
2023년 한이음 ICT멘토링 프로젝트 결과보고서

프로젝트명	실시간 소스코드 라이선스 검증 서비스
-------	----------------------

(본문) 프로젝트 결과보고서

I. 프로젝트 개요

1. 프로젝트 소개

1) 기획 의도

- 소프트웨어 라이선스 검증 웹 서비스를 무료로 제공
- 기업에서 개발 중간중간 라이선스 위반 문제를 사전에 예방

2) 프로젝트 내용

- Github 저장소에서 코드 변경이 일어날 때, 라이선스 위반 가능성을 검증
- 각 PR마다 모든 소스코드의 라이선스를 분석하고, 결과를 Comment로 알림
- 웹 서비스로 분석 리포트를 제공해 저장소별 라이선스 검증 결과를 쉽게 파악



History Dashboard Support 한이음

Kim/project_repo Issue Detected pull/5

Info

Scan Date: 2023-06-30 13:59:47

Detected Licenses: Apache 2.0, GPL-3.0-only, Open SSL, GPL-2.0 or later, MIT

Detected License Ratio



Detected file list

Status	Path	Match Line	Match Value	Detected Licenses	Component
Permitted	main.go	110-170	71%	Apache 2.0	Component >
Restricted	main.go	11-70	90%	GPL-3.0-only	Component >
Prohibited	servercontrol.go	141-198	81%	Open SSL	Component >
Permitted	authenticator_test.go	11-71	80%	GPL-2.0 or later	Component >
Permitted	authenticator_test.go	11-20	20%	MIT	Component >

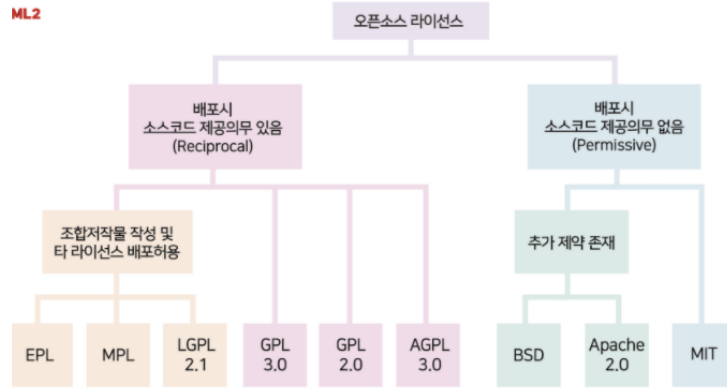
< 1 2 3 4 5 >

Pull Request 분석 리포트

2. 개발 배경 및 필요성

1) 개발 배경

- 오픈소스 소프트웨어는 개발 효율을 높여 주지만 이를 상업적으로 배포할 시에는 반드시 라이선스를 준수해야 하며, 위반할 경우 법적 분쟁, 재정적 손실, 평판 손실 등의 심각한 결과를 초래할 수 있음
- 국내의 경우 과거 2018년 한글과컴퓨터에서 라이선스를 위반하여 법정 분쟁 끝에 205만 달러의 합의금을 지불한 사례가 있으며, 해외의 경우 2022년 마이크로소프트가 오픈소스 라이선스 위반 혐의로 피소가 된 사례가 있음



다양한 라이선스와 서로 다른 의무사항

- 개발 중간중간 라이선스 준수 여부를 확인해야 하지만 Commit 되는 모든 소스코드마다 라이선스를 검증하는 것은 많은 시간과 자원이 소비됨. 만약 개발 마지막 단계에서만 검증한다면, 위반 시 프로젝트 전체에 대한 수정이 불가피함

2) 필요성

- 오픈소스 코드를 가져와 소스코드를 작성하는 경우 다양한 라이선스가 포함될 수 있기에 개발자들이 소스코드를 적용하기 전에 사용 가능한 라이선스인지 확인 필요
- 프로젝트 배포 전 라이선스 위반 가능성을 충분히 검증했다면 추후 배포를 진행할 때, 배포하는 기업과 고객 양쪽 모두에게 신뢰를 제공할 수 있음
- 오픈소스 라이선스 위반 문제를 예방하기 위해 'BlackDuck', 'FOSSology' 와 같은 상업용 검증 툴을 많은 기업에서 활용하고 있음
- 검증에 대한 필요성이 증가함에 따라 LG의 경우 'FOSSLight', Kakao의 경우 'OLIVE Platform' 등과 같이 일부 기업은 검증 서비스를 직접 개발하기도 함

3. 프로젝트 특·장점

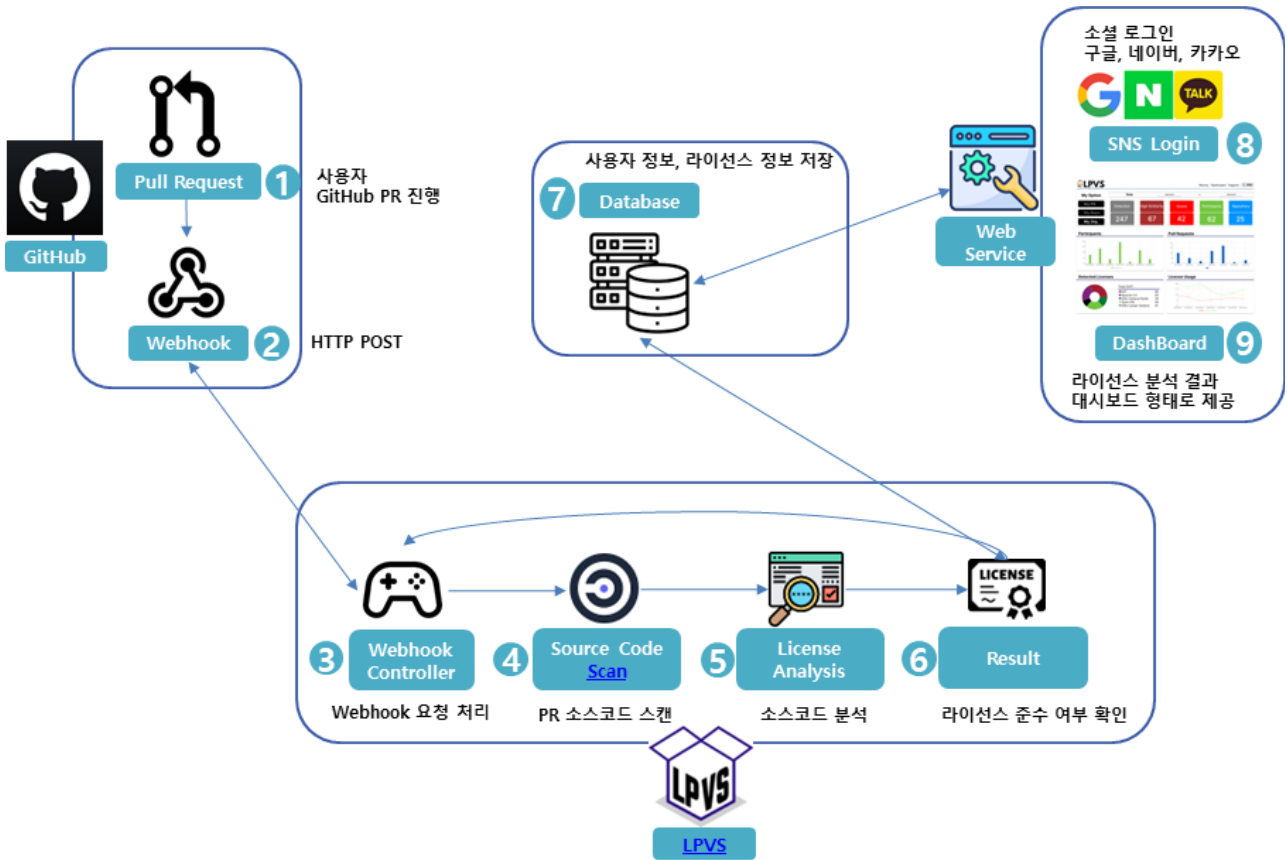
1) 특·장점

- 간편한 무료 서비스: 현재 제공되는 서비스의 경우 사용 절차가 까다롭고, 중소기업 또는 개인 개발자가 사용하기에는 비용에 부담이 됨. 본 서비스는 누구나 간단히 회원가입만 하면, 무료로 언제든지 실시간 검증이 가능하므로 오픈소스를 사용하더라도 위반 걱정 없이 개발자는 개발에만 집중할 수 있음
- GitHub 플랫폼 지원: 본 서비스는 GitHub 플랫폼과 원활하게 통합되어 협업 및 코드 리뷰 프로세스를 강화함. GitHub PR마다 소스코드 내 잠재적인 라이선스 위반 또는 문제점을 Comment로 알려주어 개발자가 라이선스 관련 문제를 쉽게 식별하고 해결할 수 있도록 도움
- 쉬운 모니터링 도구: 웹 대시보드로 개발 프로세스 전반에 걸친 라이선스 관련 모니터링 결과를 개발자/관리자에게 쉽게 요약하여 보여줌으로 생산성을 향상

II. 프로젝트 내용

1. 프로젝트 구성도

1) 전체 기능 목록




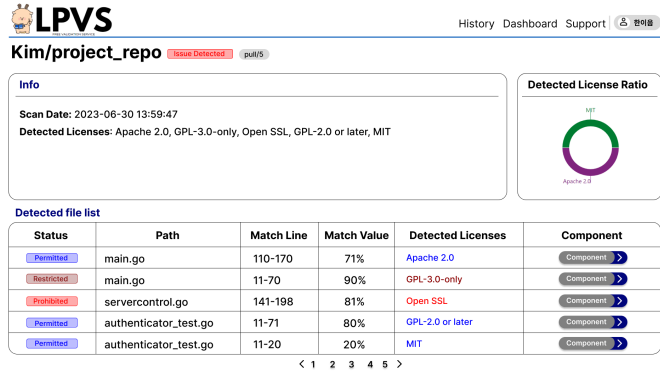
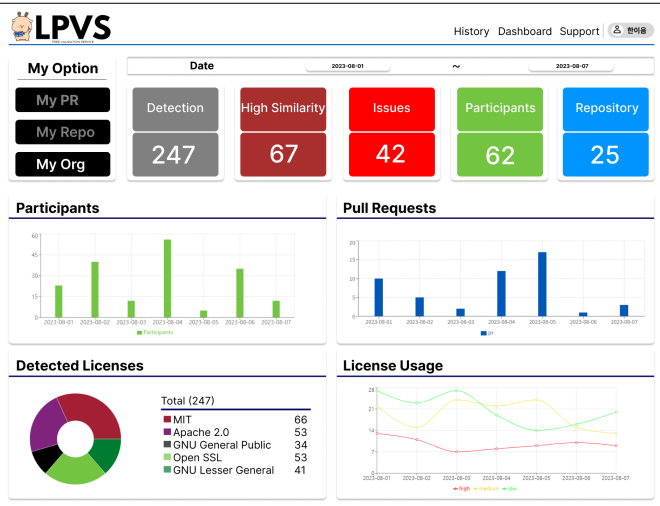
- 사용자가 GitHub 저장소에 Pull Request를 생성하면, Pull Request 정보를 담은 GitHub Webhook이 발생
- LPVS는 해당 Webhook 정보를 바탕으로 소스코드 라이선스 분석을 진행하고 분석 결과를 데이터베이스에 저장
- 사용자는 로그인 이후 자신과 관련된 Pull Request 분석 결과를 웹 서비스에서 확인할 수 있음

2. 프로젝트 기능

1) 전체 기능 목록

구분	기능	설명	현재진척도(%)
S/W	소셜 로그인	SNS(네이버, 카카오, 구글) 로그인을 지원하여, 해당 SNS 계정으로 본 서비스를 이용할 수 있음	100%
	소스코드 스캔	사용자가 본인이 작성한 소스코드를 GitHub에서 Pull Request를 하면, 스캐너가 업로드된 소스코드를 스캔	100%
	라이선스 분석	소스코드 스캔 데이터를 받아 라이선스 사용 여부를 감지하고, 위반 여부를 분석/검증하고 이를 DB에 저장	100%
	대시보드	분석/검증 결과를 DB로부터 받아 해당 데이터를 사용자가 직관적으로 볼 수 있도록 리포트 형식의 대시보드를 제공	100%
H/W			

2) S/W 주요 기능

기능	설명	프로젝트실물사진
소스코드 스캔	라이선스 스캐너가 프로젝트를 구성 하는 모든 소스코드에 대해 스캔을 진행하고 결과 제공	
라이선스 분석	스캔된 소스코드를 바탕으로 라이선스 준수 여부를 분석 리포트로 제공. 이때 소스코드에서 라이선스가 사용된 파일, 라인, 일치율 등을 확인할 수 있음	
대시보드	라이선스 분석/검증 결과를 하나의 대시보드로 제시. 프로젝트를 진행하면서 참여한 구성원 수, 생성된 Pull Request 수, 감지된 License 수 및 Usage 정보를 제공	

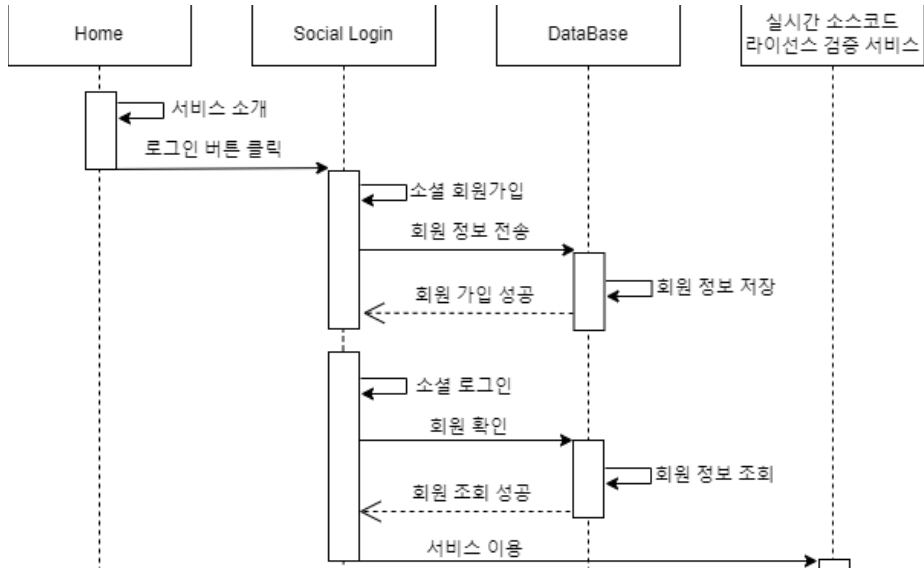
3) H/W 주요 기능

기능/부품	설명	프로젝트실물사진

3. 주요 적용 기술

1) S/W기능 구현을 위한 시나리오

- 소셜 로그인



사용자는 SNS 계정을 선택하여 로그인해야 서비스의 접근 권한을 얻음. Google, Naver, Kakao 중 하나의 플랫폼을 선택하면 SNS의 로그인 페이지로 이동하고, 로그인이 성공하면 Access Token을 통해 웹 페이지의 세션을 유지하고 필요한 회원 정보를 데이터베이스에 저장함.

적용 기술: OAuth 2.0 인증 프로토콜, React, Spring

- 라이선스 스캔

```

ild/package/run.sh": [
  "component": "edge-home-orchestration-go",
  "file": "GoMain/run.sh",
  "file_hash": "591f6594b6c599b513b753d70daf1afc",
  "file_url": "https://osskb.org/api/file_contents/591f6594b6c599b513b753d70daf1afc",
  "id": "file",
  "latest": "1.2.0",
  "licenses": [
    {
      "checklist_url": "https://www.osadl.org/fileadmin/copyleft",
      "copyright": "no",
      "name": "Apache-2.0",
      "osadl_updated": "2023-08-13T02:13:00+00:00",
      "patent_hints": "yes",
      "source": "component_declared",
      "url": "https://spdx.org/licenses/Apache-2.0.html"
    },
    {
      "lines": "all",
      "matched": "100%",
      "oss_lines": "all",
      "purl": [
        {
          "pkg": "github:lf-edge/edge-home-orchestration-go"
        }
      ],
      "release_date": "2020-10-30",
      "server": {
        "kb_version": {
          "daily": "23.08.17",
          "monthly": "23.08"
        },
        "version": "5.3.0"
      },
      "source_hash": "591f6594b6c599b513b753d70daf1afc",
      "status": "pending",
      "url": "https://github.com/lf-edge/edge-home-orchestration-go",
      "url_hash": "21ee1a519111ae45bc6a5e1d9fe5b720",
      "vendor": "lf-edge",
      "version": "coconut"
    }
  ]
]
  
```

사용자가 GitHub의 Webhook을 설정하고 Pull Request 요청 시 LPVS 서버로 POST 요청이 발생함. 이후 Pull Request한 코드의 라이선스를 검증하기 위해 SCANOSS를 실행해 스캔 결과를 JSON으로 저장. Webhook 정보와 SCANOSS의 결과인 JSON을 라이선스 분석에 사용.

적용 기술: SCANOSS

- 라이선스 분석

quswjdgma83 commented 2 weeks ago

[License Pre-Validation Service] No license issue detected

Detected licenses:

File: main.go

License(s): MIT (permitted), Apache-2.0 (permitted)

Component: edge-home-orchestration-go (GoMain/src/main/main.go)

Matched Lines: 18-201

Snippet Match: 91%

추출된 라이선스 정보를 분석하여 라이선스 위반 여부를 판단하고, 결과 값을 LPVS Database에 저장. [LPVS](#)는 Webhook에서 받아온 Pull Request 정보와 [SCANOSS](#)에서 스캔한 라이선스 일치 라인, 일치 정도 등의 정보를 알맞은 DB의 테이블에 저장함.

적용 기술: [LPVS](#), Spring JPA

2) 적용 알고리즘 및 주요 적용 기술

- RESTful API: 클라이언트가 보내는 HTTP 요청에 대해 서버가 필요한 정보를 응답하여 사용자가 서비스를 이용할 수 있도록 함. RESTful 아키텍처 원칙을 따르며, JSON 형식을 이용하여 데이터를 주고받음.

```
useEffect(() => {
  axios
    .get(`/dashboard/${type}/${name}`)
    .then((response) => {
      console.log(response.data);
      setDashboardInfo(response.data);
    })
    .catch((error) => console.log(error));
}, [type, name]);

const navigateToOrg = () => {
  navigate(`/dashboard/org/${username?.organization}`);
};
const navigateToOwn = () => {
  navigate(`/dashboard/own/${username?.nickname}`);
};
const navigateToSend = () => {
  navigate(`/dashboard/send/${username?.nickname}`);
};

@ResponseBody
@PostMapping("dashboard/{type}/{name}")
public Dashboard dashboardPage(
    @PathVariable("type") String type,
    @PathVariable("name") String name,
    @RequestParam("search_name") String searchName,
    @RequestParam("repo_name") String repoName,
    Authentication authentication) {

    Dashboard dashboardEntity;

    if (searchName.isEmpty() && repoName.isEmpty()) {
        dashboardEntity = statisticsService
            .getDashboardEntity(type, name, authentication);
    } else {
        dashboardEntity = statisticsService
            .getDashboardEntity(type, name,
                searchName, repoName, authentication);
    }

    return dashboardEntity;
}
```

프론트엔드는 Axios 통신을 활용해서 백엔드로 필요한 데이터를 받을 수 있는 EndPoint로 GET 요청을 보내면, 백엔드는 프론트엔드에 Endpoint에 맞춰서 해당 데이터를 JSON 형태의 객체로 전송함.

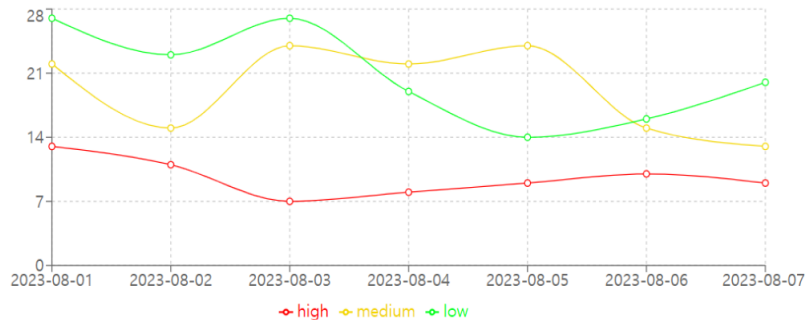
- Spring: 프로젝트에 서버를 구축하는데 사용된 프레임워크. Spring Data JPA를 통해 서버와 클라이언트, DB를 연결하고 JPA를 통해 받아온 DB의 데이터를 가공하고 클라이언트가 필요한 형식으로 변환하여 JSON을 반환하는 기능 담당.

```
{
  "name": "hwan5180",
  "licenseCountMap": {
    "AGPL-3.0-only": 1,
    "Apache-2.0": 223,
    "MIT": 71,
    "GPL-2.0-or-later": 0,
    "GPL-3.0-only": 1,
    "OpenSSL": 0
  },
  "totalDetectionCount": 296,
  "highSimilarityCount": 287,
  "totalIssueCount": 2,
  "totalParticipantsCount": 1,
  "totalRepositoryCount": 6,
  "dashboardElementsByDates": [
    {
      "date": "2023-08-07",
      "participantCount": 1,
      "pullRequestCount": 10,
      "riskGradeMap": {
        "MIDDLE": 8,
        "LOW": 0,
        "NONE": 0,
        "HIGH": 18
      }
    }
  ]
}
```

프론트엔드에서 대시보드 기능을 동작하는데 필요한 data를 백엔드에 요청하면 데이터를 파싱하여 전송함

- React: 웹 프론트엔드 개발을 위한 JavaScript 라이브러리로 사용자 인터페이스를 구축하기 위해 활용

License Usage



Recharts 라이브러리를 활용해 사용자가 분석 결과를 직관적으로 이해할 수 있도록 위와 같은 차트를 제작

```
//=====License Usage=====

const LU_dataMaps = dashBoardInfo?.dashboardElementsByDates?.
  map(function(element){
    | return element.date
  })

const high_Maps = dashBoardInfo?.dashboardElementsByDates?.
  map(function(element){
    | return element.riskGradeMap.HIGH
  })

const medium_Maps = dashBoardInfo?.dashboardElementsByDates?.
  map(function(element){
    | return element.riskGradeMap.MIDDLE
  })

const low_Maps = dashBoardInfo?.dashboardElementsByDates?.
  map(function(element){
    | return element.riskGradeMap.LOW
  })

const LU_data=[];

for (var i=0; i < LU_dataMaps?.length; i++) {
  LU_data.push({'name': LU_dataMaps[i], 'high': high_Maps[i],
    'medium': medium_Maps[i],
    'low': low_Maps[i]});
}
```

Rechart의 XAxis, YAxis API를 활용하여 차트를 구현. Rechart API의 형식에 맞게 x축에 날짜, y축에 License Usage를 설정. 백엔드에서 수신한 JSON을 map을 사용해 적절하게 데이터를 표시함

4. 프로젝트 개발 환경

구분		상세내용
S/W 개발환경	OS	Window, Ubuntu
	개발환경(IDE)	IntelliJ IDEA, Visual Studio Code
	개발도구	Spring, React
	개발언어	Java, JavaScript
	기타사항	MySQL, AWS
H/W 구성장비	디바이스	-
	센서	-
	통신	-
	언어	-
	기타사항	-
프로젝트 관리환경	형상관리	GitLab
	의사소통관리	Zoom, Notion
	기타사항	Figma, draw.io, dbdiagram.io

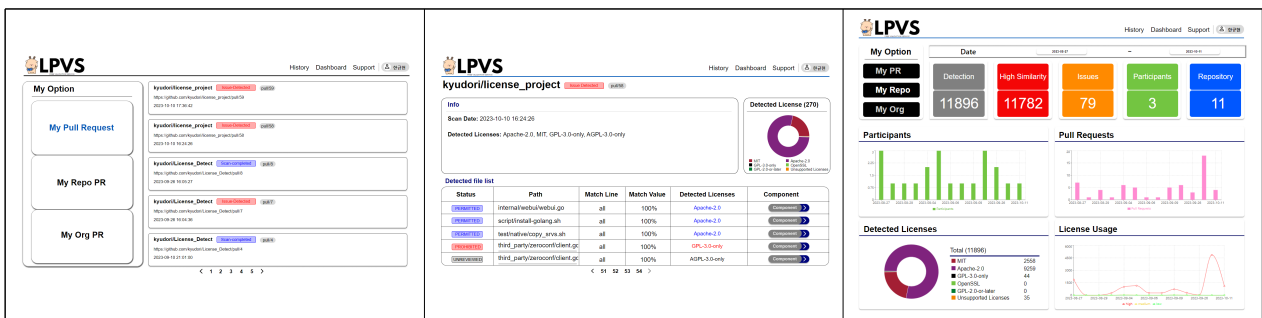
5. 장비(기자재/재료) 활용

번호	품명	작품에서의 주요기능

6. 프로젝트 작동 동영상

- <https://www.youtube.com/watch?v=xmcyV2glcpU>

7. 결과물 상세 이미지



8. 달성 성과

<input type="checkbox"/> 논문게재 및 포스터발표	게재(발표)자명	논문(포스터)명	게재(발표)처	게재(발표)일자
				2023. 00. 00.
<input type="checkbox"/> 앱(APP) 등록	등록자명	앱(APP)명	등록처	등록일자
				2023. 00. 00.
<input type="checkbox"/> 프로그램 등록	등록자명	프로그램명	등록처	등록일자
				2023. 00. 00.
<input type="checkbox"/> 특허/실용신안 출원	출원자명	특허/실용신안명	출원번호	출원일자
				2023. 00. 00.
<input type="checkbox"/> 기술이전	기술이전기업명	기술명	금액	이전일자
				2023. 00. 00.
<input checked="" type="checkbox"/> 공모전	구분(교내/대외)	공모전명	수상여부(출품/수상)	상격
	대외	2023 한이음 ICT멘토링 공모전	수상	입선
<input checked="" type="checkbox"/> 실용화	프로젝트 결과물을 오픈소스 프로젝트에 기여. 누구나 서비스를 이용할 수 있을 뿐만 아니라 소스코드를 수정 또는 변형하여 자유롭게 사용할 수 있도록 함.			
<input type="checkbox"/> 기타				

III. 프로젝트 수행 내용

1. 업무분장

번호	성명	역할	담당업무
1	김태완	멘 토	- 프로젝트 피드백
2		지도교수	
3	한규현	팀 장	- 프로젝트 관리, 서비스 설계 및 배포, 이슈 관리
4	변정흠	팀 원2	- 프론트엔드 개발, 웹 페이지 구현
5	이종민	팀 원3	- 백엔드 개발, LPVS 개선
6	전세환	팀 원4	- 프론트엔드 개발, 웹 페이지 구현

2. 프로젝트 수행일정

구분	추진내용	수행일정									
		3월	4월	5월	6월	7월	8월	9월	10월	11월	
계획	역할 분담, 기술 스택 확인										
분석	프로젝트 이해, 기술 스택 공부, 시스템 아키텍처 설계										
설계	프론트엔드 - UI, UX (화면) 설계										
	백엔드 - 데이터베이스, API 설계										
개발	LPVS 개선										
	프론트엔드 - 로그인 기능 개발										
	프론트엔드 - 메인 대시보드 개발										
	프론트엔드 - 사용자 세부 옵션 개발										
	백엔드 - Spring MVC 기반 REST API 개발										
	백엔드 - Spring JPA를 이용한 데이터베이스와의 연동										
	백엔드, 프론트엔드 통신 및 개발 통합										
	AWS EC2 클라우드 환경 서비스 배포										
테스트	QA, 피드백 수렴										
종료	프로젝트 종료										
	공모전 참가										
온·오프라인 미팅	결과물 피드백 및 이슈 해결										

3. 프로젝트 추진 과정에서의 문제점 및 해결방안

1) 프로젝트 관리 측면

- 일정 관리 문제: 팀원들 개인의 학사 일정과 프로젝트를 함께 진행하다 보니 계획했던 일정이 지연되는 문제가 있었음. 이를 해결하기 위해 GitLab에 해야 할 업무들을 정하고, Issue와 MileStone을 활용해서 프로젝트를 진행함
- 요구사항 변경과 Scope 문제: 프로젝트 경험이 부족과 팀원 개인별 수준 파악 미흡으로 인한 잦은 요구사항 변경과 프로젝트 범위가 달라지는 문제 발생. 이를 해결하기 위해 프로젝트를 프론트엔드와 백엔드로 두 파트를 나눠 팀원별 맡은 부분을 공부하고 집중적으로 해당 파트를 개발할 수 있도록 함

2) 프로젝트 개발 측면

- 새로운 기술 사용: React와 Spring 기술 스택을 처음 사용하여 개발을 진행하다 보니 개발 초기 익숙하지 않아 어려움을 겪음. 이를 해결하기 위해 Udemy 강의를 적극 활용하여 해당 기술 스택에 대한 실력을 키움
- 버전 및 호환성 문제: React와 Spring 프레임워크 및 라이브러리에서 버전에 따른 호환성 문제가 발생. 이를 해결하기 위해 업데이트 시 신중히 평가 후 진행했으며, 공식 문서를 참고하여 호환성 정보를 참고함
- 형상 관리 문제: 다수의 팀원이 코드를 수정하다 보니 형상 관리 및 테스트 시에 어려움을 겪음. 이러한 문제를 해결하고자 GitLab에서 역할 단위로 Sub Project를 나누고, branch들을 정리하여 팀원 간 빠르게 수정한 소스코드를 공유할 수 있도록 함

4. 프로젝트를 통해 배우거나 느낀 점

- 협업과 커뮤니케이션: 팀 프로젝트를 진행하기 위해 팀원들과 소통과 업무 분담, 문제 해결 등을 진행하는 과정에서 협업 능력을 배우고 소통에 중요성을 느낌
- 문제 해결 능력: 프로젝트 진행 과정에서 발생하는 리스크를 관리하고 실제로 발생한 이슈들을 팀원들과 소통하면서 해결하는 방법을 배움
- 실무 지식 습득: 실질적으로 필요한 기술과 도구에 대해 공부하고 프론트/백엔드 개발, 데이터베이스 설계 및 관리, API 설계 및 구현, 클라우드 환경 배포 등 실제 전공 지식이 실무에 어떻게 적용되는지를 배움
- 사용자 중심 개발: 실제 사용자에게 배포까지 진행될 것을 고려해 사용자들에게 제공할 가치에 대해서 고민해 보면서, 사용자 중심 개발의 중요성을 깨닫고, 사용자 경험과 피드백에 집중하는 방법에 대해 배움
- 기업 관점 개발: 기업은 단순 개발만 하는 것이 아닌, 개발 관련 여러 의존성을 점검하고 라이선스를 준수하는 등의 고려해야 할 지점이 많은 것을 느낌

IV. 기대효과 및 활용분야

1. 프로젝트의 기대효과

- 비용 부담 완화: ‘FOSSID’ 와 같은 유료 검증 서비스의 경우 자본 규모가 작은 중소기업이나 개인 개발자들이 구매하여 사용하기에는 부담이 됨. 이에 본 서비스를 활용해 라이선스 검증에 들이는 비용 부담을 줄이는데 기여함
- 법적 분쟁 감소: 실시간 소스코드 검증을 바탕으로 라이선스 사용 위반으로 인해 생기는 불필요한 법적 분쟁을 줄이는데 기여함
- 활발한 오픈소스 사용: 라이선스 이슈로 오픈소스 사용을 주저하거나 라이선스 의무사항을 고려하지 않고 사용하는 개발자들의 오픈소스 사용 진입장벽을 낮추고, 문제 없는 오픈소스 사용을 활성화하여 오픈소스 생태계 발전에 기여함
- 개발 효율 상승: ‘OLIS’ 와 같은 서비스의 경우 검증까지 오랜 시간이 걸려 수시로 라이선스 검증을 받기에는 어려움이 있음. 본 서비스는 누구나 간단한 회원가입 이후 수시로 검증 결과를 확인할 수 있으므로, 라이선스 검증에 들이는 시간을 줄여 프로젝트 개발에 투자해 프로젝트 결과물에 품질을 높일 수 있음

2. 프로젝트의 활용분야

- ‘유료’ 오픈소스 검증 서비스 상품을 구매하기 어려운 규모가 작은 회사에 소속된 중소기업 개발자나 개인 개발자가 본 웹 서비스를 활용해 라이선스 위반 여부를 ‘무료’ 로 검증할 수 있음
- 편리한 UI/UX를 제공하여 웹페이지에서 어려움 없이 라이선스 검증 결과를 쉽게 확인할 수 있음
- 개발자는 소프트웨어 개발을 하는 과정에서 수시로 자신이 작성한 코드가 라이선스를 위반하는지 검증할 수 있음
- 기업은 본 서비스를 사용하여 라이선스 측면에서 상시로 전체 현황을 파악하여 관리를 체계적으로 수행할 수 있음

V. 참고자료

1. 참고 및 인용자료

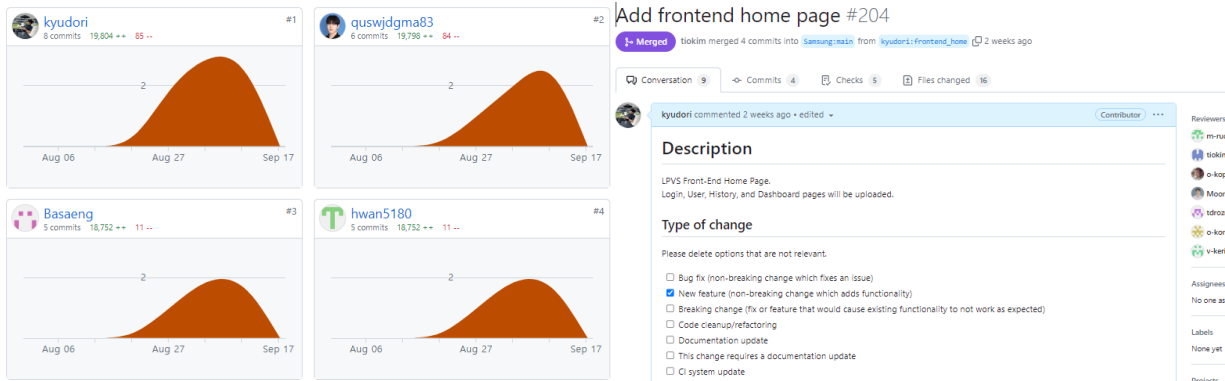
- LPVS, License Pre-Validation Service, <https://github.com/Samsung/LPVS>
- OLIVE Platform, 사용자 문서, <https://olive.kakao.com/docs/main>
- OLIS, 라이선스 소개, <https://www.olis.or.kr/license/introduction.do>

- 공모전: 2023년 한이음 ICT 멘토링 공모전 입선

프로젝트 참여현황

P.NO	프로젝트명	트랙	프로젝트수행기간	참여 인원	상태
17729	AI를 이용한 쓰레기 분리수거 로봇	한이음	2023-04-17 ~ 2023-11-15	4 명	입선
17717	실시간 소스코드 라이선스 검증 서비스	한이음	2023-04-17 ~ 2023-11-15	5 명	입선

- 실용화: 본 프로젝트를 수행하며 작성한 소스코드를 메인테이너의 리뷰를 받으며 오픈소스에 기여. 이를 통해 누구나 서비스를 이용할 수 있을 뿐만 아니라 소스코드를 수정 또는 변형하여 자유롭게 사용할 수 있음



오픈소스 기여(Commit) 현황

PR 생성 및 Merged