

# 개발 완료 보고서

제출일 : 2023년 2월 12일

팀명		참여 인원																																																																																																																																																																																																													
의명박		김명은, 박규환, 박의용																																																																																																																																																																																																													
개발프로젝트 소개																																																																																																																																																																																																															
프로젝트 명		Education Application																																																																																																																																																																																																													
활동일시		2/6 ~ 2/12		장소		광주인력개발원 공학1관 드론융합실																																																																																																																																																																																																									
주요 주제		<p>공공데이터를 활용하여 서버 클라이언트 구조의 '대한민국 100대 명산' 학습 프로그램을 구현한다.</p> <p>학생용 클라이언트는 '100대 명산'에 관한 문제를 풀고 직접적으로 학습한다.</p> <p>문제를 다 푼 후 점수에 따라 point를 제공받고 학습자의 실력을 점검한다.</p> <p>또한 학습 진행내용을 저장 및 불러오기 할 수 있다.</p> <p>교사용 클라이언트는 '100대 명산'에 관한 문제를 작성하고 문제를 학생이 일정이상 맞추면 단계등급을 부여해 준다.</p> <p>학생과 교사 클라이언트는 실시간 상담을 할 수 있고, Q&amp;A로 질문과 답변을 할 수 있다.</p>																																																																																																																																																																																																													
개발목적		<ul style="list-style-type: none"><li>- 공공데이터 API 활용 방법 학습</li><li>- TCP/IP를 통한 클라이언트, 서버간 유기적 네트워크 형성</li><li>- 클라이언트와 서버 각각의 역할 숙지</li></ul>																																																																																																																																																																																																													
파트별 임무분담		김명은	교사용 클라이언트 구현																																																																																																																																																																																																												
		박규환	서버 파트 구현																																																																																																																																																																																																												
		박의용	학생용 클라이언트 구현																																																																																																																																																																																																												
개발환경		Windows OS / pycharm / Qt Designer / MySQL / GitHub																																																																																																																																																																																																													
일정표		<div><div>필수</div><div>실습실 퇴근 시간 : 휴일 포함 매일 21시</div></div> <table><tr><th colspan="2">학습 일정표</th><th colspan="2">2월 6일(월)</th><th colspan="2">2월 7일(화)</th><th colspan="2">2월 8일(수)</th><th colspan="2">2월 9일(목)</th><th colspan="2">2월 10일(금)</th><th colspan="2">2월 11일(토)</th><th colspan="2">2월 12일(일)</th></tr><tr><td rowspan="6">분류</td><td>회의</td><td>06-12시</td><td>12-18시</td><td>18-24시</td><td>06-12시</td><td>12-18시</td><td>18-24시</td><td>06-12시</td><td>12-18시</td><td>18-24시</td><td>06-12시</td><td>12-18시</td><td>18-24시</td><td>06-12시</td><td>12-17시</td></tr><tr><td rowspan="5">학습</td><td>개발 계획서 작성 및 제출</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>git branch 설정</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>DB 선정</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>서버구현</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>클라이언트구현</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>디버깅, 회합, 보고서 작성 등</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td colspan="2">프로젝트 완료 보고서 제출</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td colspan="2">개인 일정</td><td colspan="2">2/6</td><td colspan="2">2/7</td><td colspan="2">2/8</td><td colspan="2">2/9</td><td colspan="2">2/10</td><td colspan="2">2/11</td><td colspan="2">2/12</td></tr><tr><td colspan="2">김명은</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td></tr><tr><td colspan="2">박규환</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">차량경사(오전 11시-오후 2시)</td><td colspan="2">특이사항없음</td></tr><tr><td colspan="2">박의용</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td><td colspan="2">특이사항없음</td></tr></table>										학습 일정표		2월 6일(월)		2월 7일(화)		2월 8일(수)		2월 9일(목)		2월 10일(금)		2월 11일(토)		2월 12일(일)		분류	회의	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-17시	학습	개발 계획서 작성 및 제출														git branch 설정														DB 선정														서버구현														클라이언트구현														디버깅, 회합, 보고서 작성 등														프로젝트 완료 보고서 제출															개인 일정		2/6		2/7		2/8		2/9		2/10		2/11		2/12		김명은		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		박규환		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		차량경사(오전 11시-오후 2시)		특이사항없음		박의용		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음	
학습 일정표		2월 6일(월)		2월 7일(화)		2월 8일(수)		2월 9일(목)		2월 10일(금)		2월 11일(토)		2월 12일(일)																																																																																																																																																																																																	
분류	회의	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-18시	18-24시	06-12시	12-17시																																																																																																																																																																																																
	학습	개발 계획서 작성 및 제출																																																																																																																																																																																																													
		git branch 설정																																																																																																																																																																																																													
		DB 선정																																																																																																																																																																																																													
		서버구현																																																																																																																																																																																																													
		클라이언트구현																																																																																																																																																																																																													
디버깅, 회합, 보고서 작성 등																																																																																																																																																																																																															
프로젝트 완료 보고서 제출																																																																																																																																																																																																															
개인 일정		2/6		2/7		2/8		2/9		2/10		2/11		2/12																																																																																																																																																																																																	
김명은		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음																																																																																																																																																																																																	
박규환		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		차량경사(오전 11시-오후 2시)		특이사항없음																																																																																																																																																																																																	
박의용		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음		특이사항없음																																																																																																																																																																																																	

요구분석서	6팀 의명박 팀 요구사항분석서 - 데이터베이스 활용 프로그램									
	번호	유형	요약	요구사항	요구분석 내용				정	부
	1	Server	Program Server	회원 관리	회원가입 기능을 통해 교사, 학생을 분류하여 DB에 User 정보 생성	박규환	박의웅	김명은		
					회원 가입시 이름, 회원분류, 전화번호의 정보를 받음					
					각 User의 ID, PW를 Check하여 일치할 경우 Login 처리					
					User의 Logout시 Logout 처리					
				권한 관리	회원 분류에 따른 권한 부여					
				접속 관리	회원 분류에 따른 Client UI 출력 지시					
					API를 활용하여 Client에서 요청하는 DB 전송					
				문제 관리	교사가 출제한 문제와 답안을 교사 Client로부터 받아 DB에 저장					
					학생 Client에서 문제풀이 요청시 Random 문제 출제					
					학생 Client에서 답안 입력시 문제와 답안의 일치 여부 확인 및 결과를 학생 Client에 전송					
					학생의 문제풀이 결과와 Point를 DB에 저장					
				질문 게시판	질문 게시판에 학생이 올린 게시물과 교사가 올린 답변을 DB에 저장					
					Client가 질문 게시판에 접근시 DB에 저장된 질문과 답변을 Client에 전송					
	2	클라이언트	학생용 클라이언트	1. 퀴즈, 문제 풀기 2. 문제별 point 제공 3. 질문과 답변(Q&A) 4. 상담 요청 5. 학습 진행 내용 확인	1. 주제에 맞게 UI가 구성이 되면 문제 풀기	박의웅	박규환			
					2. 직접적으로 학습하는 요구사항에 맞게 구성해야 한다. 흥미를 가질 수 있어야 한다.					
					2. 문제별 점수가 있으므로 점수에 맞게 POINT 제공					
					3. 문제를 푸는 페이지가 따로 존재하고, 문제를 다 푼 후에 점수를 제공한다 ( 점수에 맞게 포인트 제공 )					
					3. 질문과 답변 게시판이 존재함 ( UI 구성 요소 )					
					4. 질문하고 답변 할 수 있게 구성 한다.					
					4. 상담시간에 상담을 제공한다. 상담요청시 생성이 받아들이면 상담 가능					
					5. 학습 진행내용을 저장 및 불러오기 기능이 있다.					
					2. 예를 들어서 문제가 10번까지 있는데, 문제를 다 안 푼고 저장할 수 있고, 현재 문제풀이까지 불러오기.					
					1. 문제를 UPDATE할 수 있다. ( 서버에 문제 출제 후 보낸다 )					
					2. '학습주제' 데이터 서버에 요청, '주제'에 대한 데이터를 받아서 문제 출제후 해당 데이터와 함께 서버 전송					
					2. 서버에 학생들 문제 푼 데이터를 요청하여 받아와서 점수 확인한다.					
코드 설명	김명은				3. 학생 문제 데이터를 요청한 후 해당 데이터를 테이블 위젯이든 확인가능하게 띄운다	김명은	박규환			
					3. 문제 푼 데이터에는 오답률, 시간이 얼마걸렸는지 데이터가 존재한다. ( 2번과 밀접한 연관이 있음 )					
					4. 질문과 답변 게시판에 답글을 작성한다 ( UI 구성 요소 )					
					5. 학생이 상담을 요청하면 받아들이고 실시간 상담이 이루어 진다.					
	3	클라이언트	교사용 클라이언트	1. 문제 출제 (update) 2. 학생들 문제 점수 확인 3. 문제별 오답률,시간 확인 4. 질문과 답변 답글 가능 5. 실시간 상담기능	1. 문제 출제 (update)	김명은	박규환			
					2. 학생들 문제 점수 확인					
					3. 문제별 오답률,시간 확인					
					4. 질문과 답변 답글 가능					
					5. 실시간 상담기능					

		<p><b>def receive_message(self, so):</b>          서버에게로부터 필요한 정보들을 받는 함수 / 로그인, 회원가입, 학생의 질문, 학생의 성적, 실시간상담 등          식별자와 정보를 서버에게로부터 받은 결과를 PyQt5에 반영되게 하는 함수</p>
	박규환	<p><b>class ServerTool:</b> 비교적 범용성 있게 사용 가능한 함수들 클래스</p> <ol style="list-style-type: none"> <li>1. <b>socket_initialize(ip, port):</b> ip와 port를 인수로 받아 소켓 설정, 오픈 후 리스트에 담음. 포트가 열렸음을 콘솔에 출력하고 (소켓 리스트, 설정한 소켓) 반환.</li> <li>2. <b>add_client_to_socket_list(서버 소켓, 소켓 리스트):</b> 서버 소켓에 접속한 클라이언트 소켓 정보를 소켓 리스트에 담음. 클라이언트 소켓이 접속했음을 콘솔에 출력하고 (클라이언트 소켓, 클라이언트 주소, 소켓 리스트) 반환.</li> <li>3. <b>send_command(식별자, 내용, 클라이언트 소켓):</b> 클라이언트에 송신할 식별자와 내용을 json 형식의 리스트로 변환하여 바이트 형태로 인코딩 후 클라이언트에게 전송, 전송 내용과 시간을 콘솔에 출력.</li> <li>4. <b>null_to_zero(DB로부터 fetchall로 받아온 데이터):</b> 데이터에 null이 있을 경우 반복문을 실행해 null을 0으로 변경.</li> <li>5. <b>check_if_exist(요소, DB테이블, DB테이블의 칼럼):</b> DB에서 받아온 데이터를 반복문을 활용해 대조하여 인수 요소가 DB에 이미 입력되어 있는지 확인.</li> <li>6. <b>get_single_item(아이템 칼럼, DB테이블, 키 칼럼, 키 아이템):</b> DB 테이블 중 키 칼럼이 키 값인 행의 아이템 칼럼 데이터 추출.</li> <li>7. <b>get_whole_data(테이블):</b> 해당 테이블의 모든 데이터 추출.</li> <li>8. <b>get_whole_data_where(테이블, 키 칼럼, 키 아이템):</b> DB 테이블중 키 칼럼이 키 값인 행의 데이터 전부 추출.</li> <li>9. <b>get_database_from_url(url):</b> request를 통해 url의 데이터를 다운로드하고 텍스트로 변환하여 반환</li> <li>10. <b>xml_to_json(xml_string):</b> xml 형태인 문자열을 인수로 받아 json 형태로 변환</li> <li>11. <b>execute_db(sql):</b> 인수로 받아온 sql문을 실행, 커밋 및 결과 반환</li> </ol> <p><b>class MainServer:</b> 매개변수와 기능이 제한적인(이번 프로그램에서만 사용 가능해 보이는) 함수 클래스, education application의 서버 로직</p> <ol style="list-style-type: none"> <li>1. <b>turn_server_on(함수, 서버 소켓, 소켓 리스트):</b> select를 활용한 소켓 통신, 통신의 처리 과정을 콘솔에 출력함. 받은 메시지를 eval을 활용하여 형을 복원한 뒤 클라이언트 소켓과 함께 인수로 받아온 함수의 매개변수로 사용.</li> <li>2. <b>connection_lost(클라이언트 소켓, 소켓 리스트):</b> 클라이언트 소켓 연결 종료를 콘솔 창에 출력, 클라이언트 소켓과의 연결을 닫고 소켓 리스트에서 제거 후 소켓 리스트 반환.</li> <li>3. <b>get_useful_data(100대 명산 미정제 데이터):</b> 100대 명산 API 공공데이터 중 산 이름, 선정 이유, 개요, 상세 데이터를 선별해 반환하는 함수.</li> <li>4. <b>command_processor(메시지, 클라이언트 소켓):</b> 식별자와 내용을 분리하여 식별자에 해당하는 함수에 내용과 클라이언트 소켓을 매개 변수로 넣어줌.</li> <li>5. <b>check_registrable(회원 가입 정보, 클라이언트 소켓):</b> 서버 툴의 check_if_exist 함수를 통해 user_id 중복 여부를 확인, 중복되었을 경우 회원 가입 실패 메시지를</li> </ol>

		<p>클라이언트에 전달, 가입 가능한 경우 클라이언트에서 입력한 회원 가입 정보와 클라이언트 소켓을 매개변수로 <b>self.regist_user</b> 함수 호출.</p> <p>a. <b>regist_user</b>(회원 가입 정보, 클라이언트 소켓): DB에 회원 가입 정보 저장, 클라이언트에게 가입 성공 메시지 송출.</p> <p>6. <b>student_login_process</b>(로그인 정보, 클라이언트 소켓): 학생의 로그인 프로세스. 수신한 아이디와 비밀번호의 일치 여부를 확인해 실패시 각각의 실패 메시지를 클라이언트에게 전달, 아이디와 비밀번호가 함께 일치시 DB로부터 유저 정보와 퀴즈 정보를 불러와 학생 클라이언트에게 로그인 성공 메시지와 함께 전달.</p> <p>7. <b>teacher_login_process</b>(로그인 정보, 클라이언트 소켓): 교사의 로그인 프로세스, id와 비밀번호 일치 여부 확인 후 일치시 로그인 성공 메시지와 함께 로그인한 유저명을 교사 클라이언트에게 전송, 로그인 프로세스는 각각의 클라이언트 클래스에 해당하는 디렉터리에 클라이언트 소켓을 담음</p> <p>8. <b>question_from_student</b>(질문 데이터, 클라이언트 소켓): 질문 데이터를 게시 시간, 학생명, 질문 내용으로 세분화 후 DB에 저장, 학생 클라이언트에 작업 완료를 알람.</p> <p>9. <b>send_whole_qna_data</b>(더미, 클라이언트 소켓): DB로부터 qna의 모든 데이터를 읽어오고 null 값을 'X'로 치환, 가공한 데이터를 요청한 클라이언트에 전송.</p> <p>10. <b>insert_score</b>(답변 데이터, 클라이언트 소켓): <b>score_board</b> DB 테이블에 학생 클라이언트로부터 받아온 문제풀이 데이터 입력</p> <p>11. <b>insert_qna_answer_to_database</b>(답변, 클라이언트 소켓): 인수로 받은 데이터를 답변과 <b>qna_index</b>로 세분화하여 <b>qna</b> 테이블에 데이터 삽입.</p> <p>12. <b>student_score</b>(클라이언트 소켓): DB로부터 유저 수만큼의 [유저명, [[점수, 시간, 지역], [점수, 시간, 지역]...]]의 데이터를 획득 후 교사에게 송신</p> <p>13. <b>add_quiz</b>(문제 데이터, 클라이언트 소켓): 교사 클라이언트가 출제한 문제의 데이터를 받아 DB의 <b>quiz</b> 테이블에 입력</p> <p>14. <b>send_quiz_by_location</b>(지역 데이터, 클라이언트 소켓): 클라이언트로부터 수신하는 데이터가 'quize_영어지역명'이기 때문에 앞 6자를 제외한 7자부터를 새로이 지역 데이터로 선언. 영어로 된 지역명을 한글로 변환하여 DB상에서 해당 지역 퀴즈의 <b>index</b>, 문제내용, 정답여부, 점수를 찾아 클라이언트에게 전송.</p> <p>15. <b>send_login_member_list</b>(클라이언트 소켓): 요청하는 클라이언트의 클래스에 따라 실시간 상담에 필요한 상대방 클래스의 접속한 유저의 <b>index</b>를 전송</p> <p>16. <b>get_user_name</b>(유저 인덱스 목록, 클라이언트 소켓): 유저 인덱스를 통해 DB에서 유저명을 추출하여 클라이언트에게 전송</p> <p>17. <b>load_learning_user</b>(클라이언트 소켓): <b>get_user_index</b> 함수를 호출, 요청을 보낸 클라이언트의 학습정보를 불러와 전송</p> <p>18. <b>get_student_index</b>(클라이언트 소켓): 인수로 받은 클라이언트 소켓의 유저 번호를 반환</p> <p>19. <b>get_past_chat</b>(유저명 목록, 클라이언트 소켓): 실시간 상담중인 두 사람의 유저명을 통해 둘 사이의 대화를 DB로부터 로드</p> <p>20. <b>receive_chat_message</b>(내용, 클라이언트 소켓): 채팅을 송신하고 수신한 유저명을 통해 DB에서 <b>user_index</b>를 추출, '송신인: 내용'의 형태로 DB 형식에 맞게 저장</p> <p>21. <b>get_user_index_by_name</b>(유저명): 인수로 받은 유저명을 통해 유저 인덱스를</p>
--	--	--

		<p>찾아 반환</p> <p>22. <b>send_chat_message</b>(내용, 클라이언트 소켓): 유저명: 내용 의 형태로 현재 상담중인 각 클라이언트에게 전송</p> <p>23. <b>get_client_socket_by_index</b>(유저 인덱스): 인수로 받은 유저 인덱스를 통해 유저 소켓 딕셔너리 상 존재유무를 파악한 뒤 유저 소켓을 반환</p> <p><b>class ChatClient:</b> 실시간 상담 기능 클라이언트, 테스트를 위해 임시로 <b>ui</b>와 소켓설정을 하였음.</p> <ol style="list-style-type: none"> <li>1. <b>set_socket:</b> 서버 <b>ip</b>와 포트 설정 및 입력</li> <li>2. <b>set_gui:</b> 클라이언트 창 크기 설정, 윈도우 출력</li> <li>3. <b>login_process:</b> 로그인</li> <li>4. <b>receive_message</b>(소켓): 송신한 메시지를 명령으로 연결</li> <li>5. <b>chat_client:</b> 채팅 <b>gui</b> 설정, 각 <b>ui</b> 시그널 설정, 유저 목록 요청</li> <li>6. <b>renew_user_list</b>(연결 가능 유저 목록): 서버로부터 받아온 상대 유저(학생&lt;-&gt;교사) 목록을 콤보박스에 추가, 추가된 유저가 없을 시 상담 불가를 알림</li> <li>7. <b>request_past_chat_data:</b> 콤보박스 아이템 변경시 발동, 채팅창 초기화, 현재 상담이 가능한 상태일 경우 지난 상담 데이터를 서버에 요청</li> <li>8. <b>print_past_chat:</b> 지난 상담 내역 출력 및 스크롤 하단 이동</li> <li>9. <b>send_chat:</b> 채팅 가능 상태일 경우 입력창 초기화, 현재 입력창에 적힌 <b>text</b>를 서버로 전송(채팅창에 바로 출력하지 않음)</li> <li>10. <b>receive_chat</b>(채팅 내용): 채팅 내용을 채팅창에 출력</li> </ol>
--	--	---

	박의용	<div data-bbox="443 203 1481 271" data-label="Text"> <pre>def register_server(self):</pre> </div> <div data-bbox="491 309 1096 349" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 1. 회원가입 데이터 서버에 전송 함수</li> </ul> </div> <div data-bbox="536 349 1160 396" data-label="Text"> <pre>def register_success(self)</pre> </div> <div data-bbox="491 414 890 452" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 1.1 회원가입 완료 함수</li> </ul> </div> <div data-bbox="536 452 1114 499" data-label="Text"> <pre>def register_fail(self):</pre> </div> <div data-bbox="491 515 890 555" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 1.2 회원가입 중복 함수</li> </ul> </div> <div data-bbox="443 595 1481 663" data-label="Text"> <pre>def login(self):</pre> </div> <div data-bbox="491 696 983 736" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 2. 로그인 데이터 서버에 전송</li> </ul> </div> <div data-bbox="536 736 1114 784" data-label="Text"> <pre>def login_id_fail(self):</pre> </div> <div data-bbox="536 799 1260 846" data-label="Text"> <pre>def login_password_fail(self):</pre> </div> <div data-bbox="491 862 857 900" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 2.1 로그인 실패 알림</li> </ul> </div> <div data-bbox="536 900 1335 947" data-label="Text"> <pre>def login_success(self, content):</pre> </div> <div data-bbox="491 965 1473 1005" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 2.2 로그인 성공 알림, 서버에서 데이터 받아와서 데이터 배치</li> </ul> </div> <div data-bbox="443 1046 1481 1171" data-label="Text"> <pre>def send_command(self, command, content):</pre> </div> <div data-bbox="491 1207 1436 1249" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 3. ( 서버에 알릴 명령어 / 담아있는 정보 ) 서버에 전달 함수</li> </ul> </div> <div data-bbox="443 1290 1481 1357" data-label="Text"> <pre>def get_message(self):</pre> </div> <div data-bbox="491 1391 983 1431" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 4. 서버에서 메시지 받는 함수</li> </ul> </div> <div data-bbox="443 1509 1481 1635" data-label="Text"> <pre>def command_processor(self, command, content):</pre> </div> <div data-bbox="491 1673 1342 1715" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 5. ( 서버에 받은 메시지 ) 명령어, 정보 받아오는 함수</li> </ul> </div> <div data-bbox="443 1756 1481 1823" data-label="Text"> <pre>def quiz_request(self):</pre> </div> <div data-bbox="491 1856 1227 1897" data-label="List-Group"> <ul style="list-style-type: none"> <li>□ 6. 퀴즈 풀이 버튼 지역 마다 데이터 배치 준비</li> </ul> </div>
--	-----	--

		<pre>def quiz_receive(self, quiz=list, quiz_total=list, area_name=str):</pre> <p><input type="checkbox"/> 7. 지역마다 데이터 배치 하기 위한 데이터 정렬</p> <pre>def quiz_solving(self, rbutton):</pre> <p><input type="checkbox"/> 8. 퀴즈 푸는 함수</p> <pre>def quiz_submit(self):</pre> <p><input type="checkbox"/> 9. 퀴즈 정답 제출 함수</p> <pre>def quiz_to_server(self, useranswer=str):</pre> <p><input type="checkbox"/> 10. 퀴즈 정답 제출한 정보를 서버에 보내는 함수</p> <pre>def quiz_last_check(self, quizindex, answer):</pre> <p><input type="checkbox"/> 11. 이미 푼 퀴즈인지 확인하는 함수</p> <pre>def learning_submit(self):</pre> <p><input type="checkbox"/> 12. 학습 단계 서버에 보내는 함수</p> <pre>def nextstep(self):</pre> <p><input type="checkbox"/> 13. 다음 문제 푸는 함수</p> <pre>def load_learning_success(self, content):</pre> <p><input type="checkbox"/> 14. 지난 학습 데이터 서버에 요청하여 불러오기</p>
--	--	--

		<pre>def connectserver(self):</pre> <p>□ 15. 서버와 연결하는 함수</p>
후기	김명은	<p>백차다. 여기서 나는 무엇을 얻어갔을까 프로젝트를 진행하면서 식별자를 서버에게 요청하면 서버파트가 식별자를 보고 다시 클라이언트에게 보내주는 것, 인코딩해서 서버에게 전달, 디코딩해서 서버에게 정보를 받는것, 로그인, 회원가입, 문제출제, 문제추가, 점수현황, Q&amp;A등 모든기능을 최대한 내가 스스로 해보려고 한것이 그 과정에는 힘들었지만 지금은 성취감이 남아서 나중에 기억에 남을 것 같은 경험이었다. 그리고 프로젝트를 진행하면서 스스로 부족하다고 느끼고있어서 꾸준히 노력해야겠다고 생각하고 도움을 준 팀원들에게 감사하다.</p>
	박규환	<p>TCP/IP에 조금 더 익숙해질 수 있는 좋은 경험이었습니다. 프로젝트를 진행하며 따로 모듈(?) 비슷한 것도 만들어 보았고 함수들을 최대한 독립적으로 구성해보기 위해 self 사용도 최소화 해봤는데 확실히 함수들을 기존보다 범용성 있게 사용할 수 있는 느낌입니다. 끝까지 남아 열심히 해주신 두 분 모두 고생하셨습니다.</p>
	박의용	<p>이번 프로젝트를 진행하면서 TCP/IP 서버와 데이터를 주고받는 것에 친숙 해 질수 있었다. 짧은 데이터 뿐만 아니라, 데이터가 크기가 커진다면 어떤 상황에 직면하는지 알 수 있었다. 또한 각 데이터를 배치하는 과정에서 데이터 리스트, 인덱스 사용에 익숙 해 질 수 있었다. 다만 서버에서 데이터를 주고 받는 과정에서 처음에 너무 많은 데이터를 받아서 유저 클라이언트에서 처리 한게 아닐까 하는 아쉬움이 남았다. 서버와 클라이언트 간 데이터를 주고 받는 과정을 최대한 줄일 수 없을까? 하는 고민을 하였는데, 그 이유는 서버와의 접속이 끊기더라도 기존 로그인 할 시 남아있는 정보로 사용자가 학습을 할 수 있진 않을까 하는 고민에서 나왔지만, 결국 최선의 방법은 아니었음을 깨달았다. 프로젝트 성공/실패 여부를 떠나서 할 수 있는 최대한의 노력을 하였고, 결과보다 과정에서 많은 배움이 있었다고 생각했다.</p>
비고		