고 프론트엔드에 응답.
 - 호출 API 정보:
 - 사업자 정보 확인 API (외부 연동): 필요시 구현.
 - 계산서 발행 요청 API: **POST /api/v1/admin/tax-invoices**
-

예외 처리 및 유의 사항

- **외부 서비스 연동:** 전자 계산서 발행 서비스(예: Popbill, Smartbill 등) 연동 필수. API 연동 및 인증 정보 관리 필요.
 - **세법 준수:** 공급가액, 세액 계산, 면세 처리 등 세법 관련 요구사항 정확히 반영 필요.
 - **수정/취소 발행:** 이미 발행된 계산서의 수정 또는 취소 발행 기능 지원 필요 (외부 서비스 연동).
-

관리자 로그인 화면 (SYS-SCR-019)

사용 역할자 : 시스템 관리자, 운영자 등 관리자 역할을 가진 사용자

관련 메뉴 : 관리자 사이트 접근 시 최초 화면 (별도 메뉴 없음)

기능 1: 관리자 ID/비밀번호 기반 로그인 처리

- 관련 요구사항: **SYS-REQ-006 (관리자 사이트 분리 구성)** - 분리된 관리자 사이트의 진입점
 - 입력 정보:
 - 관리자 ID (이메일 또는 별도 ID): 텍스트 필드. 필수.
 - 비밀번호: 비밀번호 필드. 필수. 마스킹 처리.
 - 출력 정보:
 - 로그인 성공 시: 관리자 대시보드 화면으로 이동. 관리자 역할에 맞는 메뉴 표시. JWT 토큰 및 관리자 정보 수신.
 - 로그인 실패 시 (ID/PW 불일치): "아이디 또는 비밀번호가 일치하지 않습니다." 오류 메시지 표시.
 - 로그인 실패 시 (권한 없음): "관리자 계정이 아니거나 접근 권한이 없습니다." 메시지 표시.
 - 로그인 실패 시 (계정 상태 오류): 해당 상태 안내 메시지 표시.
 - 네트워크/서버 오류 시: "일시적인 오류가 발생했습니다." 메시지 표시.
 - 프로세스 설명: 사용자 로그인 화면(USR-SCR-001)의 기능 1과 유사하나, 대상 사용자가 관리자 역할(예: ROLE_ADMIN, ROLE_OPERATOR)을 가진 사용자로 제한된다.
 1. 관리자가 ID와 비밀번호를 입력하고 '로그인' 버튼 클릭.
 2. 프론트엔드 유효성 검사.
 3. 백엔드 관리자 로그인 API (**POST /api/v1/admin/auth/login**) 호출.
 4. 백엔드는 사용자 인증 및 관리자 역할 보유 여부 확인.
 5. 성공 시 관리자용 JWT 토큰 및 정보 반환, 실패 시 오류 반환.
 6. 프론트엔드는 결과에 따라 처리 (토큰 저장, 화면 이동 또는 오류 메시지 표시).
 - 호출 API 정보:
 - HTTP 메서드: **POST**
 - API 엔드포인트: **/api/v1/admin/auth/login**
 - API 목적: 관리자 계정 인증 및 관리자 세션(토큰) 생성.
 - 요청/응답 형식: 사용자 로그인 API와 유사하나, 역할 검증 로직 추가.
 - Mockup 데이터 예시: 생략 (USR-SCR-001 참조)
-

기능 2: (선택적) 비밀번호 찾기 이동

- 관련 요구사항: 관리자 계정의 비밀번호 분실 대비.
 - 입력 정보: '비밀번호 찾기' 링크 클릭.
 - 출력 정보: 관리자용 비밀번호 찾기 화면(별도 설계 또는 사용자용 화면 연계 - 정책 결정 필요)으로 이동.
 - 프로세스 설명: 사용자 비밀번호 찾기 이동(USR-SCR-001 기능 5)과 유사.
-

예외 처리 및 유의 사항

- 접근 제어: 관리자 로그인 화면 자체는 접근 가능해야 하나, 로그인 성공 시 반드시 관리자 역할이 있는지 확인해야 함.

- **보안 강화:** 관리자 계정은 일반 사용자 계정보다 더 강력한 보안 조치(예: 2단계 인증(MFA), IP 접근 제한) 적용 고려.

권한(Permission) 관리 화면 (SYS-SCR-021)

사용 역할자: 시스템 관리자

관련 메뉴: 시스템 관리 > 권한 관리 > 세부 권한 관리

기능 1: 권한(Permission) 목록 조회

- 관련 요구사항: **SYS-REQ-010 (개인 정보 접근 권한 관리)** 를 포함한 시스템 내 모든 세부 기능 권한 관리의 기반.
 - 입력 정보:
 - 검색 조건:
 - 권한 ID (코드): 텍스트 필드.
 - 권한명: 텍스트 필드.
 - 권한 그룹: Select 박스 (SYS-SCR-004에서 관리되는 그룹).
 - HTTP 메서드: Select 박스 (GET, POST, PUT, DELETE 등).
 - URL 패턴: 텍스트 필드.
 - '검색' 버튼, 페이지네이션, 정렬.
 - 출력 정보:
 - 권한 목록 테이블: 체크박스, 번호(No), 권한 ID, 권한명, 권한 그룹, 설명, (필요시) HTTP 메서드, URL 패턴.
 - 총 권한 수, 페이지네이션 컴포넌트.
 - 프로세스 설명: 시스템 내 정의된 모든 세부 기능 권한(퍼미션) 목록을 조회. 퍼미션은 특정 기능(예: 사용자 상세 조회) 또는 API 엔드포인트 접근 제어 단위가 될 수 있음.
 - 호출 API 정보:
 - HTTP 메서드: GET
 - API 엔드포인트: `/api/v1/admin/permissions`
 - 파라미터: `page`, `size`, `sort`, `permissionId`, `permissionName`, `groupId`, `httpMethod`, `urlPattern`
 - 응답 형식: 표준 페이징 목록 응답 (권한 정보 포함).
-

기능 2: 권한(Permission) 상세 조회/수정

- 관련 요구사항: **SYS-REQ-010** 등 세부 권한 관리.
- 입력 정보:
 - (목록에서 권한 ID 클릭 시) 해당 권한 정보가 채워진 폼 표시.
 - 권한 ID: 텍스트 필드 (수정 불가).
 - 권한명: 텍스트 필드 (수정 가능). 필수.
 - 권한 그룹: Select 박스 (변경 가능).
 - 설명: 텍스트 영역 (수정 가능).
 - (API 기반 권한일 경우):
 - HTTP 메서드: Select 박스 (수정 가능).
 - URL 패턴: 텍스트 필드 (Ant/Glob 패턴 등 사용).
 - '저장' 버튼, '취소' 버튼.

- 출력 정보: 저장 성공/실패 메시지.
 - 프로세스 설명: 개별 퍼미션의 속성(이름, 설명, 그룹, API 매핑 정보 등)을 수정.
 - 호출 API 정보:
 - 상세 조회: `GET /api/v1/admin/permissions/{permissionId}`
 - 수정: `PUT /api/v1/admin/permissions/{permissionId}`
 - 요청 바디: `{"permissionName": "string", "groupId": "string", "description": "string", "httpMethod": "string", "urlPattern": "string"}`
-

기능 3: 권한(Permission) 신규 등록

- 관련 요구사항: **SYS-REQ-010** 등 세부 권한 관리.
 - 입력 정보:
 - '신규 등록' 버튼.
 - **권한 ID**: 텍스트 필드. 필수. 중복 불가. (예: `USER_VIEW_DETAIL`, `COURSE_MANAGE`)
 - **권한명**: 텍스트 필드. 필수. (예: '사용자 상세 정보 조회', '과정 관리')
 - **권한 그룹**: Select 박스.
 - **설명**: 텍스트 영역.
 - **(API 기반 권한일 경우)**: HTTP 메서드, URL 패턴.
 - '등록' 버튼, '취소' 버튼.
 - 출력 정보: ID 중복 시 오류, 등록 성공/실패 메시지.
 - 프로세스 설명: 새로운 세부 기능 권한을 시스템에 등록.
 - 호출 API 정보:
 - ID 중복 확인: `GET /api/v1/admin/permissions/check-id?permissionId=...`
 - 등록: `POST /api/v1/admin/permissions`
 - 요청 바디: 수정 API와 유사한 구조.
-

예외 처리 및 유의 사항

- **권한 정의**: 어떤 단위(기능 버튼 클릭, 특정 API 호출 등)로 권한을 정의할지 일관된 정책 필요.
`USER_VIEW_DETAIL` 같은 명명 규칙 사용 권장.
- **개인 정보 접근 권한**: **SYS-REQ-010** 을 충족하기 위해, 개인 정보에 접근하는 기능(예: 사용자 상세 조회, 수정)에 대한 퍼미션(예: `PRIVACY_USER_READ`, `PRIVACY_USER_WRITE`)을 명확히 정의하고, 이 권한을 가진 역할에게만 해당 기능 접근을 허용해야 함. 역할 관리(SYS-SCR-005) 화면에서 이 퍼미션들을 역할에 매핑.
- **자동 생성**: 시스템 개발 시 특정 API 엔드포인트에 대한 권한을 자동으로 스캔하여 등록하는 기능 고려 가능 (개발 생산성 향상).
- **역할 매핑**: 여기서 정의된 권한은 역할 관리(SYS-SCR-005) 화면에서 역할에 매핑되어야 실질적인 효과를 발휘함.