

실습 명세서

계절학기

"SSAFY GIT 시스템 활용"

1학기 종합 PJT를 활용한 프로젝트 학습 플랫폼 실습

목차

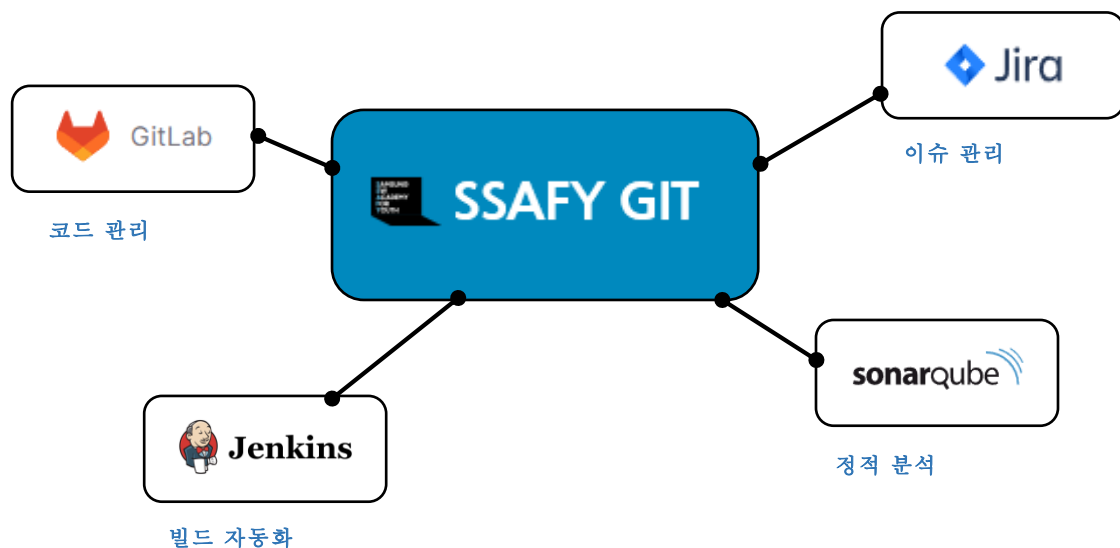
| | |
|-----------------------------|----|
| 1. 실습 개요 | 3 |
| 2. 실습 목표 | 5 |
| 3. 필수 사전 학습..... | 6 |
| 1) GITLAB..... | 6 |
| 2) JIRA..... | 6 |
| 3) JENKINS..... | 7 |
| 4. 과제 명세 | 9 |
| 1) 과제 목록..... | 9 |
| 2) 과제 상세..... | 10 |
| 5. 심화 과제 | 29 |
| 1) 빌드 설정..... | 29 |
| 2) SSAFY GIT 제공 DB 활용 | 29 |

1. 실습 개요

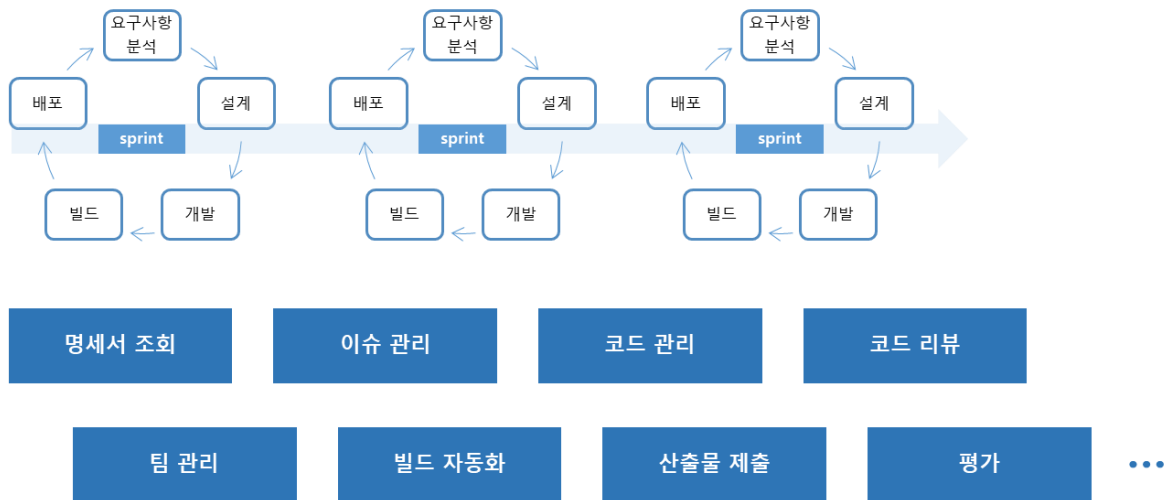
2학기 심화 과정은 자기주도형 프로젝트의 형태로써, 팀을 이루어 협업으로 개발을 진행합니다. 협업 환경에서 프로젝트를 원활하게 진행하기 위해서는 다양한 시스템이 필요하며 SSAFY에서는 프로젝트 학습 플랫폼(이하 SSAFY GIT)을 중심으로 이러한 시스템을 개발 과정에서 활용할 수 있도록 제공하고 있습니다.



현업에서도 개발 프로젝트를 효과적으로 진행하기 위해 다양한 개발 도구를 활용하고 있습니다. 협업 개발 환경에서 프로젝트 진척도 및 형상 관리 등의 효율성을 높이고 팀원 간 커뮤니케이션을 원활히 수행하게 하며 빌드, 릴리즈를 쉽게 하도록 하는 등 각각의 목적에 따라 사용하는 개발 도구 또한 매우 다양합니다.



SSAFY GIT은 팀 프로젝트를 효과적으로 수행함과 동시에 현업에서 많이 사용하는 시스템을 경험할 수 있도록 GitLab, Jira 등 유용한 시스템들과 연계하여 구성되어 있습니다.



요구사항 분석, 설계, 개발, 빌드, 배포 등 일련의 개발 프로세스를 수행하기 위해 개발적인 측면에서 필요한 부분들이 있으며, 프로젝트의 전반적인 관리와 진행을 위해 과정적인 측면에서 필요한 부분들이 있습니다. 이를 위해 SSAFY GIT은 크게 명세서 조회, 이슈 관리, 코드 관리, 코드 리뷰, 팀 관리, 빌드 자동화, 산출물 관리, 평가 등의 기능을 제공합니다.

각 프로젝트가 여러 주차에 걸쳐 팀으로 진행되는 만큼, 제공되는 시스템을 프로젝트 기간동안 잘 활용할수록 여러분의 프로젝트가 보다 원활해지고 결과의 수준 또한 향상되는 데에 큰 도움이 될 것입니다.

본 실습에서는 SSAFY GIT 및 연계 시스템에서 제공하는 기능들을 전반적으로 살펴보고 기본적인 사용법을 익히는 시간을 갖도록 하겠습니다.

2. 실습 목표

- ① 개발 프로세스의 이해
- ② SSAFY GIT의 인터페이스 이해
- ③ 각 개발도구의 활용 목적과 기본적인 사용법 숙지
- ④ 1학기 종합 프로젝트 리뷰 및 시스템 기능 적용

3. 필수 사전 학습

1) GitLab

버전 관리 시스템(VCS)인 git을 기반으로 저장소 및 단일 애플리케이션 개발에 필요한 기능을 제공하는 대표적인 DevOps 플랫폼 중 하나입니다. 프로젝트별로 멤버를 지정하여 공동 작업자와 함께 프로젝트를 진행할 수 있는 환경을 갖출 수 있으며, Merge Request(MR)를 통해 웹 상에서 손쉽게 코드 리뷰를 진행할 수 있습니다.

Activity, Commit 이력 등을 빠르게 확인할 수 있고 JIRA, Mattermost 등 다양한 도구와 연동이 가능하여 GitHub과 더불어 개발자들에 많은 유용성을 제공합니다.

하지만 GitLab은 git의 버전 관리에 대한 이해를 선행해야 보다 의미 있게 사용할 수 있습니다. git의 일부 기능을 웹으로 제공하고는 있으나 대부분의 과정을 git 커맨드 또는 IDE의 git Plug-in에서 수행하는 경우가 많기 때문입니다. 아래의 공식 웹사이트 링크를 통해 git의 기본 기능들을 살펴보고, 이외에도 git 브랜치 전략, GitLab 활용법 등을 찾아 학습할 것을 권장합니다.

- Git 공식 웹사이트: <https://git-scm.com/doc>

2) Jira

Atlassian에서 제공하는 이슈 관리 소프트웨어인 Jira는 이슈와 스프린트, 스크럼 보드 제공 등 이슈로 정의할 수 있는 개발 항목들(기능, 작업 단위, 버그 등)을 관리하는 데에 높은 활용성을 제공합니다.

Jira 시스템을 활용하는 조직마다 나름의 기준을 가지고 사용하는데 Jira에서 추구하는 애자일 프로세스에 대체로 기반하고 있으며, 이슈의 유형은 보통 아래와 같은 기준으로 구분하여 사용합니다.

| Issue Type | 설명 | 예시 |
|------------|---|--------------------------------------|
| Epic | 최상위 수준의 기능/작업 단위 (프로젝트 전반 또는 여러 Sprint에 걸쳐 진행할 정도의 범위) | 회원 관리, 로그인 관리 |
| Story | Epic에 대한 하위 Level 수준의 기능/작업 단위 | 회원 가입, 회원 정보 수정, 회원 탈퇴, 로그인, 비밀번호 찾기 |
| Bug | 프로젝트 개발/검증 중 발견된 버그 | 상품 검색 시 특정 상품이 조회되지 않는 문제 |
| Task | 개발에 직접 해당되지는 않으나 Sprint 안에 포함하여 해야 할 일 | ERD 작성, 테스트케이스 작성 |
| Sub-task | 위 Issue들과 관련하여 세부 단위 작업 등이 필요할 때 등록 | |

실습에서는 이슈를 생성하는 정도로 진행하겠지만, 본격적인 프로젝트를 시작하고 나면 여러 명의 개발자가 많은 기능을 동시에 개발하고 확인하는 과정에서 이슈 관리와 스크럼 보드의 중요성을 더욱 느끼고 적극 활용하게 될 것입니다. 아래 Jira 사용에 대해 잘 설명된 블로그 링크를 소개하니 사전에 이슈 관리에 대해 살펴보거나 또 다른 자료들을 찾아 학습할 것을 권장합니다.

- Jira 공식 가이드: <https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software>
- Jira를 이용한 애자일 scrum 프로젝트 관리: <https://bcho.tistory.com/826>

3) Jenkins

여러 명의 개발자가 하나의 프로젝트를 진행할 경우 서로 다른 소스 코드의 변경, 병합 등을 하면서 버전의 충돌이 발생할 수가 있습니다. 가장 큰 문제는 이전에 빌드에 문제가 없던 소스 코드가 변경 사항이 반영된 후 빌드 되지 않는 경우입니다. 이 때에는 이전의 어느 시점에서 빌드 문제를 야기했는지 찾기 위한 작업이 필요할 수밖에 없는데 Jenkins는 이러한 문제에 대응할 수 있도록 지속적인 통합(CI, Continuous Integration)을 제공하는 도구입니다.

빌드를 자동화함으로써 각자 작업한 내용을 서로 공유하고 있는 git 저장소에 업로드할 때마다 빌드를 수행하여 뒤늦게 발견될 수 있는 문제를 조기에 파악하고 어떤 변경 사

항이 어떤 영향을 미쳤는지 확인할 수 있습니다.

이에 더 나아가 자동화된 배포 설정을 통해 지속적인 배포(CD, Continuous Deploy) 또한 가능하도록 합니다.

즉, Git 저장소에 업로드된 소스 코드는 Jenkins와의 연동을 통해 빌드 및 배포 단계까지 자동으로 진행될 수 있습니다.

SSAFY GIT에서는 각 프로젝트별로 배포 환경이 상이함을 고려하여 빌드 현황을 관리하는 기능만 제공하지만, 배포 자동화에도 관심이 있는 경우 Jenkins 또는 CI/CD와 관련한 자료를 손쉽게 찾을 수 있으므로 이런 자료들을 통해 학습하고 팀 프로젝트에 적용해 볼 것을 권장합니다.

4. 과제 명세

1) 과제 목록

| 순번 | 과제명 |
|----|-----------|
| 1 | 프로젝트 준비 |
| 2 | 기능 정의서 작성 |
| 3 | 이슈 관리 |
| 4 | 소스코드 관리 |
| 5 | 빌드 관리 |
| 6 | 실행 기능 |
| 7 | 정리 단계 |

2) 과제 상세

아래의 과제들을 순서에 따라 진행합니다. 진행 중 **시스템적인 문제가 발생하는 경우**에는 [SSAFY GIT > Help > Q&A](#)에 문의하시기 바랍니다.

(1) 프로젝트 준비

① 프로젝트 팀 정보 확인

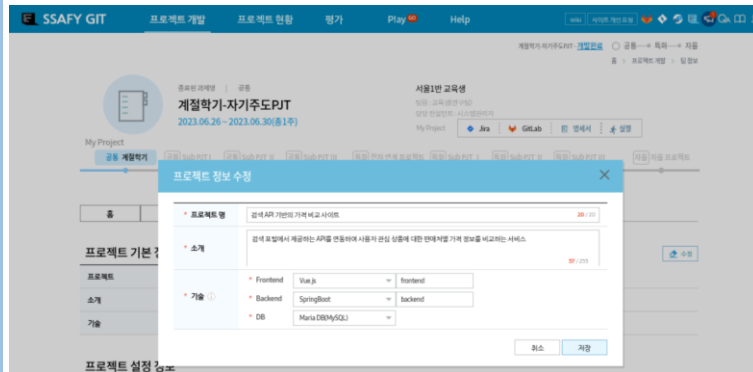
- SSAFY GIT > 프로젝트 개발 > 팀 정보
- 학사 일정에 따라 SSAFY GIT에 팀 정보가 등록됩니다. 개인 프로젝트의 경우 본인을 팀장으로 하는 1인 팀이, 팀 프로젝트의 경우 팀장과 팀원으로 구성된 다인 팀이 생성되어 있는 것을 볼 수 있습니다.

The screenshot shows the SSAFY GIT interface. At the top, there's a navigation bar with 'SSAFY GIT', '프로젝트 개발', '프로젝트 현황', '평가', 'Play', and 'Help'. Below this, the main content area displays project information for '계절학기-자기주도PJT' (2023.06.26 ~ 2023.06.30(총1주)). The '팀 정보' (Team Information) tab is selected, showing a table of team members. The table has columns for '역할' (Role), '이름' (Name), '이슈 현황' (Issue Status), '리뷰 현황' (Review Status), '활동 현황' (Activity Status), and '기술/역량 현황' (Skill/Capability Status). The '이슈 현황' column is further divided into '할일' (To-do), '진행' (In Progress), and '완료' (Completed). The '리뷰 현황' column is divided into '요청' (Request), '완료' (Completed), and '취소' (Cancelled). The '활동 현황' column is divided into '참여일' (Participation Date), '최종 방문일' (Last Visit Date), and '기술/역량 현황' (Skill/Capability Status). The '기술/역량 현황' column is further divided into 'URL', 'Backend URL', and '실명 주소' (Real Name Address).

| 역할 | 이름 | 이슈 현황 | | | 리뷰 현황 | | | 활동 현황 | | |
|----|----------|-------|----|----|-------|----|----|------------|------------|----------|
| | | 할일 | 진행 | 완료 | 요청 | 완료 | 취소 | 참여일 | 최종 방문일 | 기술/역량 현황 |
| 팀장 | 교과생(연구팀) | 0 | 0 | 0 | 0 | 0 | 0 | 2023.12.11 | 2023.12.11 | |

② 프로젝트 기본 정보 설정

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 기본 정보
- 수정 버튼을 눌러 본인의 1학기 종합 PJT의 내용으로 프로젝트 정보를 입력합니다.
 - 기술 선택 옵션에 맞는 항목이 없는 경우나 폴더명이 정확하지 않은 경우 공란으로 두고 추후에 수정할 수 있습니다.
 - 기술 항목의 설정값은 프로젝트 실행 기능과의 연동에 사용됩니다.



(2) 기능 정의서 작성

① 기능 정의서 양식 다운로드

- SSIFY GIT > 프로젝트 개발 > 산출물
- 양식 다운로드 버튼을 클릭하여 양식 파일(.xlsx)을 PC에 저장합니다.

The screenshot shows the SSIFY GIT project interface. The '산출물' (Deliverables) tab is selected, showing a list of deliverables. The '기능 정의서' (Functional Definition) deliverable is highlighted, and a download button is visible.

| 단계 | 산출물 | 설명 | 양식 | 생물 | 제출 현황 | 제출일 | 제출자 | 현황 |
|---|--------|----|------|----|-------|-----|-----|----|
| 요구 정의 <td>기능 정의서</td> <td></td> <td>다운로드</td> <td></td> <td></td> <td></td> <td></td> <td></td> | 기능 정의서 | | 다운로드 | | | | | |

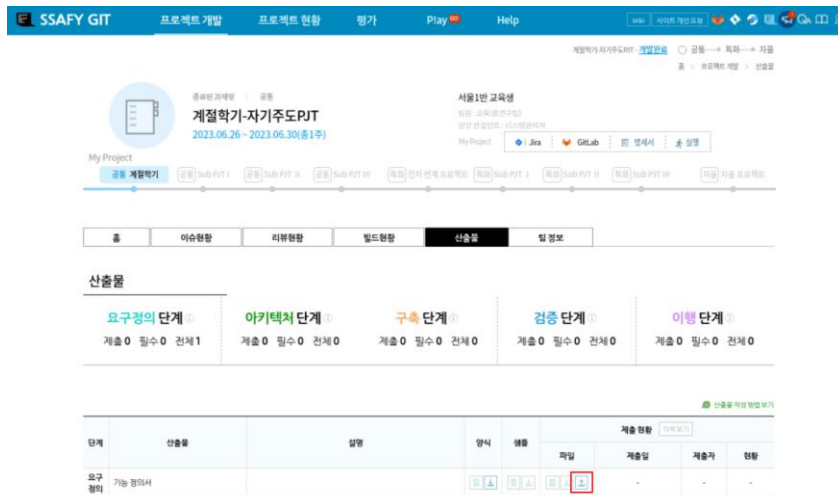
② 기능 정의서 작성

- 다운로드 받은 파일을 열어 1학기 종합 PJT의 결과물을 기준으로 기능 정의서를 작성합니다.
- 본 과제는 1학기 종합 PJT의 리뷰와 함께 Jira 이슈 관리에 등록하기 위한 기초 자료로 활용하기 위한 목적입니다. 따라서 전반적인 기능에 대해 적절히 분류하여 부담없이 작성하시기 바랍니다.

| 기능 정의서 | | | | |
|--------|---------|----------|---|----|
| ID | Depth 1 | Depth 2 | 내용 | 비고 |
| F01-1 | 핵심 관리 | 핵심 가입 | <ul style="list-style-type: none"> 이메일 주소, 이름, 연락처, 비밀번호 입력 이메일 주소 중복 시 중복 여부 확인 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| F01-2 | 회원 관리 | 회원 정보 수정 | <ul style="list-style-type: none"> 이메일 주소, 이름, 연락처, 비밀번호 변경 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| F01-3 | 회원 관리 | 회원 탈퇴 | <ul style="list-style-type: none"> 이메일 주소, 이름, 연락처, 비밀번호 입력 이메일 주소 중복 시 중복 여부 확인 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| F02-1 | 로그인 관리 | 로그인 | <ul style="list-style-type: none"> 이메일 주소, 비밀번호 입력 이메일 주소 중복 시 중복 여부 확인 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| F02-2 | 로그인 관리 | 비밀번호 찾기 | <ul style="list-style-type: none"> 이메일 주소, 비밀번호 입력 이메일 주소 중복 시 중복 여부 확인 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| F02-3 | 로그인 관리 | 로그아웃 | <ul style="list-style-type: none"> 이메일 주소, 비밀번호 입력 이메일 주소 중복 시 중복 여부 확인 비밀번호 입력 시 중복 여부 확인 비밀번호 입력과 비밀번호를 비교하여 일치 여부 확인 가입 완료 시 로그인 페이지로 이동 | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

③ 기능 정의서 제출

- SSAFY GIT > 프로젝트 개발 > 산출물
- 업로드 버튼을 클릭하여 작성한 기능 정의서 파일을 제출합니다.
 - 잘못 제출한 파일은 제출 후 활성화된 버튼을 클릭하여 삭제하거나 다시 제출할 수 있습니다.



The screenshot shows the SSAFY GIT project management interface. At the top, there's a navigation bar with 'SSAFY GIT', '프로젝트 개발', '프로젝트 현황', '평가', 'Play', and 'Help'. Below this, the project details for '계절학기-자기주도PJT' are shown, including the dates '2023.06.26 ~ 2023.06.30(총1주)'. A timeline of milestones is visible, with '기능 정의서' (Functional Definition) highlighted. Below the timeline, there's a section for '산출물' (Deliverables) with tabs for '요구정의 단계', '아키텍처 단계', '구축 단계', '검증 단계', and '이행 단계'. Each tab shows '제출 0', '필수 0', and '전체 0'. At the bottom, there's a table for '기능 정의서' (Functional Definition) with columns for '단계', '산출물', '설명', '양식', '샘플', '파일', '제출일', '제출자', and '현황'. The '파일' column has a red box around the '기능 정의서' file icon.

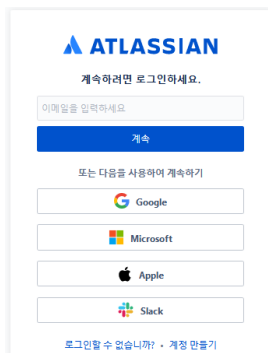
(3) 이슈 관리(Jira)

① Jira 계정 확인

- Jira를 사용하기 위해서는 SSAFY GIT과는 별도로 로그인을 해야 합니다.
- SSAFY GIT 우측 상단의 Jira 바로가기 아이콘을 클릭하여 Jira로 이동합니다.



- 이메일을 입력하여 로그인합니다.
 - 이메일: SSAFY GIT과 동일한 이메일 주소
 - 비밀번호: 처음 접속 또는 분실 시 하단의 "로그인할 수 없습니까?"를 통해 설정



② Jira 프로젝트 생성

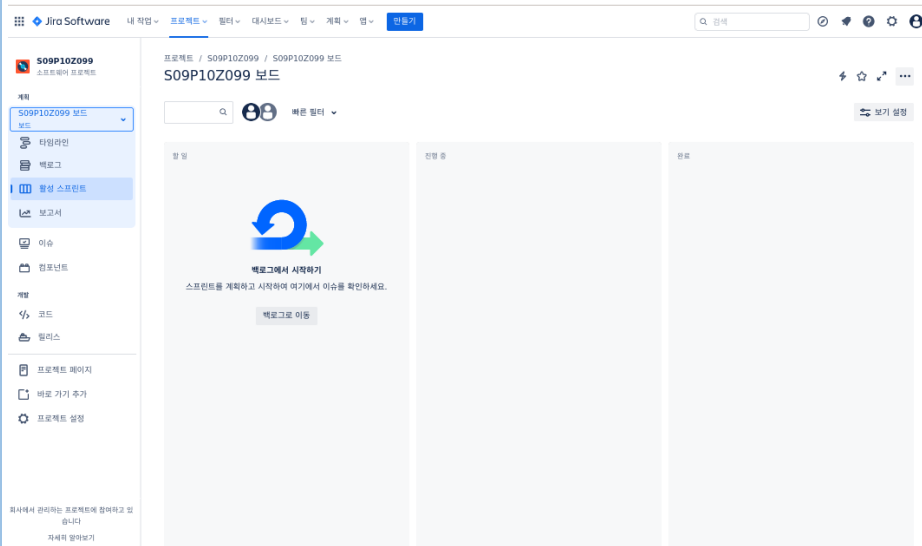
- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- Jira 계정이 정상 로그인되는 것을 확인했다면 SSAFY GIT으로 돌아와 Jira 우측의 [생성] 버튼을 클릭하여 Jira 프로젝트 생성을 요청합니다.
 - 모든 팀원의 SSAFY GIT 이메일과 Jira 이메일 정보가 동일해야 생성됩니다.

프로젝트 설정 정보

| | | |
|---------|--------------------------------|------|
| JIRA | | [생성] |
| Gitlab | Deploy Info User Name : Token | [생성] |
| Jenkins | | [생성] |
| MariaDB | User Info User Name : Password | [생성] |
| MongoDB | User Info User Name : Password | [생성] |
| 상행 이미지 | URL | [생성] |
| | Backend URL | |
| | 상행 소스 | |

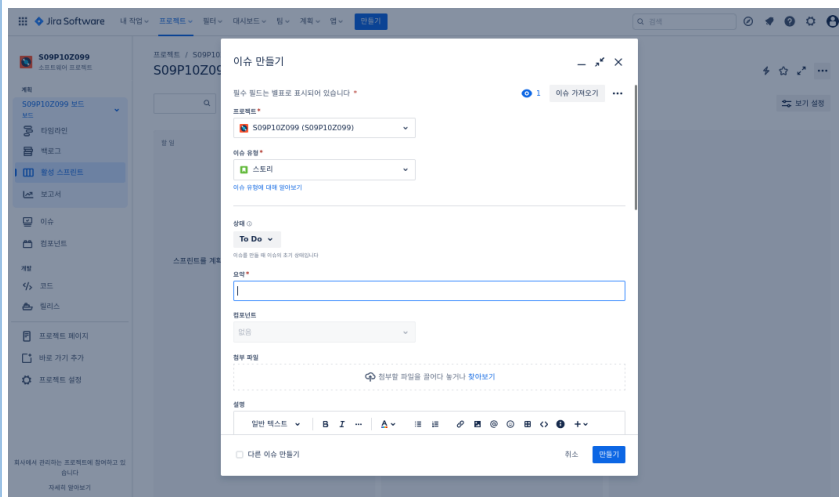
③ Jira 둘러보기

- 생성된 Jira URL을 클릭하여 Jira 프로젝트로 이동합니다.
- Jira에서 활용할 수 있는 기능들에 무엇이 있는지 살펴봅니다.



④ 이슈 등록하기

- [만들기] 버튼을 클릭하여 앞서 작성한 기능 정의서를 토대로 이슈를 스토리 유형으로 등록합니다.
- 인터페이스에 익숙해지는 목적으로 간단하게 등록을 진행하고, 다른 유형, 스토리 포인트, 스프린트 등에 대해서는 차차 배워가며 이후에 적용해보도록 합니다.



⑤ 이슈 목록 확인하기

- 백로그로 이동하여 등록된 이슈 목록을 확인합니다.

The screenshot shows the Jira Software interface for project 509P10Z099. The left sidebar contains navigation options like '509P10Z099 보드', '타임라인', '백로그', '할당 스프린트', '보고서', '이슈', '템플릿', '재방', '코드', '탈리스', '프로젝트 페이지', '바로 가기 추가', and '프로젝트 설정'. The main area displays the '백로그' (Backlog) for the project, showing a list of 6 issues. Each issue has a status of '대여할 할 일' (To Do) and a priority of '1'. The issues are: 509P10Z099-1 login, 509P10Z099-2 signup, 509P10Z099-3 logout, 509P10Z099-4 send email, 509P10Z099-5 reset password, and 509P10Z099-6 show items.

| 이슈 ID | 이슈 제목 | 상태 | 우선 순위 |
|--------------|----------------|---------|-------|
| 509P10Z099-1 | login | 대여할 할 일 | 1 |
| 509P10Z099-2 | signup | 대여할 할 일 | 1 |
| 509P10Z099-3 | logout | 대여할 할 일 | 1 |
| 509P10Z099-4 | send email | 대여할 할 일 | 1 |
| 509P10Z099-5 | reset password | 대여할 할 일 | 1 |
| 509P10Z099-6 | show items | 대여할 할 일 | 1 |

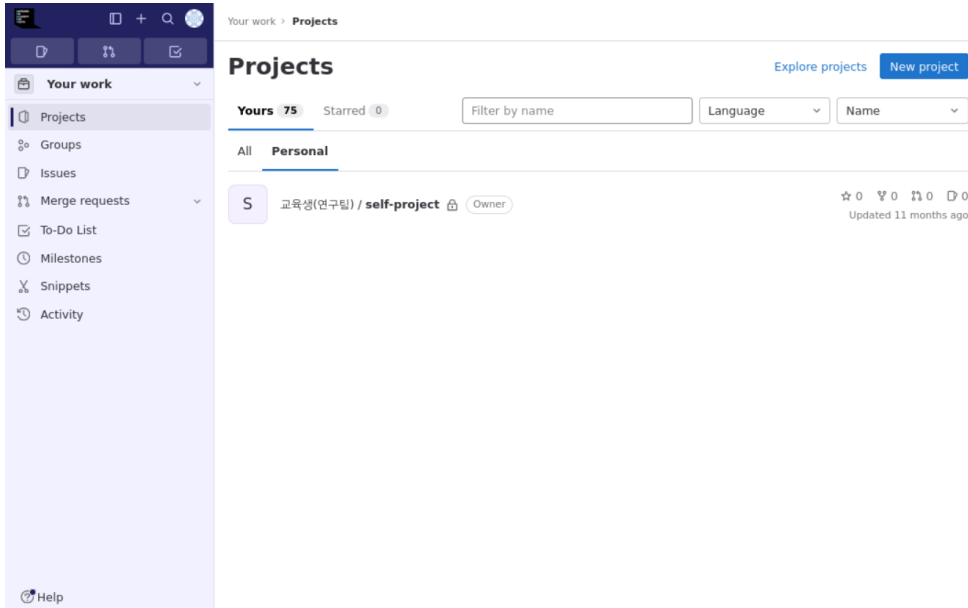
(4) 소스코드 관리

① GitLab 계정 확인

- GitLab은 SSAFY GIT과 SSO 연동이 되므로 SSAFY GIT 로그인 상태에서는 바로 접속 가능합니다.
 - SSAFY GIT의 이메일과 GitLab의 Primary Email이 같아야 연동 됩니다.
 - GitLab 로그아웃을 하려면 SSO 상태를 비활성화하기 위해 먼저 SSAFY GIT을 로그아웃한 후에 GitLab을 로그아웃해야 정상적으로 로그아웃됩니다.
- SSAFY GIT 우측 상단의 GitLab 바로가기 아이콘을 클릭하여 GitLab으로 이동합니다.



- GitLab 페이지가 올바르게 출력되는지 확인합니다.



② Personal Access Token 발급

- GitLab > 프로필 아이콘 > Preferences > Access Token
- Personal Access Token은 username과 password로 원격 인증하는 대신 사용자를 인증하게 해주는 Token입니다.
- Personal Access Token은 다음과 같은 장점이 있기에 사용을 권장합니다.
 - 개인의 password 노출 가능성을 최소화
 - 필요한 권한 범위 내에서, 지정된 기간에 한해 접근을 허용
 - 여러 개의 Token을 발급하여 각각 용도에 맞게 사용 가능
 - 유출, 분실 시 폐기 및 재발급 가능

- Access Token 메뉴로 이동하여 [Add new token] 버튼을 클릭합니다.

The first screenshot shows the GitLab user interface with the 'Preferences' menu item highlighted in the left sidebar. The second screenshot shows the 'Access Tokens' page with the 'Add new token' button highlighted.

- Token 이름과 만료일, 권한 범위를 입력/선택하고 [Create...] 버튼 클릭

The screenshot shows the 'Add a personal access token' form. The 'Token name' field contains 'develop'. The 'Expiration date' is set to '2024-01-10'. The 'Select scopes' section has several checkboxes selected: 'api', 'read_api', 'read_user', 'create_runner', 'read_repository', and 'write_repository'. The 'Create personal access token' button is highlighted.

- 생성된 Token 값은 한 번만 출력되니 별도로 기록해 둡니다.

③ GitLab 프로젝트(Repository) 생성

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- GitLab에서 사용자가 직접 프로젝트 생성이 가능하지만, 2학기 각 과정을 위한 프로젝트는 그룹핑과 권한 설정 등의 자동화를 위해 SSAFY GIT을 통해 생성해야 합니다.
- 프로젝트 설정 정보에서 GitLab 우측의 [생성] 버튼을 클릭합니다.
 - 모든 팀원의 SSAFY GIT 이메일과 GitLab 이메일 정보가 동일해야 생성됩니다.

프로젝트 설정 정보

| | | |
|----------|--------------------------------|------|
| JIRA | | [생성] |
| GitLab | Deploy Info User Name : Token | [생성] |
| Jenkins | | [생성] |
| Maria DB | User Info User Name : Password | [생성] |
| Mongo DB | User Info User Name : Password | [생성] |
| 실행 이미지 | URL : Backend URL | [생성] |
| | 실행 소스 | |

④ 생성된 GitLab 프로젝트로 이동

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- 생성된 GitLab URL을 클릭하여 GitLab 프로젝트로 이동합니다.

프로젝트 설정 정보

| | | |
|--------|---|------|
| JIRA | | [생성] |
| GitLab | https://lab.ssafty.com/s09-study/S9P06162001 : Token | [생성] |

- 프로젝트가 정상 생성되었음을 확인합니다.

9기 개발팀 > S9P06162001

S9P06162001
Project ID: S04172 Leave project

Invite your team
Add members to this project and start collaborating with your team.
[invite members]

The repository for this project is empty
You can get started by cloning the repository or start adding files to it with one of the following options.

[Clone] [Upload File] [New file] [Add README] [Add LICENSE] [Add CHANGELOG]
[Add CONTRIBUTING] [Set up CI/CD] [Add Wiki] [Configure integrations]

Command line instructions
You can also upload existing files from your computer using the instructions below.

Git global setup

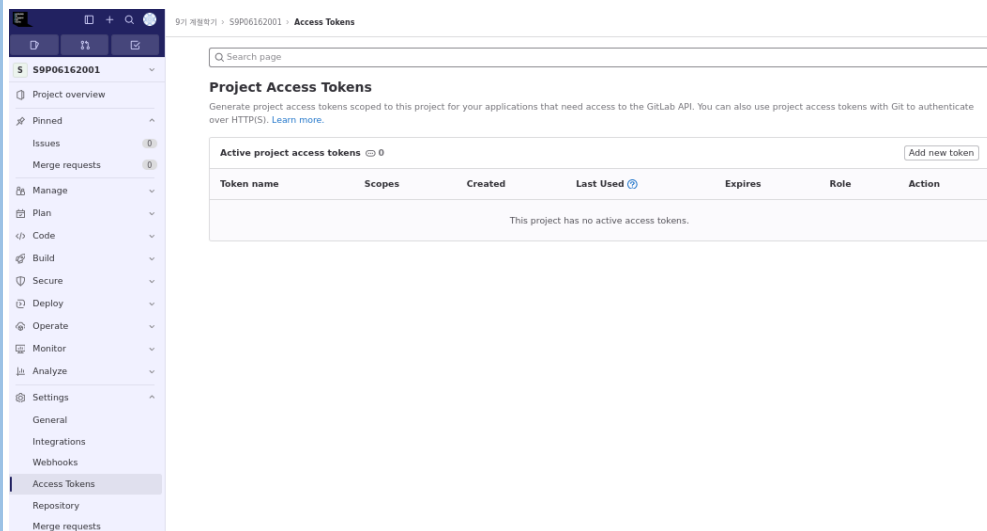
```
git config --global user.name "교육성(연구팀)"
git config --global user.email "project@ssafty.com"
```

Create a new repository

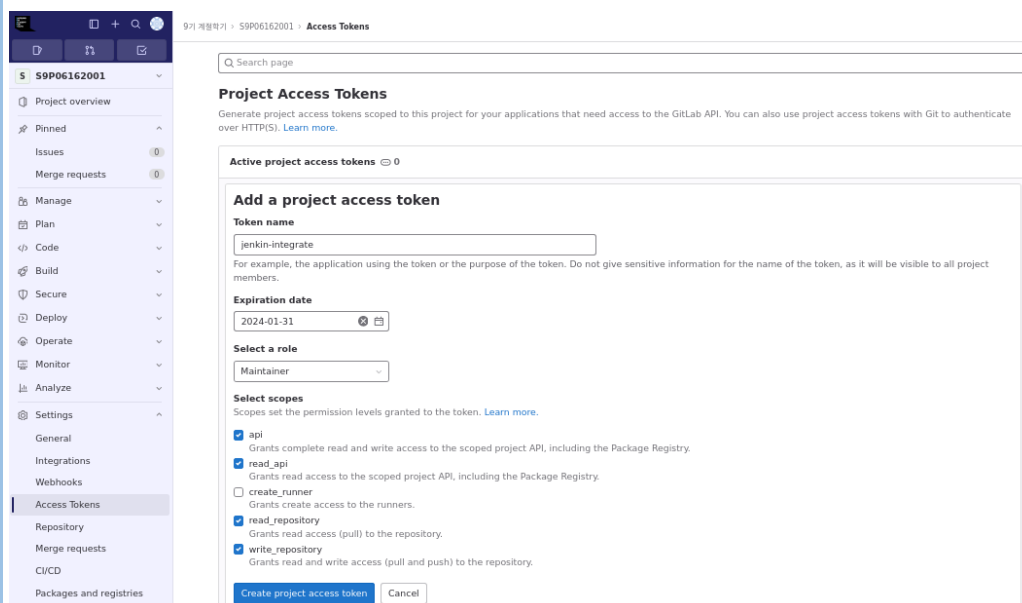
```
git clone https://lab.ssafty.com/s09-study/S9P06162001.git
cd S9P06162001
git switch --create master
touch README.md
```

⑤ Project Access Token 발급

- SSAFY GIT에서 생성된 GitLab 프로젝트 > Settings > Access Tokens
- Personal Access Token이 사용자 개인의 username/password를 대체하여 해당 사용자가 접근 가능한 범위 내에서 지정된 권한(Scope)의 접근을 허용하는 것인 반면, Project Access Token은 해당 프로젝트에 대한 원격 접속 권한을 얻는 것입니다.
- 일반적으로 이러한 Project Access Token은 사용자가 직접 사용하기보다는 다른 시스템으로부터 GitLab의 특정 프로젝트에 대한 권한만 허용할 필요가 있을 때에 사용됩니다.
- 프로젝트의 Access Token 메뉴에서 [Add new Token] 버튼을 클릭합니다.



- 필요한 정보들을 입력/선택하고 [Create...] 버튼을 클릭합니다.



- 생성된 Token 값은 한 번만 출력되니 별도로 기록해 둡니다.

⑥ 소스코드 업로드

- 생성한 프로젝트의 Repository에 1학기 종합 PJT 소스코드를 업로드합니다.
- 편의상 하나의 Repository에 모든 소스코드를 업로드합니다.(필요시 폴더로 구분)
- git 커맨드를 사용하거나 IDE 에디터 등에서 제공하는 기능을 사용할 수 있습니다..
- git 커맨드의 예

- PC에 git이 설치되어 있어야 합니다.(<https://git-scm.com> 참고)
- 프로젝트 최상위 경로에서 cmd 또는 git bash를 실행합니다.
- (git 관리 이력이 없다면) git 초기화 및 commit을 진행합니다.

```
> git init
> git config user.name <사용자명>
> git config user.email <이메일 주소>
> git add .
> git commit -m "initial commit"
```

- 위에서 생성된 프로젝트를 원격 Repository로 설정하고 업로드합니다.
- 업로드 인증 시에는 앞서 생성한 Personal Access Token을 password 대신 사용합니다.

```
> git remote add origin <프로젝트의 URL>
> git push -u origin main
```

(현재 브랜치가 master라면 main대신 master로 입력)

⑦ Repository 확인

- GitLab URL로 접속하여 Repository에 파일이 잘 업로드되었는지 확인합니다.

9기 개발팀 - S09P10Z099

S09P10Z099

Project ID: 504187

1 Commit 1 Branch 0 Tags 56 KIB Project Storage

Initial commit
교육영(연구팀) authored 9 minutes ago

b625f42b

main S09P10Z099 / +

History Find file Edit Clone

Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD Add Wiki

Configure integrations

| Name | Last commit | Last update |
|----------|----------------|---------------|
| backend | initial commit | 9 minutes ago |
| frontend | initial commit | 9 minutes ago |

⑧ branch 생성

- branch는 웹 인터페이스에서 생성할 수도 있고 로컬에서 생성 후 원격에 반영할 수도 있습니다.
- 로컬에서 git 커맨드 또는 IDE 등의 도구를 사용해 develop branch를 생성합니다.
- git 커맨드의 예
 - 프로젝트 최상위 경로에서 cmd 또는 git bash를 실행합니다.

```
> git checkout -b develop
```

⑨ README.md 파일 추가

- develop branch로 체크아웃된 상태에서 프로젝트 최상위 경로에 README.md 파일을 작성합니다.
 - 메모장, 마크다운 편집기, IDE 등 무엇을 사용해서 작성해도 무방합니다.
 - README.md의 내용은 프로젝트의 주제와 속성 등을 간단히 기술합니다.

⑩ 변경사항 반영

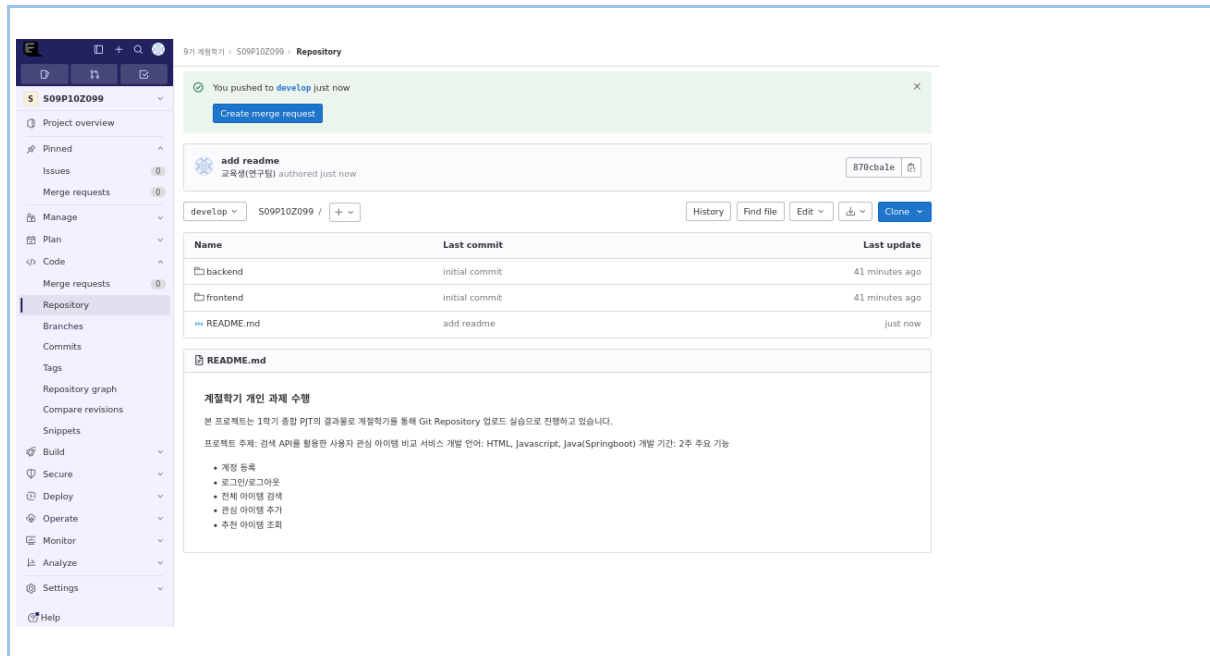
- develop branch의 변경사항을 반영합니다.
- git 커맨드의 예

```
> git add README.md
> git commit -m "add readme"
```

- commit 후 원격 Repository에도 업로드합니다.

```
> git push -u origin develop
```

- 이제 로컬과 원격 Repository에는 원본 상태의 main(또는 master) branch와 현재 수정 작업을 진행중인 develop branch가 존재하는 것을 확인할 수 있습니다.



(5) 빌드 관리

① Gitlab token 입력

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- GitLab 우측의 [수정] 버튼을 클릭하여 앞서 생성한 Project access token 정보를 입력합니다.

프로젝트 설정 정보

JIRA <https://ssafy.atlassian.net/projects/S09P10Z099> 수정

Gitlab 수정

Jenkins 생성

Maria DB 생성

Mongo DB 생성

실명 이미지 생성

URL 생성

Backend URL

실명 소스

GITLAB 수정

Gitlab

Deploy Info User Name : Token

Deploy token 생성 방법보기

취소 저장

② Jenkins 생성

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- Jenkins 우측의 [생성] 버튼을 클릭합니다.
 - 앞서 선택한 프로젝트 기본 정보의 기술 입력값과 위에 입력한 Deploy info를 바탕으로 Jenkins job이 생성됩니다.

프로젝트 설정 정보

JIRA <https://ssafy.atlassian.net/projects/S09P10Z099> 수정

Gitlab <https://lab.ssafy.com/s09-study/S09P10Z099> 수정

Deploy Info User Name : Token

Jenkins 생성

Maria DB User Info User Name : Password 생성

Mongo DB User Info User Name : Password 생성

실명 이미지 생성

URL 생성

Backend URL

실명 소스

③ 빌드 진행

- Jenkins URL이 생성되면 빌드 기능이 활성화됩니다.
- SSAFY GIT > 프로젝트 개발 > 빌드현황으로 이동합니다.

SSAFY GIT 프로젝트 개발 프로젝트 현황 평가 Play Help

계절학기-자기주도PJT

My Project

계절학기-자기주도PJT

2023.06.26 ~ 2023.06.30(총 1주)

공통 개발하기

공통 Sub PJT I

공통 Sub PJT II

공통 Sub PJT III

독립 전자 연계 프로젝트

독립 Sub PJT I

독립 Sub PJT II

독립 Sub PJT III

자율 자율 프로젝트

홈 이슈현황 리뷰현황 빌드현황 산출물 팀 정보

빌드현황

마지막 빌드 대기

2023.12.18 00:16:53 #5

마지막 성공 빌드

2023.12.18 00:15:51 #4

마지막 실패 빌드

2023.12.18 00:15:01 #3

마지막 변경

-

빌드결과

| No | 일시 | 시간 | 결과 |
|----|---------------------|-----|----|
| 5 | 2023.12.18 00:16:53 | 0초 | 대기 |
| 4 | 2023.12.18 00:15:51 | 11초 | 성공 |
| 3 | 2023.12.18 00:15:01 | 2초 | 실패 |
| 2 | 2023.12.18 00:13:01 | 2초 | 실패 |
| 1 | 2023.12.18 00:11:10 | 1초 | 실패 |

변경현황

| No | 일시 | 내용 | 결과 | 담당자 |
|------------|----|----|----|-----|
| 데이터가 없습니다. | | | | |

< 이전 1 다음 >

- [빌드] 버튼을 클릭하고 결과를 기다립니다.
- 성공 또는 실패 결과를 확인합니다.

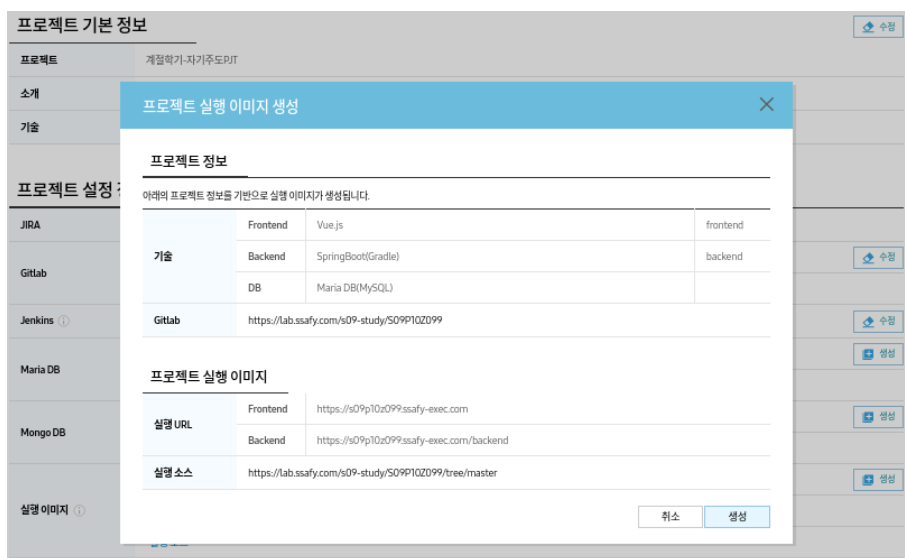
소스 코드의 개발 환경에 따라 빌드 구성 방법이 상이하므로, 시스템에 기본값으로 설정된 빌드는 실패할 수도 있습니다. 따라서 실패하더라도 결과의 로그를 확인하고 CI/CD에 대한 개념만 이해한 채 다음으로 넘어가도 됩니다.(이후 설정을 수정하여 정상적인 빌드 수행을 가능하게 할 수 있습니다.)

(6) 실행 기능

현재 SSAFY GIT의 실행 기능은 제한된 기술 스택의 Web 개발 산출물만 지원하고 있습니다. 따라서 프로젝트 정보에서 선택할 수 없는 기술의 프로젝트이거나 실행 로그에 오류가 발행하는 경우에는, 기능과 설정 방법만은 익히고 넘어가도록 합니다.

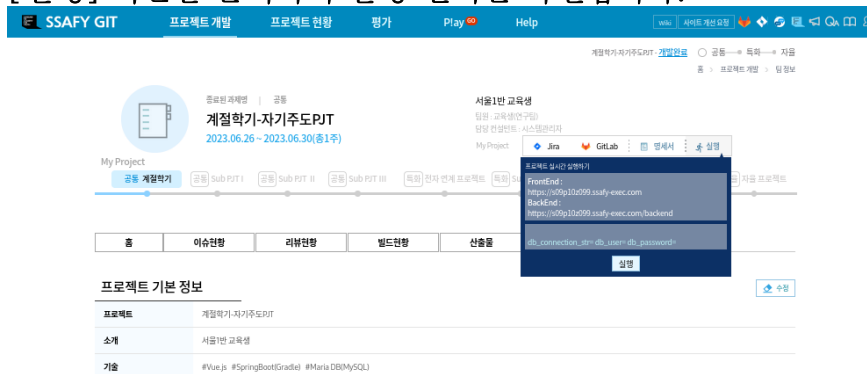
① 실행 이미지 생성

- SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보
- 실행 이미지 우측의 [생성] 버튼을 클릭합니다.



② 프로젝트 실행

- SSAFY GIT > 프로젝트 개발 > 실행
- [실행] 버튼을 클릭하여 실행 결과를 확인합니다.



- 실행 팝업에서 빌드 대기 시간을 거쳐 실행 화면이 출력됩니다.
- 단, 빌드에 오류가 있는 경우 실행은 중단되며, 오류 내용은 좌측 로그 탭에서 확인할 수 있습니다.

제철학기-자기주도PJT
9기 | 서울1반 교육생

| | | |
|-------|---|---|
| URL | FrontEnd https://s09p10z099ssafy-exec.com | : BackEnd https://s09p10z099ssafy-exec.com/backend |
| 소스 코드 | https://lab.ssafy.com/s09-study/S09P10Z099/tree/b025f42b64fd9231f7d635d22f182775f9f62e2 | |
| 실행 변수 | db_connection_str= db_user= db_password= | |

로그
→
←
<https://s09p10z099ssafy-exec.com>

Step 2/4 : COPY ./dist ./usr/share/nginx/html
COPY failed: file not found in build context or excluded by .dockerignore: stat dist: file does not exist
[Pipeline]
[Pipeline] // dir
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] {(SERVER build)
Stage "SERVER build" skipped due to earlier failure(s)
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] {(Clear)
Stage "Clear" skipped due to earlier failure(s)
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
Build | FrontEnd | BackEnd

빌드 에러 (FAILURE)

(7) 정리 단계

① Merge

- GitLab에서 진행
- develop branch를 main branch로 병합하는 Merge Request를 생성합니다.
- Merge Request에서 Changes를 확인하고 확인했다는 comment를 남깁니다.
- develop branch를 main branch로 병합하고 develop branch는 삭제합니다.

② 이슈 정리

- Jira에서 진행
- 생성한 모든 이슈의 상태를 done(완료됨)으로 변경합니다.

③ SSAFY GIT 둘러보기

- 기타 다른 메뉴와 게시판 등 SSAFY GIT의 여러 기능들을 살펴봅니다.

5. 심화 과제

1) 빌드 설정

프로젝트의 빌드는 개발 환경에 따라 상이합니다. 따라서 시스템의 "기본 설정 상태"로만 진행한 일부 과제는 결과가 실패로 끝났을 것입니다.

심화 과제로 여러분의 프로젝트 구조와 환경을 이해하고 각 시스템의 설정 방법을 확인하여 성공적인 결과가 나올 수 있도록 설정을 변경하여 재시도해보시기 바랍니다.

● SSAFY GIT > 프로젝트 개발 > 빌드 현황

- [설정] 버튼을 클릭하여 branch, execute shell 항목을 확인하고 알맞게 수정
- 빌드 결과 최근 시도가 성공이 되도록 시도

빌드결과

| No | 일시 | 시간 | 결과 |
|----|---------------------|-----|----|
| 5 | 2023.12.18 00:17:01 | 5초 | 성공 |
| 4 | 2023.12.18 00:15:51 | 11초 | 성공 |
| 3 | 2023.12.18 00:15:01 | 2초 | 실패 |
| 2 | 2023.12.18 00:15:01 | 2초 | 실패 |
| 1 | 2023.12.18 00:11:10 | 1초 | 실패 |

변경현황

| No | 일시 | 내용 | 결과 | 담당자 |
|------------|----|----|----|-----|
| 데이터가 없습니다. | | | | |

2) SSAFY GIT 제공 DB 활용

SSAFY GIT에서 제공하는 MariaDB, MongoDB를 사용하면 로컬 개발 환경이 갖는 단점을 보완할 수 있습니다.

● SSAFY GIT > 프로젝트 개발 > 팀 정보 > 프로젝트 설정 정보

- 필요한 유형의 DB를 생성합니다.
- 1학기 종합 PJT의 DB를 생성한 DB로 이관합니다.
- 소스코드의 DB 정보를 SSAFY GIT DB로 업데이트하고 실행합니다.