**과제 제출 시스템 개발 보고서**

**과목 : 딥러닝 프로그래밍 활용**

**한국폴리텍대학**

**서울정수캠퍼스**

**인공지능소프트웨어과**

**2401110268**

**차기환**

**1. 개발 배경 및 목적 🧐**

본 프로젝트는 한국폴리텍대학 인공지능소프트웨어과 학생들의 원활한 과제 제출 및 관리를 위한 웹 기반 시스템을 구축하는 것을 목표로 합니다.

기존에 사용하던 Microsoft Teams는 편리한 협업 도구이지만, 과제 제출 시스템으로 활용될 경우 발생하는 **용량 문제**로 인해 많은 학생들이 불편을 겪었습니다. 특히, 대용량의 프로젝트 파일이나 설계 문서 제출 시 용량 제한에 걸려 과제를 제출하지 못하는 사례가 빈번하게 발생했습니다.

이러한 문제를 해결하기 위해, Django 프레임워크를 기반으로 교수자와 학생 모두가 효율적으로 사용할 수 있는 **자체 과제 제출 시스템**을 개발하게 되었습니다. 이 시스템을 통해 학생들은 용량 걱정 없이 과제를 제출하고, 교수자는 수강생의 과제를 체계적으로 관리 및 평가할 수 있습니다.

**2. 주요 기능 ✨**

본 시스템은 별도의 교수자/학생 구분 없이, **모든 가입자가 자유롭게 팀을 생성하거나 참여**할 수 있는 유연한 권한 구조를 가집니다. 사용자는 시스템 내에서의 행동에 따라 다음과 같이 두 가지 주요 역할을 수행하게 됩니다.

* **팀 생성자 (Owner)**: 강의나 스터디 그룹에 해당하는 '팀'을 직접 만드는 사용자입니다.
* **팀 참여자 (Member)**: 다른 사용자가 만든 팀에 코드를 통해 가입을 요청하고 참여하는 사용자입니다.

각 역할에 따라 사용할 수 있는 핵심 기능은 다음과 같습니다.

**2.1. 팀 생성자(Owner) 기능**

팀을 직접 생성한 사용자는 해당 팀의 관리자로서 다음과 같은 권한을 가집니다.

* **팀 생성 및 관리**:
  + 자신이 관리할 강의나 그룹별로 '팀'을 생성하고, 설명과 대표 이미지를 등록할 수 있습니다.
  + 팀의 정보(이름, 설명, 이미지)를 언제든지 수정하거나, 팀을 삭제할 수 있습니다.
* **참여자 가입 요청 관리**:
  + 사용자들이 팀 코드를 통해 보낸 가입 요청을 **승인**하거나 **거절**할 수 있습니다.
  + 이 기능은 실제 강의 수강생 등 허가된 사용자만 선별하여 팀에 참여시키기 위한 핵심적인 역할을 합니다.
* **과제 등록 및 관리**:
  + 팀에 소속될 과제를 등록하고, 제목, 설명, 마감일시, 배점 등을 자유롭게 설정할 수 있습니다.
  + 필요에 따라 진행 중인 과제를 **수동으로 마감**하거나, 마감된 과제를 **다시 열어** 추가 제출을 허용하는 등 유연하게 과제를 관리할 수 있습니다.
* **제출 현황 확인 및 채점**:
  + 팀 참여자들이 제출한 과제 현황을 목록으로 한눈에 파악하고, 제출된 파일을 직접 다운로드하여 확인할 수 있습니다.
  + 개별 제출물에 대해 **점수를 부여**하고 **피드백**을 남길 수 있는 채점 기능을 제공합니다.

**2.2. 팀 참여자(Member) 기능**

다른 사용자가 생성한 팀에 참여하는 사용자는 다음과 같은 기능을 이용할 수 있습니다.

* **회원가입 및 로그인**:
  + 학번, 이름, 이메일 등의 정보를 입력하여 간편하게 회원가입을 할 수 있으며, 로그인하여 시스템을 이용할 수 있습니다.
* **팀 참가**:
  + 팀 생성자에게 공유받은 **6자리 팀 코드**를 입력하여 팀에 가입을 요청할 수 있습니다.
  + 자신이 보낸 참가 요청의 상태(대기, 승인, 거절)를 실시간으로 확인할 수 있습니다.
* **과제 제출**:
  + 가입이 승인된 팀에 등록된 과제를 확인하고, 파일을 첨부하여 제출할 수 있습니다.
  + 제출 시 간단한 코멘트를 남길 수 있으며, 과제가 마감되기 전까지는 언제든지 **수정하여 다시 제출**할 수 있습니다.
* **제출 내역 및 성적 확인**:
  + 과제별로 자신의 제출 상태(미제출, 제출됨, 채점 완료 등)와 제출 시간, 제출한 파일 목록을 확인할 수 있습니다.
  + 채점이 완료되면 자신의 점수와 팀 생성자가 남긴 피드백을 확인할 수 있습니다.

**3. 시스템 구조 및 개발 환경 💻**

* **개발 언어**: Python
* **웹 프레임워크**: Django
* **데이터베이스**: SQLite
* **프론트엔드**: HTML, Tailwind CSS
* **애플리케이션 구조**:
  + config: 프로젝트의 전반적인 설정을 관리합니다. (settings.py, urls.py)
  + submit: 핵심 비즈니스 로직을 담고 있는 메인 앱입니다. (models.py, views.py, admin.py)
  + templates: 사용자가 보는 웹 페이지의 HTML 파일들을 관리합니다.
  + static: 로고 이미지 등 정적 파일을 관리합니다.
  + media: 사용자가 업로드하는 파일(팀 대표 이미지, 과제 제출 파일 등)이 저장됩니다.

**4. 데이터베이스 설계 (ERD) 🧱**

시스템의 데이터는 Django의 ORM(Object-Relational Mapping)을 통해 관리되며, 주요 모델 간의 관계는 다음과 같습니다.

* **User**: Django의 기본 User 모델을 사용하여 사용자 인증 및 정보를 관리합니다. 팀 생성자와 참여자 모두 이 모델을 사용합니다.
* **StudentProfile**: User 모델을 확장하여 학생의 '학번' 정보를 추가로 저장합니다. (User와 1:1 관계)
* **Team**: 강의에 해당하는 그룹입니다. 한 명의 owner(팀 생성자, User 모델)를 가집니다.
* **TeamMembership**: Team과 student(팀 참여자, User 모델)를 연결하는 중간 모델로, 참여자의 팀 가입 상태(대기, 승인 등)를 관리합니다. (Team, User와 다대다 관계)
* **Assignment**: Team에 종속되는 과제 정보입니다. (Team과 1:N 관계)
* **Submission**: Assignment에 대해 student가 제출한 정보를 담습니다. (Assignment, User와 N:1 관계)
* **SubmissionFile**: Submission에 첨부된 개별 파일을 관리합니다. (Submission과 1:N 관계)
* **Grade**: Submission에 대한 채점 정보(점수, 피드백)입니다. (Submission과 1:1 관계)

**5. 결론 및 향후 개선 방향 🚀**

본 프로젝트를 통해 기존 Teams의 용량 문제를 해결하고, 사용자가 자유롭게 팀을 생성하고 참여할 수 있는 유연한 과제 제출 및 관리 환경을 제공할 수 있었습니다. 특히, Django의 강력한 ORM과 템플릿 시스템을 활용하여 짧은 시간 안에 안정적이고 확장 가능한 웹 애플리케이션을 구축할 수 있었습니다.

향후 아래와 같은 기능을 추가하여 시스템을 더욱 발전시킬 수 있을 것입니다.

* **알림 기능**: 과제 마감 임박, 채점 완료 등 주요 이벤트 발생 시 사용자에게 알림을 보내는 기능.
* **대시보드 고도화**: 마감 임박 과제, 미제출 과제 현황 등을 시각적으로 보여주는 대시보드 기능 강화.
* **파일 미리보기**: 제출된 문서를 다운로드 없이 웹에서 바로 확인할 수 있는 기능.
* **관리자 기능 강화**: 전체 팀, 사용자, 과제 현황을 모니터링할 수 있는 관리자 전용 페이지 구축.

**6. 개발 후기 및 느낀 점**

이번 프로젝트를 진행하며 Django 프레임워크의 강력한 기능과 개발 생산성을 직접 체감할 수 있었습니다. 특히 Django ORM을 통해 복잡한 데이터베이스 관계를 파이썬 코드로 손쉽게 정의하고 관리할 수 있었던 점이 인상 깊었습니다.

초기에는 사용자 권한을 '교수'와 '학생'으로 명확히 나누는 것을 고려했지만, 누구나 팀을 만들고 참여할 수 있는 유연한 구조로 변경하면서 서비스의 확장 가능성을 높일 수 있었습니다. 이 과정에서 모델 간의 관계를 재설계하고 TeamMembership의 역할을 정의하는 것이 중요했습니다.

다만, 프론트엔드 개발에 Tailwind CSS를 사용하면서 일부 동적인 UI(예: 실시간 알림)를 구현하는 데에는 한계가 있었습니다. 향후 JavaScript나 React 같은 기술을 접목한다면 사용자 경험을 더욱 향상시킬 수 있을 것이라 생각합니다.