

하루하루 챗봇 프로젝트 화면 개요

1. 전체 폴더/파일 구조

frontend/	
—— public/	
—— logo,fruit.png...	# 아이콘, 이미지 저장. 직접 URL로 접근 가능
—— src/	
—— assets/	# 이미지, 정적 파일
—— pages/	# 라우팅되는 주요 페이지
—— WelcomePage.tsx	# 메인(주제선택) 페이지
—— WelcomePage.css	# WelcomePage 전용 스타일
—— ChatBotPage.tsx	# 챗봇 메인 페이지
—— ChatBotPage.css	# ChatBotPage 전용 스타일
—— TopicSelectPage.tsx	# 보조(주제선택) 페이지
—— TopicSelectPage.css	# TopicSelectPage 전용 스타일
—— Common.css	# 공통 스타일
—— App.tsx	# 라우터 설정 포함
—— main.tsx	# React 진입점
—— index.html	# 웹페이지 뼈대
—— package.json	# Node 기반 환경 설정 파일
—— tsconfig.json	# TypeScript 컴파일 옵션을 정의한 설정 파일
—— vite.config.ts	# 타입 환경파일



- package.json: 설치된 라이브러리, 실행 스크립트, 프로젝트 정보 등을 정의
- tsconfig.json: 어떤 파일을 포함/제외할지, 어떤 수준의 엄격한 타입 검사를 할지 설정

⊙ `assets/` 폴더는 이미지 등 정적 파일 관리를 위해 구성되었으나, 현재는 이미지 접근을 `public/` 경로를 통한 URL 사용 방식으로 처리하고 있어 `assets/` 폴더는 사용되지 않음

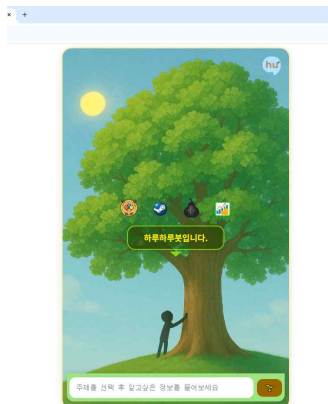
2. WelcomePage (첫 페이지, 주제 선택 화면) 상세 정의

2-1. 화면 정보

- 화면 이름: WelcomePage (메인/주제선택)
- 화면 경로: /
- 목적: 아이콘을 통해 확실한 주제를 선택하고 첫 질문을 입력, 대화를 시작하는 화면

2-2. UI 요소 및 구조

요소명	클래스/ID	위치	설명/역할	상태/변형
배경 이미지	.welcome-background	전체	푸른 하늘, 높은 언덕에 큰 나무 한 그루와 사람 한 명	고정
상단 로고	.welcome-logo-inside	우측 상단	앱 로고(hu)	고정
주제 아이콘	.fruit-icon	나무 위	주제 선택 (축제, 스팀, 리니지2M, 주식)	클릭 활성화, 0 or 1개 선택
말풍선 (코멘트)	.comment-bubble	나무 중앙	챗봇 주제별 안내	주제별 텍스트/색상 변경
입력창	.chat-input	하단	사용자 질문 입력	입력 없으면 전송 비활성
전송버튼	.chat-btn	입력창 오른쪽	메시지 전송, ChatBotPage로 이동	이벤트 없으면 씨앗 유지
안내문구	.guide-text	입력창 내부	"주제를 선택 후 알고싶은 ..."	고정



- 주제 아이콘 클릭:
해당 아이콘 활성화(테두리/확대), 말풍선 텍스트 변경
- 입력창 메시지 전송: /chat?msg=xxx&topic=yyy로 이동, 입력값 없으면 버튼 비활성
- 주제 미선택 시: topic=common으로 이동, 말풍선은 "하루하루봇입니다."
- 엔터 입력: 줄바꿈 없이 전송 (줄바꿈: shift + enter 사용으로 줄바꿈 가능)
- 접근성/반응형: 모바일, PC 모두 대응, Tab 이동, aria-label 적용

2-3. 주요 CSS/스타일

- 전체 컨테이너: `.chatbot-container { max-width: 56vh; background: #e9e9e9; ... }`
- 배경: `.welcome-background { background-image: url('/nature_scene.png'); ... }`
- 주제 아이콘: `.fruit-icon.selected`
`{ box-shadow: 0 0 10px 3px #82d300; transform: scale(1.15); }`
- 말풍선: `.comment-bubble`
`{ background: rgba(34,66,0,0.582); color: #ffe400; border: 2.5px solid #82d300; }`
- 입력창/버튼: `.chat-input`, `.chat-btn` (둥근 모서리, 흙색/연두색 등)

2-4. 주요 코드/데이터 흐름

- `selectedTopic`: 현재 선택 주제 상태
- `userInput`: 입력창 텍스트 상태
- `handleTopicClick`: 주제 선택 핸들러
- `handleSubmit`: 메시지 전송 및 페이지 이동

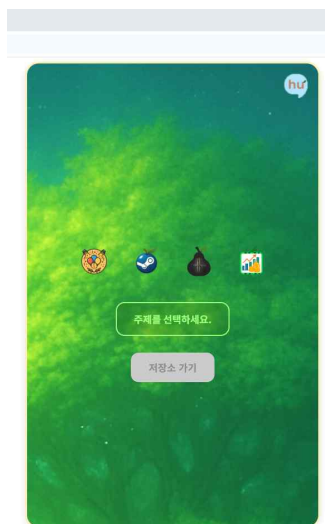
3. TopicSelectPage (보조 주제 선택 화면) 화면 정의

3-1. 화면 정보

- 화면 이름: TopicSelectPage (보조/주제 선택)
- 화면 경로: /topic-select
- 목적: 주제를 선택한 후 챗봇 대화에 자유롭게 진입할 수 있도록 추가적으로 설계된 화면

3-2. UI 요소 및 구조

요소명	클래스/ID	위치	설명/역할	상태/변형
배경 이미지	.topic-select-background	전체	밤의 몽환적인 분위기 일러스트 적용	고정
상단 로고	.topic-select-logo	우측 상단	앱 로고(hu)	고정
주제 아이콘	.topic-icon	중앙 상단	열매처럼 배치된 주제 선택 (축제, 스팀, 리니지2M, 주식)	클릭 활성화, 1개 선택
말풍선 (코멘트)	.topic-comment-bubble	주제 아래	챗봇 주제별 안내	주제별 텍스트/색상 변경
저장소 이동 버튼	.go-storage-btn	화면 하단	주제 선택 완료 후 활성화, 클릭 시 ChatBotPage로 이동	선택 전에는 비활성화 상태



- 주제 아이콘 클릭:
해당 아이콘 활성화(테두리/확대), 말풍선 텍스트 변경
- 말풍선 텍스트: 선택된 주제에 따라 "축제봇입니다.", 등 동적 출력
- 비선택 시 버튼 비활성화: disabled={!selectedTopic}
- 접근성/반응형: 모바일, PC 모두 대응, Tab 이동, aria-label 적용
- 말풍선 및 버튼에 충분한 색상 대비 확보
- 로고 클릭 시 홈으로 이동하는 기능 포함

3-3. 주요 CSS/스타일

- 전체 컨테이너: `.chatbot-container { max-width: 56vh; background: #e9e9e9; ... }`
- 배경: `.topic-select-background { background-image: url('/nature_scene_part.png'); }`
- 주제 아이콘 선택 효과: `.topic-icon.selected { box-shadow: 0 0 10px 3px #ffe400; transform: scale(1.13); }`
- 말풍선: `.topic-comment-bubble { background: rgba(34, 66, 0, 0.18); color: #9cff7a; border: 2px solid #9cff7a; }`
- 저장소 버튼: `.go-storage-btn` (노란색 #ffe400 배경, 선택 전에는 회색 비활성 처리)

3-4. 주요 코드/데이터 흐름

- `selectedTopic`: 현재 선택된 주제 상태 (`useState`)
- `handleTopicClick`: 주제 클릭 이벤트 → 상태 변경
- `handleGoToChatBot`: 버튼 클릭 시 `/chat?topic=선택값`으로 이동

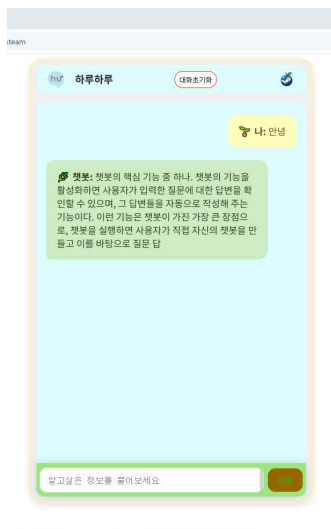
4. ChatBotPage (챗봇페이지) 화면정의

4-1. 화면 정보

- 화면 이름: ChatBotPage (챗봇 메인)
- 화면 경로: /chat
- 목적: 선택 주제에 따라 사용자와 챗봇이 대화하는 화면

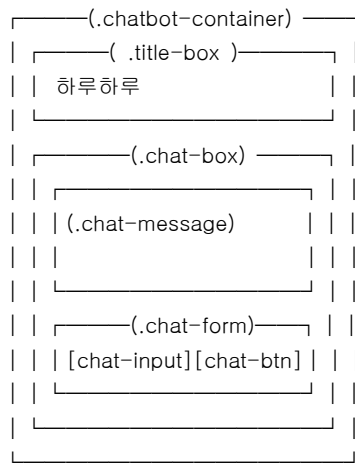
4-2. UI 요소 및 구조

요소명	클래스/ID	위치	설명/역할	상태/변형
상단 바	.title-box	상단	앱명, 대화초기화 버튼, 주제 아이콘	대화초기화 클릭 시 전체 메시지 삭제
주제 아이콘	.title-topic-icon	상단 우측	선택 주제 (과일 아이콘) 표시	topic 값에 따라 이미지 변경
채팅 메시지	.chat-box	중앙	대화 내역(챗봇/사용자)	메시지 추가 시 자동 스크롤 하단
챗봇 말풍선	.chat-message.bot	좌측	챗봇 응답(연두색 말풍선, 타이핑 효과)	타이핑 효과 후 메시지로 치환
사용자 말풍선	.chat-message.user	우측	사용자 메시지(연한 레몬색 말풍선)	입력 후 바로 추가
입력창	.chat-input	하단	메시지 입력	입력 없으면 전송 버튼 비활성
전송 버튼	.chat-btn	입력창 오른쪽	메시지 전송	클릭/엔터 시 sendMessage 실행



- 메시지 전송:
 - 입력 후 버튼 클릭/엔터 → 사용자 메시지 추가
 - 챗봇 응답(타이핑 효과)
 - 챗봇 응답은 axios로 서버 요청, 오류 시 안내 메시지
- 대화 초기화:
 - 상단 버튼 클릭 시 localStorage 해당 키 전체 삭제
- 로컬스토리지:
 - 주제별로 저장/불러오기 (chatbot_messages_{topic})
- 초기 메시지 처리:
 - WelcomePage에서 전달된 msg 쿼리스트링이 있으면 자동 첫 메시지 전송
- 자동 스크롤
 - 메시지 추가/타이핑 시 채팅박스 하단으로 자동 스크롤
 - 입력값 없음/로딩 중: 전송 불가(버튼 비활성)
 - 서버 오류: "오류가 발생했습니다." 메시지 표시

4-3. 주요 CSS/스타일



- 전체 컨테이너: `.chatbot-container`
{ max-width: 56vh; background: #e9e9e9; ... }
- 상단: `.title-box`
{ background: #d8f6fa; border-radius: 18px; ... }
- 챗봇 말풍선: `.chat-message.bot`
{ background: #c7e5be; color: #2b3d1d; ... }
- 사용자 말풍선: `.chat-message.user`
{ background: #fff8b8; color: #5a4b1c; ... }
- 입력창/버튼: `.chat-input`, `.chat-btn`
(둥근 모서리, 흡색/연두색 등)

4-4. 주요 코드/데이터 흐름

- `messages: { type: 'user' | 'bot'; text: string }[]`
- `typingMessage`: 챗봇 타이핑 효과용 임시 메시지
- `userInput`: 입력창 상태
- `isLoading`: 챗봇 응답 대기 상태
- `sendMessage`: 메시지 전송 및 응답 처리
- `handleClearChat`: 대화초기화
- `showTypingEffect`: 챗봇 타이핑 애니메이션

4-5 접근성/반응형

- 모바일/PC 모두 대응: min-width, max-width, padding, flex 레이아웃
- 키보드 접근성: 입력창, 버튼 등 Tab 이동, 엔터 전송 지원
- 색상 대비: 챗봇(연두), 사용자(레몬), 배경(하늘색) 등 충분한 대비 확보
- 스크린리더: 버튼, 입력창에 `aria-label` 등 추가 권장

5. 색상 팔레트

역할	색상코드	설명
하늘 배경	 #B5D6F6	메인페이지 배경
챗봇 말풍선	 #D8F5A2,  #c7e5be	나뭇잎, 연두색
사용자 말풍선	 #FFF9C4,  #fff8b8	씨앗, 연한 레몬색
입력창/버튼	 #BCA177,  #946500	흙, 자연의 연결
주제 아이콘	각기 다름	주제별 과일/아이콘 색상

6. 기타 기능

- 비로그인 접근 및 사용 가능
- 대화 내역: localStorage 저장(주제별)
- 모든 UI 요소 접근성 준수(aria-label, 키보드 이동 등)
- 코드 품질: React Hooks(useState, useEffect, useRef) 적극 활용, 컴포넌트 분리