

테스트 케이스 ID	테스트 케이스 설명	사실 조건	테스트 결과 데이터	기대 결과	테스트 결과	주요 작업	테스트 결과
POSTING-TC-001	Post 페이지에 포스트된 글이 잘 로드되었는지 확인한다.	관리자 계정이 사전에 생성되어 있어야 한다.	[Post 페이지에 정보드롬 잘] [Title: "첫 번째 포스트", "content": "첫 번째 포스트입니다"], [Title: "두 번째 포스트", "content": "두 번째 포스트입니다"], [Title: "세 번째 포스트", "content": "세 번째 포스트입니다"], [Title: "네 번째 포스트", "content": "네 번째 포스트입니다"], [Title: "다섯 번째 포스트", "content": "다섯 번째 포스트입니다"]	Post 페이지에 접속했을 때, 앞에서 입력한 4가지 포스트 내용이 모두 보여야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 시간에 생성한 권고지 계정으로 로그인 한다. 3. 'Post' 페이지로 접속한다. 4. 'New Post' 버튼을 클릭한다. 5. 테스트 데이터를 입력해서 글을 생성한다. (4번 반복 사항) 6. Post 페이지에서 생성된 글을 확인한다.	테스트 완료 후, admin 페이지에 접속해서 생성한 글들을 삭제하고, 동일한 시나리오로 다시 테스트 해본다.	Passed
POSTING-TC-002	메인 페이지에 최신 포스트된 글 3개와 오늘의 단어 1개가 잘 로드되었는지 확인한다.	관리자 계정이 사전에 생성되어 있어야 한다.	[Post 페이지에 정보드롬 잘] [Title: "첫 번째 포스트", "content": "첫 번째 포스트입니다"], [Title: "두 번째 포스트", "content": "두 번째 포스트입니다"], [Title: "세 번째 포스트", "content": "세 번째 포스트입니다"], [Title: "네 번째 포스트", "content": "네 번째 포스트입니다"], [Title: "다섯 번째 포스트", "content": "다섯 번째 포스트입니다"] [오늘의 단어 정보드롬 잘] [Title: "첫 번째 단어", "content": "첫 번째 단어입니다"]	메인 페이지에 접속했을 때, 3가지 Post 최신글과 1가지 최신 단어 내용이 보여야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 시간에 생성한 권고지 계정으로 로그인 한다. 3. 'Post' 페이지로 접속한다. 4. 'New Post' 버튼을 클릭한다. 5. 테스트 데이터를 입력해서 글을 생성한다. (3번 반복 사항) 6. '오늘의 단어' 페이지로 접속한다. 7. 'New Word' 버튼을 클릭한다. 8. 테스트 데이터를 입력해서 글을 생성한다. (1번 사항) 9. 메인 페이지에서 최신 글과 단어를 확인한다.	테스트 완료 후, admin 페이지에 접속해서 생성한 글들을 삭제하고, 동일한 시나리오로 다시 테스트 해본다.	Passed
PAGE-TC-001	AI 만들무런서 소개 페이지가 잘 렌더링 되었는지 확인한다.	about_me.html이 사전에 생성되어 있어야 한다.	None	About Me' 페이지에 접속했을 때, AI 만들무런서 설명이 보여야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 'About Me' 페이지로 접속한다.	테스트 완료 후, 새로 고침 버튼을 클릭하여 동일한 시나리오로 다시 테스트 해본다.	Passed
POSTING-TC-001	게시물 5개를 기준으로 페이지네이션을 잘 하는지 확인한다.	관리자 계정이 사전에 생성되어 있어야 한다.	[Post 페이지에 정보드롬 잘] [Title: "첫 번째 포스트", "content": "첫 번째 포스트입니다"], [Title: "두 번째 포스트", "content": "두 번째 포스트입니다"], [Title: "세 번째 포스트", "content": "세 번째 포스트입니다"], [Title: "네 번째 포스트", "content": "네 번째 포스트입니다"], [Title: "다섯 번째 포스트", "content": "다섯 번째 포스트입니다"], [Title: "여섯 번째 포스트", "content": "여섯 번째 포스트입니다"], [Title: "일곱 번째 포스트", "content": "일곱 번째 포스트입니다"], [Title: "여덟 번째 포스트", "content": "여덟 번째 포스트입니다"], [Title: "아홉 번째 포스트", "content": "아홉 번째 포스트입니다"], [Title: "열 번째 포스트", "content": "열 번째 포스트입니다"] [오늘의 단어 정보드롬 잘] [Title: "첫 번째 단어", "content": "첫 번째 단어입니다"], [Title: "두 번째 단어", "content": "두 번째 단어입니다"], [Title: "세 번째 단어", "content": "세 번째 단어입니다"], [Title: "네 번째 단어", "content": "네 번째 단어입니다"], [Title: "다섯 번째 단어", "content": "다섯 번째 단어입니다"], [Title: "여섯 번째 단어", "content": "여섯 번째 단어입니다"], [Title: "일곱 번째 단어", "content": "일곱 번째 단어입니다"], [Title: "여덟 번째 단어", "content": "여덟 번째 단어입니다"], [Title: "아홉 번째 단어", "content": "아홉 번째 단어입니다"], [Title: "열 번째 단어", "content": "열 번째 단어입니다"]	Post 페이지와 오늘의 단어 페이지에 접속했을 때, 아래와 같이 보여야 한다. -1페이지 : 게시물 1개 -2페이지 : 게시물 5개	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 시간에 생성한 권고지 계정으로 로그인 한다. 3. '오늘의 단어' 페이지로 접속한다. 4. 'New Word' 버튼을 클릭한다. 5. 테스트 데이터를 입력해서 글을 생성한다. (5번 사항) 6. 메인 페이지에서 최신 글과 단어를 확인한다.	테스트 완료 후, admin 페이지에 접속해서 생성한 글들을 삭제하고, 동일한 시나리오로 다시 테스트 해본다.	Passed
LOGIN-TC-001	로그인 기능이 잘 수행되는지 확인한다.	계정(일반인 or 관리자)이 사전에 생성되어 있어야 한다.	[로그인 페이지] [Title: "사용자 ID", "password": "비밀번호"]	로그인 과정이 성공적으로 이뤄진 후, 메인 페이지에 자동 접속되어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 로그인 페이지로 접속한 후, ID와 비밀번호를 입력한다. 3. 로그인 성공적으로 이뤄졌을 시, 메인 페이지에 자동 리디렉션되고 확인한다.	테스트 완료 후, 로그인한 후에 동일한 시나리오로 다시 테스트 해본다.	Passed
ADMIN-TC-001	관리자 페이지 기능이 잘 수행되는지 확인한다.	관리자 계정이 사전에 생성되어 있어야 한다.	[관리자 페이지 로그인 페이지] [Title: "사용자 ID", "password": "비밀번호"]	로그인 과정이 성공적으로 이뤄진 후, 관리자 페이지에 접속되어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 주소 줄 자리에 "admin"을 붙임으로써 관리자 페이지에 접속한다. 3. 관리자 계정 ID와 비밀번호를 입력한다. 3. 로그인한 성공적으로 이뤄졌을 시, 관리자 페이지가 잘 보이는지 확인한다.	테스트 완료 후, 로그인한 후에 동일한 시나리오로 다시 테스트 해본다.	Passed
LLM-TC-001	Post 페이지에서 LLM 자동 글쓰기 기능이 잘 수행되는지 확인한다.	관리자 계정이 사전에 생성되어 있어야 한다.	None	LLM 자동 글쓰기 기능이 성공적으로 수행된 후, Post 페이지에 업로드 되어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 시간에 생성한 권고지 계정으로 로그인 한다. 3. 'Post' 페이지로 접속한다. 4. 'New Post' 버튼을 클릭한다. 5. 'Generate' 버튼을 클릭하여 글을 생성한다. 6. 글이 성공적으로 작성된 걸 확인한 후, 업로드 버튼을 클릭한다. 7. 'Post' 페이지에 업로드된 글이 잘 보이는지 확인한다.	테스트 완료 후, admin 페이지에 접속해서 생성한 글들을 삭제하고, 동일한 시나리오로 다시 테스트 해본다.	Passed
LLM-TC-002	오늘의 단어 페이지에서 LLM 자동 글쓰기 기능이 잘 수행되는지 확인한다.	None	None	예정한 시간에 LLM 자동 글쓰기 기능이 성공적으로 수행된 후, 오늘의 단어 페이지에 업로드 되어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 'Crontab' 시간 설정을 가장 가까운 시간대로 임시 변경한다. 3. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 4. 동일한 임시 시간대로 10분 정도, 새로 크론탭에 새로운 글이 생성되었는지 확인한다.	테스트 완료 후, Crontab 시간 설정을 가장 가까운 시간대로 변경한 다음에 동일한 시나리오로 다시 테스트 해본다.	Passed
LLM-TC-003	메인 페이지에서 LLM으로 작성한 Trend 기사 요약 기능이 잘 수행되는지 확인한다.	None	None	예정한 시간에 LLM 자동 글쓰기 기능이 성공적으로 수행된 후, 메인 페이지에 Trend 요약 기사가 업로드 되어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 동일한 임시 시간대로 10분 정도, 새로 크론탭에 새로운 글이 생성되었는지 확인한다. 3. 'Post' 페이지로 접속해서 해당 기사를 클릭해서 내용을 확인한다. 4. 해당 기사를 클릭해서 2가지 요약된 내용을 수직으로 분할시킨다. 5. 수직으로 분할시킨 데이터를 2개의 Cosine Similarity를 계산한다.	테스트 완료 후, Crontab 시간 설정을 가장 가까운 시간대로 변경한 다음에 동일한 시나리오로 다시 테스트 해본다.	Passed
LLM-TC-004	사전에 정의한 Post 페이지 글 소스가 잘 정의되었는지 확인한다. (중요한 물론 회피 등도)	None	'Post' 페이지 글 소스' 자동 대체할 서명	각 글 소스는 서로 겹치는 내용이 없어야 한다.	1. 'python manage.py runserver' 명령어를 사용해서 메인 페이지를 연다. 2. 동일한 임시 시간대로 10분 정도, 새로 크론탭에 새로운 글이 생성되었는지 확인한다. 3. 'Post' 페이지로 접속해서 해당 기사를 클릭해서 내용을 확인한다. 4. Cosine Similarity가 85%를 넘는 경우에는 동일 글문이라고 간주한다.	테스트 완료 후, 서로 유사한 글문들을 삭제한 후에 동일한 시나리오로 다시 테스트 해본다.	Passed