**모바일/웹서비스 프로젝트 공통평가 01**

**수행 결과 보고서**

**학번 : 2022100631**

**이름 : 김도영**

[적/부판정 범례] ○:적합, ×:부적합, △:점검제외, N/A:해당없음

첨부 그림은 모두 문서에 포함하는 형식으로 저장 할 것

| **수업 요구사항** | | | | **이행현황** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **구분** | **요구사항명** | **요구사항 내용** | **완료**  **여부** | **적/부 판정** | **관련증빙** |
| 1 | 제출문 | 폴더 구조 | - ROOT  - PhotoBlogServer[Dir]  : Django Server  - PhotoViewer[Dir]  : Android Client  -모바일/웹서비스 프로젝트 공통평가 01\_수행 결과 보고서.docx[File]  제출 : 탐색기, 파인더 또는 Git레포지토리 폴더 구조 캡처 화면 | O |  |  |
| 2 | 제출문 | Github 레파지토리 push | 아래 평가 항목에 대하여 구현 혹은 테스트를 완료하고, 캡처 이미지를 제공된 MS-Word 파일로 정리하여 Github URL을 제출 한다.  제출 : URL | O |  | <https://github.com/doyoung0013/photo_blog> |
| 3 | 강의 자료 | 포토 블로그 – API | 강의 자료 [7페이지, POST]  제출 : 1. 작성 완료 웹페이지 작성 화면 캡처  2. 게시 완료된 페이지 화면 캡처 | O |  | 1.웹페이지 작성 화면 캡처  2. 게시 완료된 페이지 화면 캡처 |
| 4 | 강의 자료 | 포토 블로그 – API 검증(2/3) | 강의 자료 [9페이지]  제출 : 터미널 화면 캡처 | O |  |  |
| 5 | 강의 자료 | 포토 블로그 – API 검증(3/3) | 강의 자료 [10페이지]  제출 : [codebeautify](https://codebeautify.org/jsonviewer) 서비스 테스트 화면 캡처 | O |  |  |
| 6 | 강의 자료 | Server/Client 연동 테스트 [Sockec Server 요청 메시지] | 강의 자료 [24페이지]  제출 : 수신 파일 VSCODE 편집화면캡처 | O |  |  |
| 7 | 강의 자료 | Server/Client 연동 테스트 [Server 브라우저 출력화면] | 강의 자료 [25페이지]  제출 : Server 브라우저 출력화면 (localhost) | O |  |  |
| 8 | 강의 자료 | Server/Client 연동 테스트 [Client 시작 화면] | 강의 자료 [25페이지]  제출 : Client 시작 화면 | O |  |  |
| 9 | 강의 자료 | Server/Client 연동 테스트 [Client 동기화 완료 화면] | 강의 자료 [25페이지]  제출 : Client 동기화 완료 화면(localhost 서버연동) |  |  |  |
| 10 | 강의 자료 | MainActivity 클래스 [Upload 구현, 19page] | MainActivity 클래스 [Upload 구현, 19page]  1. Upload (Hard cording) : 중  (1. 2. 중 하나만 제출 ) |  |  |  |
| MainActivity 클래스 [Upload 구현, 19page]  2. Upload (Post에 따른 UI제공) : 상  (1. 2. 중 하나만 제출 | O |  | - PutPost 부분 |
| 11 |  | Pythonanywhere 클라우드상 동작 | Pythonanywhere 클라우드상 서버 동작  제출 : Server 브라우저 출력화면 (클라우드 서버연동) | O |  |  |
| 12 |  | Pythonanywhere 클라우드상 동작 | Pythonanywhere 클라우드상 서버 동작  제출 : Client 동기화 완료 화면 (클라우드 서버연동) | O |  |  |
| 13 |  | 기타 기획 되지 않은 Client 편의 기능 [최대 5개] | 제출 : 해당 기능 정의, 설명 및 동작 증빙 화면 캡처  (서버/클라이언트 각각 구현이 필요한 경우 별도 요구사항으로 등록 함) | O |  | 1. 좋아요 기능 (클라이언트) : 게시물 상세 페이지에서 로그인 없이 좋아요(하트) 클릭 시, 해당 게시글 좋아요 수가 1씩 증가한다. 기존에는 좋아요 기능이 존재하지 않아 사용자는 단순히 게시물을 조회하거나 작성하는 데에 그쳤으나, 해당 기능을 추가함으로써 사용자 참여도를 높이고 콘텐츠 반응성을 강화할 수 있다. |
|  |  |  |  | O |  | 2. 익명 댓글 기능 (클라이언트) : 로그인 없이 닉네임만 입력하여 댓글을 작성할 수 있는 기능이다. 기존에는 댓글 기능이 없었으나, 이를 추가함으로써 비회원 사용자도 간편하게 의견을 공유할 수 있게 되었으며, 이를 통해 참여율 향상 및 오픈형 커뮤니티 구축이 가능하다. |
|  |  |  |  | O |  | 3. 익명 댓글 기능 (서버) : 같은 사용자가 10초 이내에 중복 댓글을 작성하지 못하도록 IP 기반 댓글 작성 쿨타임 제한 알고리즘을 적용하였다. 이를 통해 스팸성 댓글 및 도배 행위를 효과적으로 방지할 수 있으며, 서버 트래픽의 안정성을 확보하여 시스템 성능을 개선하였다. |
|  |  |  |  | O |  | 4. 게시물 정렬 기능 (클라이언트) : 사용자가 정렬 기준을 직접 선택해 게시물을 최신순 또는 좋아요순으로 정렬할 수 있다. 기존에는 사용자가 게시물이 어떤 순서로 나열되어 있는지 알 수 없었으나, 해당 기능 추가로 사용자 맞춤형 탐색이 가능해졌고, 좋아요 기준으로 인기 게시물을 중심으로 피드 흐름을 유연하게 구성할 수 있다. |
|  |  |  |  | O |  | 5. 제목 검색 기능 (클라이언트) : 상단 검색창에 키워드를 입력하면 게시물 제목에 해당 단어가 포함된 게시물만 필터링하여 표시된다. 기존에는 전체 게시물을 스크롤하며 찾아야 했으나, 이 기능을 통해 키워드 기반의 즉시 검색이 가능해져 탐색 효율과 사용자 편의성이 향상되었다. |