

PJT명	SSAFIT Project	
단계	[Web BackEnd PJT]	
진행일자	2024.03.15	
예상 구현 시간	필수기능	8H
	추가기능	3H
	심화기능	2H

## 1. 목표

- Java의 Back-End 기술인 Servlet/JSP를 이해하고 이를 활용하여 웹 서버를 구축할 수 있다.
- MVC 구조를 이해하고 이를 활용하여 프로그램을 작성할 수 있다.
- Front-End 관통 프로젝트를 참고하여 화면과 연계가 필요한 부분의 Back-End를 완성한다.

## 2. 요구 사항

Ssafit 프로젝트의 Back-End 파트를 구현하는 것을 목표로 한다.  
MVC 기반의 웹 프로젝트를 설계하고 구현하여 보자.

백앤드 데이터 관리는 이전 자바 관통에서 진행한 것처럼 메모리를 활용하여 진행한다. 소스 작성 전 작성하려는 Back-End 클래스의 구조와 네이밍룰 등을 검토하여 완성도 높은 결과물이 나오는 것을 목표로 하자.

1) 백앤드 구현시 STS를 이용하여 Dynamic Web Project로 구현한다.

2) 기능 구현

(1) 기본기능

다양한 운동영상정보와 영상의 리뷰를 위한 백앤드 기능을 구현한다.

- 메인페이지 백앤드 기능 구현

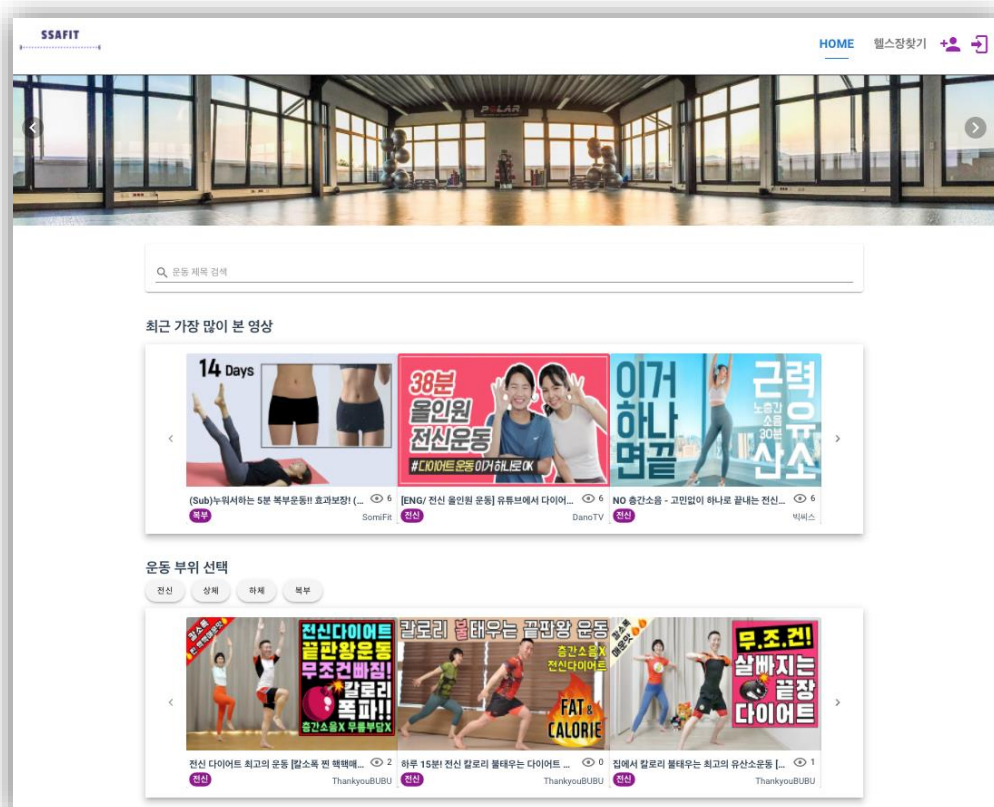
: 운동 영상 정보의 다양한 출력을 위한 백앤드 구현(조회수 기준 영상, 운동부위별 영상)

- 운동영상에 대한 리뷰관리 백앤드 기능 구현

: 목록, 등록, 수정, 상세, 삭제

메인페이지 백앤드 작성 예시

- 메인페이지 UI



- Controller

```
@WebServlet("/main")
public class MainController extends HttpServlet {
    // 비즈니스 로직 객체 선언
    private MainDao mainDao = MainDaoImpl.getInstance();

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // View 페이지에 필요한 데이터 공유
        request.setAttribute("interestList", mainDao.selectInterestViewFitVideo());
        request.setAttribute("partList", mainDao.selectPartFitVideo());
        // View 페이지 이동
        request.getRequestDispatcher("/main.jsp").forward(request, response);
    }
}
```

- Dao

```
public class MainDaoImpl implements MainDao {
    private List<Video> list;
    private static MainDaoImpl instance;
    private MainDaoImpl() {
        list = new ArrayList<Video>();
        list.add(new Video("gMaB-fG4u4g", "ThankyouBUBU", 10, "전신", "전신 다이어트 최고의 운동 [칼소폭 전
        list.add(new Video("s14NQ6Cz4QE", "힙으뜸", 12, "전신", "살 빠지는 전신운동 루틴 with 피지컬갤러리"));
    }
    public static MainDao getInstance() {
        if (instance == null) instance = new MainDaoImpl();
        return instance;
    }

    public List<Video> selectInterestViewFitVideo() {}
    public List<Video> selectPartFitVideo() {}
}
```

- Dto

```
// 화면의 비디오 정보를 표현하는 데이터 중심 클래스 정의
public class Video {
    private String title;
    private String fitPartName;
    private String youtubeId;
    private String channelName;
    private int viewCnt;
    public Video() {}
    public Video(String youtubeId, String channelName,
        int viewCnt, String fitPartName, String title) {
        this.youtubeId = youtubeId;
        this.channelName = channelName;
        this.viewCnt = viewCnt;
        this.fitPartName = fitPartName;
        this.title = title;
    }
}
```

## (2) 추가 기능

사이트의 다양한 기능을 사용하기 위해서 회원가입, 로그인 기능을 제공한다

- 회원가입 백엔드 구현
- 로그인, 로그아웃 백엔드 구현

### (3) 심화 기능

회원은 관심있는 영상을 찜 할 수 있고 찜한 영상을 삭제 할 수 있다.

회원은 다른 회원을 팔로우 할 수 있다.

- 회원의 팔로우 정보 관리 백엔드 구현
- 회원의 영상 찜 관리 백엔드 구현

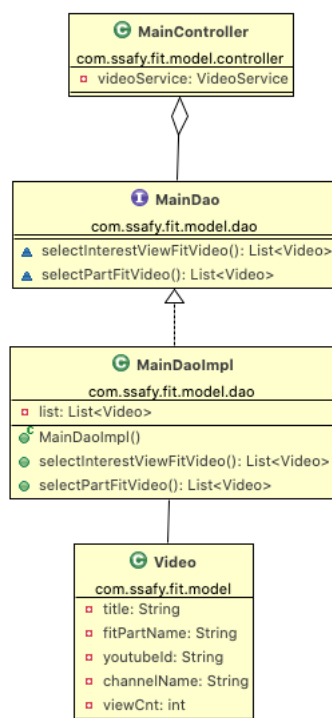
### 3. 구현 방법

- ① 관통 프로젝트의 요구사항을 이해하고, STS에서 프로젝트를 생성한다.
- ② 요구사항을 분석하고 MVC 아키텍처를 기반으로 클래스를 설계한다.
- ③ 설계된 클래스를 기반으로 Front-End 관통의 페이지와 연동하여 Back-End를 구현한다.
- ④ 프로젝트는 GitLab을 활용하여 Pair 프로젝트로 수행한다.
- ⑤ 기본 기능 구현을 모두 완료하고 추가 기능, 심화 기능을 구현한다.

### 4. 클래스 다이어그램(참고용)

➤ 프로젝트에 맞게 작성합니다.

메인페이지 클래스 다이어그램 예시



## 5. 산출물과 제출

- 프로젝트 최종적으로 제출해야 할 항목은
  - ✓ 기본 기능 구현된 화면을 Capture한 이미지와 Source
  - ✓ 추가 기능 구현된 화면을 Capture한 이미지와 Source
  - ✓ 심화 기능 구현된 화면을 Capture한 이미지와 Source
  - ✓ 구현된 기능의 클래스 다이어그램