



AI

Emotion

Diary

Chatbot

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26

PsyChat - Psychological counseling chat

- 하루 감정 대화 기록
- AI 감정 분석 및 긍·부정판단
- 키워드 추출 및 요약 제공
- 월별 감정 팔레트 생성
- 감정 변화 통계 그래프 제공

인공지능학과 202404178 강우현

인공지능학과 202404203 김예원

인공지능학과 202404273 이유림



인공지능앱개발 팀프로젝트



AI

Emotion

Diary

Chatbot

앱 소개 ; PsyChat

폴더 설명 ; components 폴더

코드 설명 ; navigation 폴더

코드 설명 ; screens 폴더

코드 설명 ; utils 폴더

코드 설명 ; App.js

코드 설명 ; firebase.js



AI

Emotion

Diary

Chatbot

PsyChat

설명

PsyChat_ AI가 감정을 분석하고 기록해주는 스마트 다이어리
하루의 텍스트를 입력하면 감정 상태·키워드·변화를 자동으로
파악하고 캘린더와 그래프로 시각화함





AI

Emotion

Diary

Chatbot

components 폴더

설명

ChatBubble.js → 채팅 말풍선 UI

MoodPalette.js → 감정 팔레트(월별 감정 색상 표)

PieChartComponent.js → 원형 그래프(부정/긍정 비율)



AI

Emotion

Diary

Chatbot

navigation 폴더

```
<Stack.Screen name="Login" component={LoginScreen} />
<Stack.Screen name="MainTabs" component={MainTabs} />
</Stack.Navigator>
```

로그인 → 메인 구조

설명

_앱의 화면 이동을 관리함

_로그인 화면과 메인 화면

흐름을 분리하고 구조화함

홈, 캘린더, 챗봇, 그래프, 기록

탭 5개

```
<Tab.Screen name="홈" component={HomeScreen} />
<Tab.Screen name="캘린더" component={CalendarScreen} />
<Tab.Screen name="챗봇" component={ChatScreen} />
<Tab.Screen name="그래프" component={GraphScreen} />
<Tab.Screen name="기록" component={HistoryScreen} />
</Tab.Navigator>
```

```
tabBarActiveTintColor: "#4F46E5",
tabBarInactiveTintColor: "#9CA3AF",
tabBarStyle: {
  height: 70,
  paddingBottom: 10,
  paddingTop: 10,
},
tabBarIcon: ({ color, size, focused }) => {
  let icon;
```

아이콘 상태 변화



AI

Emotion


Diary

Chatbot

screens 폴더 - HomeScreen.js



특징

 앱 첫 화면(메인)

_앱 전체 기록에서 긍정, 부정 감정 비율을 계산

_파이차트로 시각화하여 하루 요약 제공

_다른 화면(그래프·히스토리·캘린더·채팅)으로

이동하는 허브 역할

```
const posScore = entries.reduce(
  (acc, e) =>
    acc +
    (e.sentiment?.label === "positive" ||
     e.sentiment?.label === "very_positive"
      ? 1
      : 0),
  0
);

const negScore = entries.reduce(
  (acc, e) =>
    acc +
    (e.sentiment?.label === "negative" ||
     e.sentiment?.label === "very_negative"
      ? 1
      : 0),
  0
);

const neuScore = entries.reduce(
  (acc, e) => acc + (e.sentiment?.label === "neutral" ? 1 : 0),
  0
);

const total = entries.length;

setPositiveRatio(posScore / total);
setNegativeRatio(negScore / total);
setNeutralRatio(neuScore / total);
```



AI

Emotion


Diary

Chatbot

screens 폴더 - ChatScreen.js



특징

 일기 입력 + 감정 분석이 이루어지는 핵심 화면

_사용자가 텍스트를 작성

_AI 모델(감정 분석, 키워드 분석) 실행

결과 저장

_오늘 날짜 감정 팔레트에도 반영됨

```
const { keywords, counts, weighted } = extractKeywordsWithWeights(text);
const sentiment = estimateSentimentFromWeighted(weighted);
```

```
// ★ 오늘의 감정색 저장 기능 추가
const today = formatDateISO(selectedDate);
await saveMoodColor(today, sentiment.label);
```

```
const entry = {
  id: Date.now().toString(),
  date: entryDate.toISOString(),
  text: text.trim(),
  keywords,
  counts,
  weighted,
  sentiment,
  botReply,
};
```




AI

Emotion


Diary

Chatbot

screens 폴더 - ChatScreen.js



특징

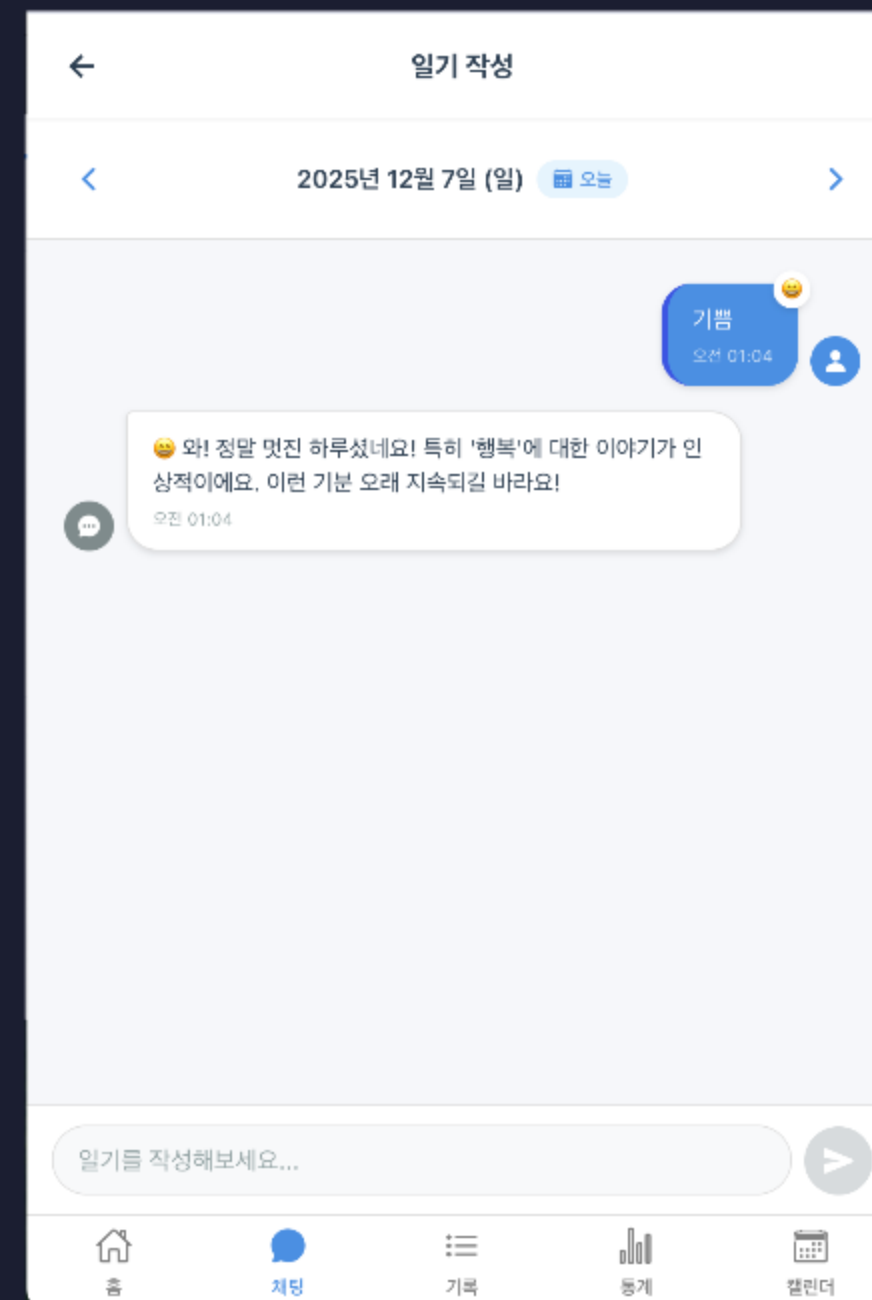
 일기 입력 + 감정 분석이 이루어지는 핵심 화면

_사용자가 텍스트를 작성

_AI 모델(감정 분석, 키워드 분석) 실행

결과 저장

_오늘 날짜 감정 팔레트에도 반영됨





AI

Emotion

Diary

Chatbot

screens 폴더 - CalendarScreen.js



특징

📌 저장된 감정 데이터를 불러와 날짜별 색으로 표시하는 감정 캘린더

시하는 감정 캘린더

_감정에 따라 긍정은 파랑색 부정은 빨강색

_보통은 보라색으로 한 달 분위기를 시각화

```
5 function emotionToColor(sentiment) {
6   if (sentiment === "positive") return "#4C8CFF"; // 파랑
7   if (sentiment === "negative") return "#FF5C5C"; // 빨강
8   if (sentiment === "neutral") return "#B26BFF"; // 보라
9   return "#2a2430"; // 기본 회색
10 }
11
```

```
53  {/* 달력 Grid */}
54  <View style={styles.grid}>
55    {daysArray.map((d, idx) => {
56      if (d === null)
57        return <View key={idx} style={styles.dayEmpty}></View>;
58
59      const dateKey = `${year}-${String(month).padStart(2, "0")}-${String(
60        d
61      ).padStart(2, "0")}`;
62      const mood = moodColors[dateKey];
63      const bg = emotionToColor(mood);
64
65      const isToday = dateKey === today;
66
67      return (
68        <View
69          key={idx}
70          style={[
71            styles.dayBox,
72            {
73              backgroundColor: "#ffffff",
74              borderColor: bg,
75              borderWidth: mood ? 2 : 1,
76            },
77            isToday && styles.todayBorder,
78          ]>
79          <Text style={styles.dayNumber, { color: bg }}>{d}</Text>
80        </View>
81      );
82    })}
83  </View>
84
```



AI

Emotion

Diary

Chatbot

screens 폴더 - CalendarScreen.js



특징

 저장된 감정 데이터를 불러와 날짜별 색으로 표시하는 감정 캘린더

시하는 감정 캘린더

_감정에 따라 긍정은 파랑색 부정은 빨강색

_보통은 보라색으로 한 달 분위기를 시각화





AI

Emotion


Diary

Chatbot

screens 폴더 - GraphScreen.js



특징

 긍정/부정 비율 그래프로 보여주는 화면

_최근 14일 동안 기록한 일기 텍스트를 분석

_자주 쓴 중요 단어 변화 추세를 그래프로 시각화

_단어가 긍정인지 부정인지에 따라 그래프 선 색

상 자동 변경

```
const [selectedPeriod, setSelectedPeriod] = useState(14); // 14일, 30일, 90일
const [sentimentData, setSentimentData] = useState(null);
const [wordData, setWordData] = useState(null);
const [stats, setStats] = useState(null);

useEffect(() => {
  loadData();
}, [selectedPeriod]);

useEffect(() => {
  const unsubscribe = navigation.addListener("focus", () => {
    loadData();
  });
  return unsubscribe;
}, [navigation, selectedPeriod]);

const loadData = async () => {
  try {
    setIsLoading(true);

    const [sentimentResult, wordResult, statsResult] = await Promise.all([
      computeSentimentTimeSeries(selectedPeriod),
      computeWordTimeSeries(selectedPeriod),
      getStatsByDateRange(selectedPeriod),
    ]);

    setSentimentData(sentimentResult);
    setWordData(wordResult);
    setStats(statsResult);
  } catch (error) {
    console.error("데이터 로딩 실패:", error);
  } finally {
    setIsLoading(false);
  }
}
```



AI

Emotion

Diary

Chatbot

screens 폴더 - GraphScreen.js



특징

📌 긍정/부정 비율 그래프로 보여주는 화면

_최근 14일 동안 기록한 일기 텍스트를 분석

_자주 쓴 중요 단어 변화 추세를 그래프로 시각화

_단어가 긍정인지 부정인지에 따라 그래프 선 색

상 자동 변경

통계

감정 변화 추이



자주 사용한 키워드

- | | |
|------|----|
| 1 슬픔 | 2회 |
| 2 분노 | 1회 |
| 3 짜증 | 1회 |
| 4 행복 | 1회 |



통계

2주

1개월

3개월



4

전체 일기



25%

긍정



75%

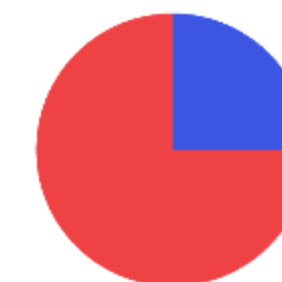
부정



-0.50

평균 점수

감정 분포



1 긍정

3 부정





AI

Emotion

Diary

Chatbot

screens 폴더 - HistoryScreen.js



특징

📌 과거 기록을 날짜별로 보는 화면

_저장된 모든 일기를 불러와서 최신 날짜가 위로 -
오게 정렬

_기록에 대해 시간, 내용, 키워드, 감정을 보여줌

_맨 아래에서 전체 삭제도 가능

```
10 const all = await getEntries();
11 const map = {};
12 all.forEach(e => {
13   const d = e.date.slice(0,10);
14   if (!map[d]) map[d] = [];
15   map[d].push(e);
16 });

const secs = Object.keys(map)
  .sort((a, b) => b.localeCompare(a))
  .map((d) => ({
    title: d,
    data: map[d].sort((a, b) => new Date(b.date) - new Date(a.date)),
  }));

setSections(secs);

<SectionList
  sections={sections}
  keyExtractor={({item}) => item.id}
  renderSectionHeader={({{ section }}) => (
    <View style={styles.sectionHeader}>
      <Text style={styles.sectionTitle}>{section.title}</Text>
    </View>
  )}
  renderItem={({{ item }}) => (
    <View style={styles.item}>
      <Text style={styles.time}>
        {new Date(item.date).toLocaleTimeString()}
      </Text>
      <Text style={styles.text}>{item.text}</Text>
      <Text style={styles.meta}>
        키워드: {item.keywords.join(", ")} || "없음" | 감정:{" "}
        {item.sentiment.label}
      </Text>
    </View>
  )}
  ListEmptyComponent={<Text style={styles.empty}>기록이 없습니다.</Text>}
/>
```



AI

Emotion

Diary

Chatbot

screens 폴더 - HistoryScreen.js



특징



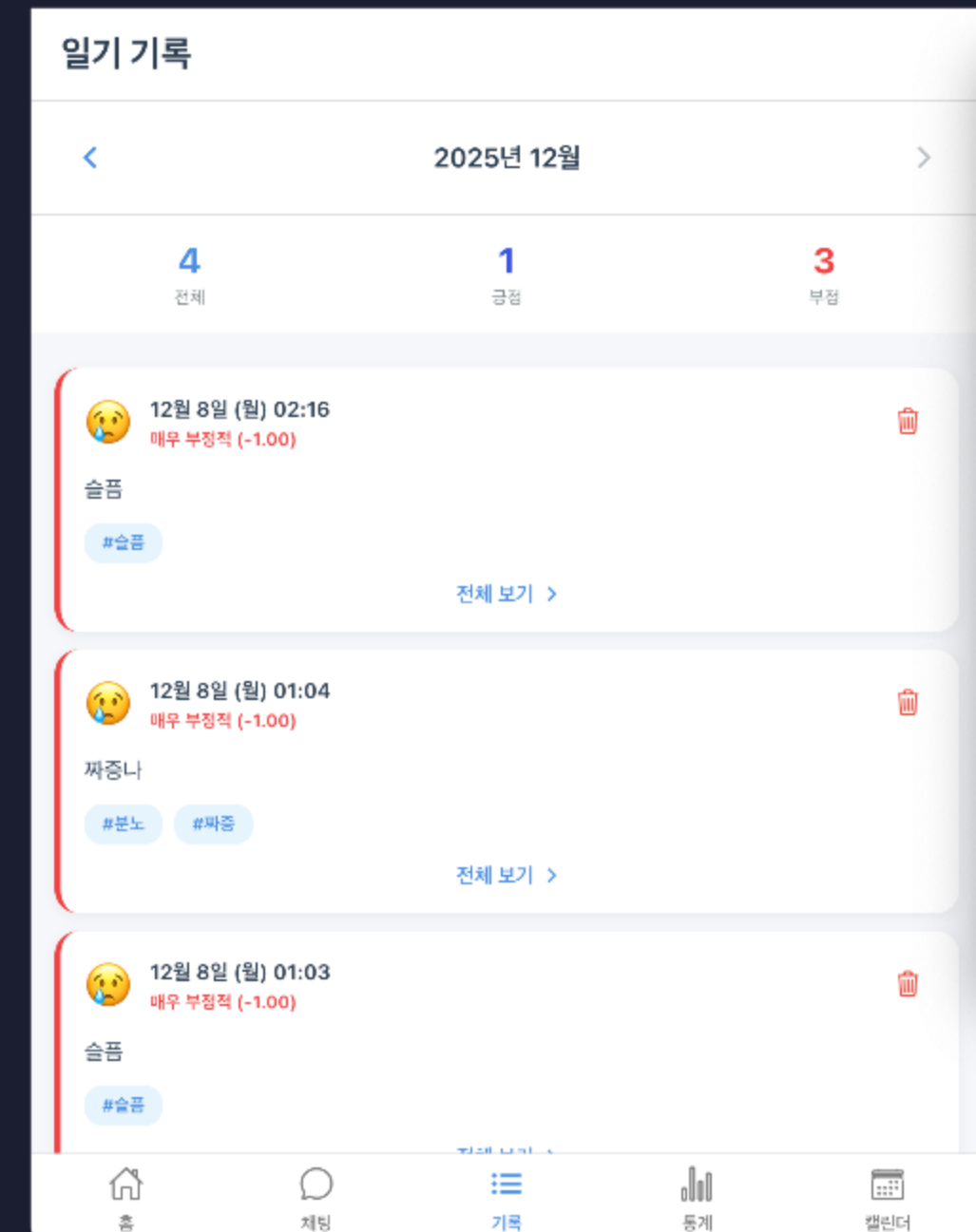
과거 기록을 날짜별로 보는 화면

_저장된 모든 일기를 불러와서 최신 날짜가 위로 -

오게 정렬

_기록에 대해 시간, 내용, 키워드, 감정을 보여줌

_맨 아래에서 전체 삭제도 가능





AI

Emotion

Diary

Chatbot

screens 폴더 - LoginScreen.js



특징



로그인 화면

_로그인

_계정 없을 경우 회원가입

```
try {
  await signInWithEmailAndPassword(auth, email, password);
  // App.js의 onAuthStateChanged가 자동으로 감지하여 MainTabs로 이동
} catch (error) {
  console.error("로그인 에러:", error);

  let errorMessage = "로그인에 실패했습니다.";

  switch (error.code) {
    case "auth/invalid-email":
      errorMessage = "잘못된 이메일 형식입니다.";
      break;
    case "auth/user-disabled":
      errorMessage = "비활성화된 계정입니다.";
      break;
    case "auth/user-not-found":
      errorMessage = "존재하지 않는 계정입니다.";
      break;
    case "auth/wrong-password":
      errorMessage = "잘못된 비밀번호입니다.";
      break;
    case "auth/invalid-credential":
      errorMessage = "이메일 또는 비밀번호가 올바르지 않습니다.";
      break;
    case "auth/network-request-failed":
      errorMessage = "네트워크 연결을 확인해주세요.";
      break;
    default:
      errorMessage = error.message;
  }

  Alert.alert("로그인 실패", errorMessage);
} finally {
  setIsLoading(false);
}
```




AI

Emotion

Diary

Chatbot

screens 폴더 - LoginScreen.js



특징

 로그인 화면

_로그인

_계정 없을 경우 회원가입

로그인

PsyChat에 오신 것을 환영합니다

로그인

[계정이 없으신가요? 회원가입](#)



AI

Emotion

Diary

Chatbot

utils - 도구 설명서처럼 로직 모음 ; wordDictionary.js

_사용자가 입력한 문장에서 감정 관련 단어를 인식하기 위한 사전

_각 감정(행복, 우울, 불안 등)에 대해

→ 자주 쓰는 표현 / 신조어 / 변형된 말들을 synonyms로 묶어서 관리

_weight로 감정의 강도(중요도) 를, sentiment로

→ 긍정(positive) / 부정(negative) 방향을 구분

_이 사전을 기반으로 키워드 추출 + 감정 점수 계산 → 오늘의 기분 색 / 챗봇 응답에 활용



AI

Emotion

Diary

Chatbot

App.js

설명

네비게이션 컨테이너 한 번만 설정

MainStack 안에 Home, Chat, Graph, History,

Calendar 등 모든 화면이 연결됨

```
1 import React from 'react';
2 import { NavigationContainer } from '@react-navigation/native';
3 import MainStack from '../navigation/MainStack';
4
5 export default function App() {
6   return (
7     <NavigationContainer>
8       <MainStack />
9     </NavigationContainer>
10   );
11 }
```



AI

Emotion

Diary

Chatbot

firebase.js

설명

네비게이션 컨테이너 한 번만 설정

MainStack 안에 Home, Chat, Graph, History,

Calendar 등 모든 화면이 연결됨

```
const auth = firebase.auth();

export const login = async ({ email, password }) => {
  const { user } = await auth.signInWithEmailAndPassword(email, password);
  return user;
};

export const signup = async ({ email, password }) => {
  const { user } = await auth.createUserWithEmailAndPassword(email, password);
  return user;
};

export { auth };
```



AI

Emotion

Diary

Chatbot

감사합니다

https://github.com/kangwoohyun999/AI_App_Project