

# MICE 행사 관리 시스템

총청권 MICE 행사/컨퍼런스를 위한 비표 발급 및 세션 관리 웹 서비스입니다. 이 프로젝트는 참가자, 연사, 관리자별 역할을 분리하고, 동적 QR 코드를 활용한 안전한 출석 체크 시스템을 제공합니다.

 **팀원 가이드:** 협업 가이드 | 초기 설정 체크리스트

## 배포 정보 (Production)

이 프로젝트는 **Vercel**과 **Railway**를 통해 배포 및 운영되고 있습니다.

- **Frontend (Vercel):** <https://mice-orcin.vercel.app/>
- **Backend (Railway):** <https://mice-production.up.railway.app/>

### 배포 아키텍처

- **Frontend:** React 기반의 정적 파일을 Vercel에 배포하여 빠른 로딩 속도와 글로벌 CDN을 활용합니다.
- **Backend:** Node.js 서버를 Railway에 배포하여 운영합니다.
- **Database:** Railway의 PostgreSQL 서비스를 백엔드 서버에 연동하여 사용합니다.

### 배포 가이드

-  **빠른 시작 가이드:** 5단계로 빠르게 전체 서비스를 배포하는 방법을 안내합니다.
- **상세 배포 가이드:** 각 단계에 대한 상세한 설명과 문제 해결 방법을 포함합니다.
- **배포 체크리스트:** 배포 전후 확인할 항목 목록입니다.

## 核心 기능

### 1. 역할 기반 접근 제어 (RBAC)

- **Admin:** 모든 데이터(사용자, 세션) CRUD, 통계 확인, 동적 QR 생성
- **Speaker:** 본인에게 할당된 세션 정보 확인, Q&A 관리, 발표 자료 업로드
- **Attendee:** 세션 목록 조회, 즐겨찾기, 질문 등록, 출석 체크

### 2. 하이브리드 QR 인증 시스템

- **정적 QR:** 참가자의 '나의 비표'로, 행사장에 입장할 때 사용됩니다.
- **동적 QR:** 60초마다 갱신되어 세션 출석 체크에 사용됩니다. (캡처본을 이용한 어뷰징 방지)

### 3. 실시간 세션 관리

- 세션별 Q&A 등록 및 조회
- 발표 자료 업로드 및 다운로드
- 세션 즐겨찾기 기능

## 기술 스택

Backend

- **Framework:** Node.js, Express.js
- **Database:** PostgreSQL
- **ORM:** Prisma
- **Authentication:** JWT (JSON Web Token), Bcrypt
- **File Handling:** Multer
- **In-Memory Cache:** Node-Cache (동적 QR 토큰 관리)

## Frontend

- **Framework:** React 18
- **Build Tool:** Vite
- **Routing:** React Router
- **Styling:** Tailwind CSS
- **API Client:** Axios
- **QR Code:** `qrcode.react` (생성), `html5-qrcode` (스캔)

## 프로젝트 구조

```

MICE/
  └── backend/
      ├── prisma/
      ├── routes/
      ├── middleware/
      └── server.js          # Node.js 백엔드
                                # Prisma 스키마 및 마이그레이션
                                # API 라우트
                                # 인증/RBAC 미들웨어
                                # 서버 엔트리포인트

  └── frontend/
      └── src/
          ├── components/    # 공통 컴포넌트
          ├── context/        # AuthContext (전역 인증 상태)
          ├── pages/          # 역할별 페이지 컴포넌트
          └── config/          # Axios 인스턴스 설정
          └── vite.config.js

  └── README.md

```

## 로컬 환경에서 실행

### 사전 요구사항

- Node.js 18+
- PostgreSQL
- `npm`

### 1. Backend 설정

```
# 백엔드 폴더로 이동
cd backend
```

```
# 의존성 설치  
npm install  
  
# .env 파일 설정 (예시 파일을 복사하여 생성)  
cp .env.example .env  
# (DATABASE_URL 등 .env 파일 내부의 값을 본인 환경에 맞게 수정)  
  
# Prisma 설정 및 데이터베이스 마이그레이션  
npx prisma generate  
npx prisma migrate dev  
  
# 개발 서버 실행  
npm run dev
```

Backend는 <http://localhost:5050>에서 실행됩니다.

## 2. Frontend 설정

```
# 프론트엔드 폴더로 이동  
cd frontend  
  
# 의존성 설치  
npm install  
  
# .env 파일은 비워두면 Vite 프록시가 자동으로 백엔드를 바라봅니다.  
  
# 개발 서버 실행  
npm run dev
```

Frontend는 <http://localhost:3000>에서 실행됩니다.

## API 엔드포인트 요약

전체 API 명세는 [Backend README](#)에서 확인할 수 있습니다.

- **인증:** POST /api/auth/register, POST /api/auth/login
- **사용자:** GET /api/users/me
- **세션:** GET /api/sessions, GET /api/sessions/:id
- **관리자:** GET/POST/PUT/DELETE /api/admin/users, /api/admin/sessions
- ... 등

## 라이선스

MIT License