

# REPORT

## 웹서버컴퓨팅 팀팀클래스 리포트



과 목	웹서버컴퓨팅
담당 교수	김상철
제출 일자	2025.06.02
전공	소프트웨어학과
팀명	팀명
팀원	송예찬(20213016), 정의현(20203135)

## - 목 차 -

I . 서론 .....	1
II . 본론 .....	1
1. Django	
2. FastAPI	
3. Spring Boot	
III . 결론 .....	4
IV . 참고문헌.....	5

## I. 서론

현대 웹과 애플리케이션은 다양한 요구사항과 기술 변화에 맞춰 빠르게 발전하고 있다. 이에 따라 개발자들이 효율적이고 안정적인 서비스를 구축하기 위해 웹 프레임워크를 만들었다. 최근 우리가 자주 사용하는 인스타그램이나 넷플릭스, 구글, 페이스북과 같은 애플리케이션이나 웹을 만드는 데 웹 프레임워크를 사용하고 있다. 웹 프레임워크는 서버, 클라이언트, 데이터베이스와의 통신을 손쉽게 처리할 수 있는 구조를 제공하고 코드의 재사용성과 유지보수성을 높이는데 기여한다.

이 리포트에서는 대표적인 웹 프레임 워크인 Django, FastAPI, Spring Boot, Node.js를 선정하여, 각 프레임 워크의 등장 배경, 역사, 장점, 간단한 프로그래밍 예제, 사용 사례, 향후 전망에 대해 조사했다. 이를 통해 각 웹 프레임워크의 강점과 차별점을 파악하고 프로젝트의 목적과 규모에 따라 적합한 프레임 워크를 선택하는 데 도움이 될 것이다.

## II. 본론

### 1. Django

Django 웹 프레임 워크의 사용 언어는 Python으로 2003년 미국의 Lawrence Journal-World라는 신문사의 웹 프로그래머 Adrian Holovaty와 Simon Willison에 의해 Python을 사용해 애플리케이션을 만들기 시작하면서 처음 개발되었다. 당시 Lawrence Journal-World의 웹사이트는 PHP기반으로 개발되었는데 Adrian Holovaty와 Simon Willison은 PHP가 규모가 큰 웹사이트에 적합하지 않다고 생각했다. 마침 마크 필그림의 “Dive Into Python”에 영향을 받아 Python이란 언어에 빠지게 되었고 Python을 이용해 웹 개발을 시작했다. 하지만 규모가 큰 웹 개발에 적합한 파이썬 도구가 없다는 것을 깨닫고 Django를 개발하기 시작했고 2005년 7월에 BSD 사용 허가서로 오픈소스로 공개되었다.

Django의 장점으로 웹 개발에서 번거로운 요소들을 새로 개발할 필요 없이 내장된 기능을 이용해 빠른 개발을 할 수 있다. 그리고 CSRF 보호, SQL injection 보호와 같은 주요 보안 기능을 기본적으로 제공한다. 또한 자동으로 생성되는 관리자 페이지를 제공하고 앱 구조로 이루어져 확장성이 있어 대규모 프로젝트에 적합하다. 마지막으로 ORM을 통해 개발 과정에서 DB와 연결이 쉽다.

Django를 사용하는 대표적인 웹사이트로는 Instagram, Disqus, Pinterest가 있다.

Django는 Python의 인기와 데이터 분석과 AI와의 연계가 용이하다는 점으로 인해 미래에 더 많이 사용될 것이다.

간단한 코드 예제:

```
from django.http import HttpResponse
```

```
def hello(request):
```

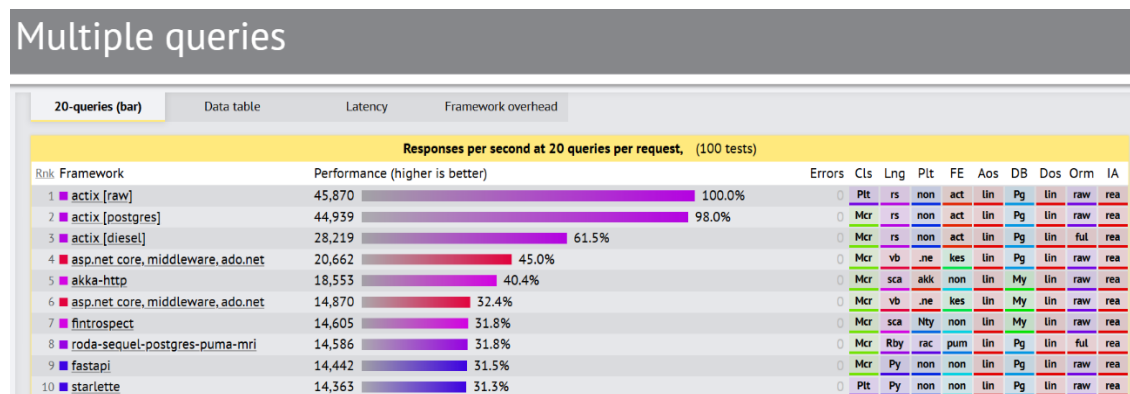
```
    return HttpResponse("Hello, Django!")
```

## 2. FastAPI

FastAPI 웹 프레임워크의 사용언어는 Python이다. 2018년 개발자 Sebastián Ramírez가 만든 웹 프레임 워크로 머신러닝, 분산 시스템, 비동기 작업과 같은 복잡한 요구사항을 가진 API를 개발해오면서 기존의 Python 웹 프레임워크인 Django, Flask로는 한계를 느꼈다. 그래서 비동기 프로그래밍 지원, 자동 문서화, 타입 기반 데이터 검증, 최신 표준 준수, 개발 생산성등의 자신이 원하는 기능을 하나의 프레임워크에 통합하여 FastAPI를 개발했다.

FastAPI는 Python 프레임워크 중 가장 높은 성능, async/await 기반의 비동기 프로그래밍 지원, 자동 문서 생성, JWT를 이용한 보안기능과 확장성을 갖추고 있다는 장점이 있다.

[다중 쿼리에 대한 FastAPI 성능 벤치마크 표]



FastAPI는 우버, 넷플릭스와 같은 웹사이트에서 사용되고 있다.

인기 언어인 Python 프레임워크이고 데이터 분석 및 AI서비스의 실시간화, 자동화된 API문서화, 데이터 자동 검증 기능으로 미래에 더 많은 사용이 있을 것이다.

간단한 코드 예제:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def hello():
```

```
    return "Hello, Flask!"
```

### 3. Spring Boot

Java를 사용하는 웹 프레임워크로 로드 존슨이 2002년에 출판한 Expert One-on-One J2EE Design and Development에 선보인 코드를 기반으로 시작하여 점점 발전했고 2003년 6월 아파치 2.0 라이선스로 최초 공개되었다. Java의 복잡한 XML설정과 반복적인 코드로 인한 생산성이 떨어지는 문제를 스프링 프레임워크가 해결했다. 하지만 스프링 역시 프로젝트 규모가 커질수록 설정이 복잡해지고 개발 초기 진입 장벽이 높다는 단점이 있었고 이에 따라 더 쉽고 빠른 개발을 위해 Spring Boot가 개발되었다.

Spring Boot는 반복 작업과 설정 부담을 줄여 개발자 친화적이라는 점, 자동 구조와 풍부한 문서들이 있어 유지보수에 용이한 점, Spring Data와 Spring Security 등 다른 Spring 프로젝트와 연동하여 다양한 확장 기능이 사용가능한 점, 마지막으로 내장 서버와 프로덕션 준비 기능으로 배포와 준영이 간편한 점 등이 장점이다.

네이버, 쿠팡, 카카오, 무신사 등 많은 국내기업들이 Spring Boot를 사용하고 있다. 또한 넷플릭스, 아마존, 알리바바 등 글로벌 기업도 사용하고 있다.

Spring Boot는 마이크로서비스, 클라우드 네이티브, 대규모 엔터프라이즈 시스템의 표준 프레임워크로서 입지를 더욱 강화하고 있다. 최신 기술 트렌드와 통합과 높은 생산성과 운영 효율성, 그리고 실무에서의 채택으로 인해 java 백엔드 개발의 중심이 될 것이라고 전망된다.

간단한 코드 예제:

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.*;

@SpringBootApplication

@RestController

public class HelloSpringBoot {

    public static void main(String[] args) {

        SpringApplication.run(HelloSpringBoot.class, args);

    }

    @GetMapping("/")

    public String hello() {

        return "Hello, Spring Boot!";

    }

}
```

### III. 결론

현재 Django, FastAPI, Spring Boot는 각각의 장점을 바탕으로 웹 개발, 데이터 분석, AI, 마이크로 서비스, 클라우드 네이티브 등의 핵심적인 역할을 하고 있다. 특히 데이터 분석과 AI의 중요성이 증가함에 따라 Python 기반의 Django와 FastAPI는 AI서비스와의 결합에서 주로 사용될 것이고 Spring Boot는 클라우드, 엔터프라이즈 환경에서 입지를 강화할 것이다.

결론적으로, 위 프레임워크들은 앞으로도 웹과 애플리케이션 개발에 핵심적인 역할을 이어갈 것이고 개발자들은 프로젝트의 목적에 맞는 프레임워크를 선택하고 각 프레임워크의 발전을 지속적으로 학습해야 한다.

## IV. 참고문헌

Django 등장 배경과 역사

<https://www.quora.com/What-is-the-history-of-the-Django-web-framework-Why-has-it-been-described-as-developed-in-a-newsroom>

Django 사용 사이트, 장점

<https://www.djangoproject.com/start/overview/>

Django 전망

<https://www.moweb.com/blog/python-django-in-2025-will-it-remain-the-top-choice-for-web-development#a-technical-breakdown-of-django>

FastAPI 역사, 미래

<https://fastapi.tiangolo.com/history-design-future/>

FastAPI 장점

<https://dotcommagazine.com/2023/06/fastapi-a-comprehensive-guide/>

FastAPI 전망

<https://www.nucamp.co/blog/coding-bootcamp-backend-with-python-2025-python-in-the-backend-in-2025-leveraging-asyncio-and-fastapi-for-highperformance-systems>

Spring Boot의 역사, 장점, 미래

<https://medium.com/javarevisited/evolution-of-spring-boot-and-microservices-4d1109b5a4d3>

Spring Boot 장점, 사용 사례

<https://staticmania.com/blog/spring-boot>

Spring Boot 전망

<https://www.credosystemz.com/blog/future-of-spring-boot-in-2025/>

웹 프레임워크 종류와 특징, 장단점

<https://bramilch.github.io/posts/kr-web-framework-comparing/>

웹 프레임워크 설명

<https://wikidocs.net/197525>