

1. 프로젝트 개요

본 프로젝트는 강의 시간표 자동 배정과 강의실 공실 분석, Google Calendar(ICS) 변환 기능을 제공하는 FastAPI 기반의 백엔드 시스템이다.

사용자는 CSV 파일로 강의 데이터를 업로드하고, 웹 UI 또는 API를 통해 다음 기능을 사용할 수 있다.

주요 기능 요약

CSV 기반 강의 배정 자동화

실습/이론에 따른 교실 배정

교수 선호 요일 반영

강의실/교수 중복 배정 방지

공실(빈 강의실) 분석

Google Calendar용 .ics 파일 생성

HTML/JSON API 모두 제공

이 프로젝트는 FastAPI의 구조적 장점을 활용하여 배정 로직(service.py) 과 라우팅(router.py) 을 명확히 분리하였고, 데이터는 데이터베이스 대신 메모리 기반 전역 상태(AppState) 로 관리한다.

2. 프로젝트 전체 구조도

```
project_root/
├── data/
│   └── courses_data.csv
├── src/
│   ├── main.py                # FastAPI 앱 엔트리포인트
│   ├── scheduler/
│   │   ├── models.py         # 전역 상태 AppState 정의
│   │   ├── schemas.py        # 요청 데이터 검증(Pydantic)
│   │   ├── service.py        # 배정/공실/ICS 핵심 알고리즘
│   │   └── router.py          # API & HTML 엔드포인트
├── requirements.txt
```

3. FastAPI 기본 구조 + 프로젝트 적용 방식

FastAPI는 “분리” 중심 아키텍처로 구성하는 것이 가장 이상적이다.

본 프로젝트의 구조는 FastAPI의 권장 패턴을 충실히 따르고 있다.

3.1 main.py - 앱 초기화 & 라우터 등록

```
app = FastAPI(  
    title="강의실 자동배정 & 공실 분석 API",  
    version="1.0.0",  
)  
  
app.include_router(scheduler_router)
```

FastAPI 객체 생성

라우터(router.py) 등록

DB 연결, 로그 설정 등도 추가할 수 있는 자리

역할: "프로그램의 입구", 실제 기능은 전부 다른 파일에 분리

3.2 router.py - 모든 API/HTML 엔드포인트 관리

```
router = APIRouter(tags=["scheduler"])
```

이 파일에는 사용자가 접근하는 모든 기능이 존재한다.

예시:

/ → 메인 HTML 페이지

/upload-csv → CSV 업로드

/assign → JSON 자동배정 API

/assign-html → HTML 시간표

/vacancy → 공실 JSON API

/vacancy-html → 공실 HTML

/free-slots-ics → ICS 파일 생성

역할: 사용자 인터페이스(UI/API) 담당

3.3 service.py - 핵심 알고리즘 구현부

이 파일이 프로젝트의 두뇌(Brain) 역할을 한다.

구현된 주요 기능은 다음과 같다.

시간표 자동 배정 알고리즘

강의실 공실률 및 빈 슬롯 계산

Google Calendar용 ICS 파일 생성

역할: “모든 로직이 들어 있는 핵심 계산 엔진”

3.4 models.py - 메모리 기반 전역 상태(AppState)

```
@dataclass
class AppState:
    original_df: Optional[pd.DataFrame]
    result_df: Optional[pd.DataFrame]
    preferred_days: Dict[str, List[str]]
```

state = AppState() 형태로 싱글톤처럼 사용된다.

저장값	설명
original_df	CSV 업로드 원본
result_df	최종 배정 결과
preferred_days	교수별 선호 요일

역할: 데이터 저장소(DB 대체)

3.5 schemas.py - Pydantic 검증 모델

요청(JSON)을 검증한다.

```
class PreferredDaysRequest(BaseModel):
    prof: str
    days: List[str]
```

잘못된 형식이 들어오면 FastAPI가 자동으로 422 에러를 반환한다.

역할: 요청 데이터 자동 검증 + 타입 안전성 보장

4. 파일별 역할 상세 설명

4.1 main.py

FastAPI 실행 파일

라우터 포함

Swagger 문서 자동 활성화

복잡한 로직을 작성하지 않고, 오직 “구성(Configuration)”만 담당한다.

4.2 router.py

이 파일은 전체 제품을 사용자에게 공개하는 인터페이스다.

요약 표:

엔드포인트	설명
/	메인 페이지
/upload-csv	CSV 업로드
/assign-html	HTML 배정 결과
/assign	JSON 배정 결과
/vacancy-html	HTML 공실 페이지
/vacancy	JSON 공실 분석
/preferred-days	교수 선호 요일 설정
/free-slots-ics	ICS 파일 다운로드

서비스(service.py)에서 계산된 모든 결과를 사용자에게 “보여주는” 역할.

4.3 models.py

데이터를 서버 메모리에 적재하여 유지한다.

이 구조는 작은 프로젝트에서 DB 없이도 훌륭히 작동하며, 속도 또한 매우 빠르다.

4.4 schemas.py

Pydantic 모델을 사용하여 JSON 요청을 자동 검증한다.

또한 API 문서(Swagger)에서 자동으로 스키마가 표시된다.

4.5 service.py - 핵심 알고리즘 파일

여기가 프로젝트의 핵심이며, 아래 알고리즘을 포함한다.

5. 알고리즘 중심 설명

5.1 시간표 자동 배정 알고리즘

전체 흐름

- 1) 실습/이론 분리 후 랜덤 셔플
- 2) 각 과목별 교수/강의실/강의 유형 분석
- 3) 교수 선호요일 반영
- 4) 중복 배정 방지
- 5) 가능한 슬롯 탐색 → 배정
- 6) 배정 결과 DataFrame 생성

5.1.1 블록 정의

BLOCKS = [(1,3), (4,6), (7,9)]

즉,

1블록 = 1-3교시

2블록 = 4-6교시

3블록 = 7-9교시

블록 단위 배정은 스케줄 관리에서 가장 일반적이고 효율적인 방식이다.

5.1.2 실습·이론 분리 & 랜덤 셔플

```
prac_df = df[df["강의유형구분"] == "실습"]  
theory_df = df[df["강의유형구분"] != "실습"]
```

실습 수업을 먼저 배정하여 실습실 확보 문제를 해결한다.

5.1.3 교수 선호 요일 반영

```
if prof in preferred_day_dict:  
    priority_days = preferred_day_dict[prof]
```

선호 요일 우선 배정

선호 요일이 없다면 CSV의 “선호요일” 필드 파싱

나머지 요일은 랜덤하게 뒤에 배치

이 로직으로 “교수 요구조건 반영”을 지원한다.

5.1.4 중복 체크 및 배정

중복 체크는 2가지 조건을 모두 만족해야 한다.

1.강의실 중복 방지

```
schedule[(day, period, room)]
```

2.교수 시간 중복 방지

```
prof_schedule[(prof, day, period)]
```

두 조건 모두 충족 시 배정 성공.

5.1.5 배정 실패 처리

배정할 수 없는 수업이 있다면 오류 대신 "배정 실패" 메시지를 남긴다.

시스템이 죽지 않도록 설계된 안전 장치.

5.2 공실(빈 강의실) 분석 알고리즘

입력: 배정된 시간표 DataFrame

출력:

강의실별 공실률

free_slots (대여 가능한 시간대)

계산 과정

전체 블록 수 = 5일 × 3블록 = 15개

사용된 블록 수 = result_df로부터 계산

남은 블록 수 = 전체 - 사용된

공실률 = (남은/전체)*100

공실 분석은 다음 기능에서 활용된다.

/vacancy-html → HTML 화면 출력

/vacancy → JSON 데이터 제공

/free-slots-ics → 캘린더 변환

5.3 ICS 파일 생성 알고리즘

Google Calendar에서 읽을 수 있는 표준 형식으로 변환.

base_monday 날짜 기준으로 요일별 날짜 계산

각 슬롯을 VEVENT 형식으로 변환

ICS 파일 전체 구조 조립

FastAPI에서 StreamingResponse로 파일 다운로드 제공

이 기능은 행정실 대여 시스템, 강의실 예약 시스템 등에 즉시 적용 가능하다.

6. 전체 파이프라인 설명 (전체 흐름)

1. CSV 업로드 (/upload-csv)
2. 서버 메모리에 original_df 저장 (AppState)
3. 자동 배정 실행 (service.assign_timetable)
4. 배정 결과 result_df 저장
5. 공실 분석 (service.compute_vacancy_stats)
6. HTML/JSON 페이지로 결과 조회
7. ICS 파일 생성하여 다운로드 (/free-slots-ics)

7. 프로젝트 장점 분석

구조가 매우 명확함 (유지보수 용이)

main / router / service / model / schema 분리

FastAPI의 장점을 100% 활용

자동 문서화(Swagger)

요청 구조 자동 검증

예외 처리 자동화

Pandas 기반 데이터 처리

CSV 기반 프로젝트에 매우 적합

배정 결과를 DataFrame으로 관리 → HTML/JSON 변환 용이

공실 분석 + 캘린더 연동

일반적인 시간표 시스템보다 실용성 강화

확장성 높음

DB 연동

인증 시스템

사용자별 시간표 저장

React/Vue 프론트엔드 결합

강의실 자동배정 시스템 (FastAPI)

현재: 업로드된 CSV 없음

1. CSV 업로드

파일 선택

선택된 파일 없음

업로드

※ 업로드 후 자동으로 이 페이지로 돌아옵니다.

2. 자동 배정 실행

자동 배정 실행 및 결과 보기

3. 공실 분석 / 대여 가능 시간

[공실 분석 및 대여 시간 HTML 보기](#)

4. Google Calendar용 ICS 파일 다운로드

기준 주의 월요일 날짜 (YYYY-MM-DD):

2025-12-01

ICS 파일 다운로드

[Swagger API 문서 \(/docs\)](#)

자동 배정 결과

교과목명	담당교수	요일	시작교시	종료교시	배정강의실
인공지능모델운영	조상구	월	1	3	1216
IoT시스템개발	정환익	월	7	9	1216
데이터분석실습	조문석	월	7	9	1217
M화면구현(2)	유선주	월	1	3	1418
M서버프로그램구현(1)	나기수	월	4	6	1418
SQL활용(2)	서공원	월	7	9	1418
M애플리케이션테스트수행	홍주영	월	1	3	외부강의실1
M애플리케이션배포	박상렬	월	4	6	외부강의실1
웹프로그래밍(2)	안철훈	월	7	9	외부강의실1
M프로젝트실습(1)	이정훈	화	1	3	1216
텍스트&영상처리	조상구	화	7	9	1217
MSQL활용(1)	안철훈	화	1	3	1418
M앱프로그래밍(1)	배희호	화	1	3	외부강의실1
데이터시각화실습	김대중	화	4	6	외부강의실1
M애플리케이션테스트수행	홍주영	화	7	9	외부강의실1
앱프로그래밍(2)	서공원	수	1	3	1215
M프로그래밍언어활용(1)	배희호	수	1	3	1216
M서버프로그램구현(1)	나기수	수	4	6	1216
캡스톤디자인	정환익	수	1	3	1217
MSQL활용(1)	안철훈	수	7	9	1217
M빅데이터처리	유소율	수	1	3	외부강의실1
서버프로그램구현(2)	안철훈	수	4	6	외부강의실1
M화면구현(2)	유선주	목	1	3	1215

공실 분석

room	total	used	free	free_rate
1215	15	4	11	73.333333
1216	15	10	5	33.333333
1217	15	7	8	53.333333
1418	15	7	8	53.333333
외부강의실1	15	11	4	26.666667

대여 가능 시간 (슬롯)

day	room	start	end
월	1215	1	3
월	1215	4	6
월	1215	7	9
월	1216	4	6
월	1217	1	3
월	1217	4	6
화	1215	1	3
화	1215	4	6
화	1215	7	9
화	1216	4	6
화	1216	7	9
화	1217	1	3
화	1217	4	6
수	1418	4	6
