

연애 매칭 프로그램 (6조)

캡스톤 최종 발표

CHAPTER

01 팀원 소개



02 웹 소개(간단히)

03 변경 및 추가된 점

04 완성도



팀원 소개



전진하

- 전반적인 프론트 제작/수정
- 일부 코드 리액트 추가
- 아이콘 작업 / 피피티 제작



김시은

- 일부 프론트 제작
- 스케치업 작업



박예송

- 전반적인 모델링 작업
- 일부 프론트 제작



최혜원

- 커뮤니티 프론트 제작
- 스케치업 작업

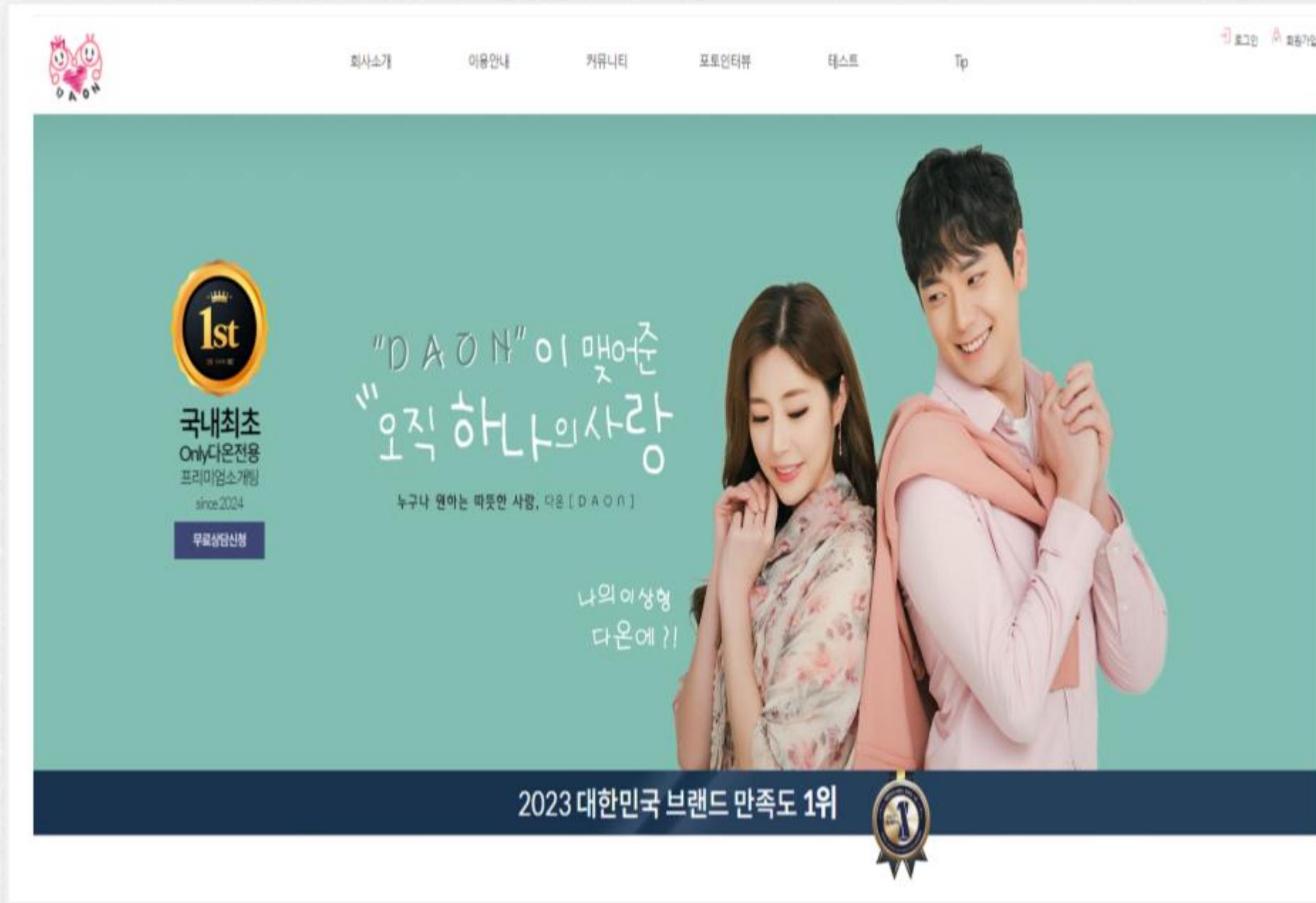
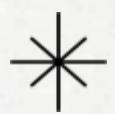


진예희

- 전반적인 서버 제작/수정
- 테스트 프론트 제작
- 전체적인 수합



연애 매칭 사이트 [DAON]



캡스톤 디자인은 웹 / 서버 개발
웹의 종류는 연애 매칭사이트
아이콘 / 로고 / 광고 직접 제작

포토샵 / HTML + CSS +
Javascript / jsp / MySQL /
스케치업 / VRay / 리액트 조금

다양한 상세페이지



5개의 카테고리 / 25개의 상세페이지

연애 매칭이라는 주제에 맞춰 회사소개 / 이용안내 / 커뮤니티 / 포토인터뷰 / 테스트 / Tip이라는 5개의 카테고리로 웹 내용을 세분화 하였습니다.

해당 카테고리는 주제에 맞춰 각각의 상세페이지를 가지고 있습니다. 왼쪽의 사진의 회사소개 페이지와 다온 뉴스라는 상세페이지의 사진입니다.

* 변경 및 추가된 점 *

기존 발표에서 변경 및 추가된 부분이 총 3가지 있습니다.

STEP 1 블라인드 인터뷰 페이지 전반적인 수정

회원 프로필을 제작할 때 리액트를 사용해보았습니다.

STEP 3 회원 간의 DM 기능

매칭된 회원들 간에 소통할 수 있는 기능을 추가하였습니다.

STEP 2 커뮤니티 서버 비밀글 기능 / 이메일 인증

커뮤니티 비밀글 설정과 댓글 기능을 추가하였습니다.
회원가입 시 이메일로 인증할 수 있도록 제작하였습니다.

블라인드 인터뷰 페이지 *

블라인드 명함 인터뷰
다온에는 어떤 회원님들이 있을까요?

Category: 열기 *

김 도 건



나이 만 26세
직업 배우
위치 제주특별자치도
자신을 한 줄로 소개
자유분방한 사람

김 도 아



나이 만 24세
직업 사장가
위치 전라남도
자신을 한 줄로 소개
다행다감한 사람

정 다 인



나이 만 21세
직업 기술직
위치 경기도
자신을 한 줄로 소개
세심하고 열심인 사람

이 윤 지



나이 만 29세
직업 회원
위치 경기도
자신을 한 줄로 소개
TTS로 듣는 즐거운 소리

박 승 민



나이 만 36세
직업 기술 전문직
위치 경상남도
자신을 한 줄로 소개
TTS로 듣는 즐거운 소리

최 해 민



나이 만 30세
직업 회사원
위치 서울특별시
자신을 한 줄로 소개
TTS로 듣는 즐거운 소리

블라인드 명함 인터뷰
다온에는 어떤 회원님들이 있을까요?

다온 블라인드 명함 인터뷰

Category: 열기 *

김 | 여성 | 26세**



나를 한줄로 표현하면
친숙한 대화와 따뜻한 분위기를 만드는
것을 좋아해요.

이런 연인을 찾아요
치유의 힘을 믿으며, 서로의 이야기를
나누는 따뜻한 관계를 원해요.

조 | 여성 | 36세**



나를 한줄로 표현하면
일상을 재미있게 만들어주는 작은 아이디어
가 넘치는 장난꾸러기입니다.

이런 연인을 찾아요
진지한 속에서도 유머를 끌어올리는 사람
을 기대합니다.

한 | 남성 | 30세**



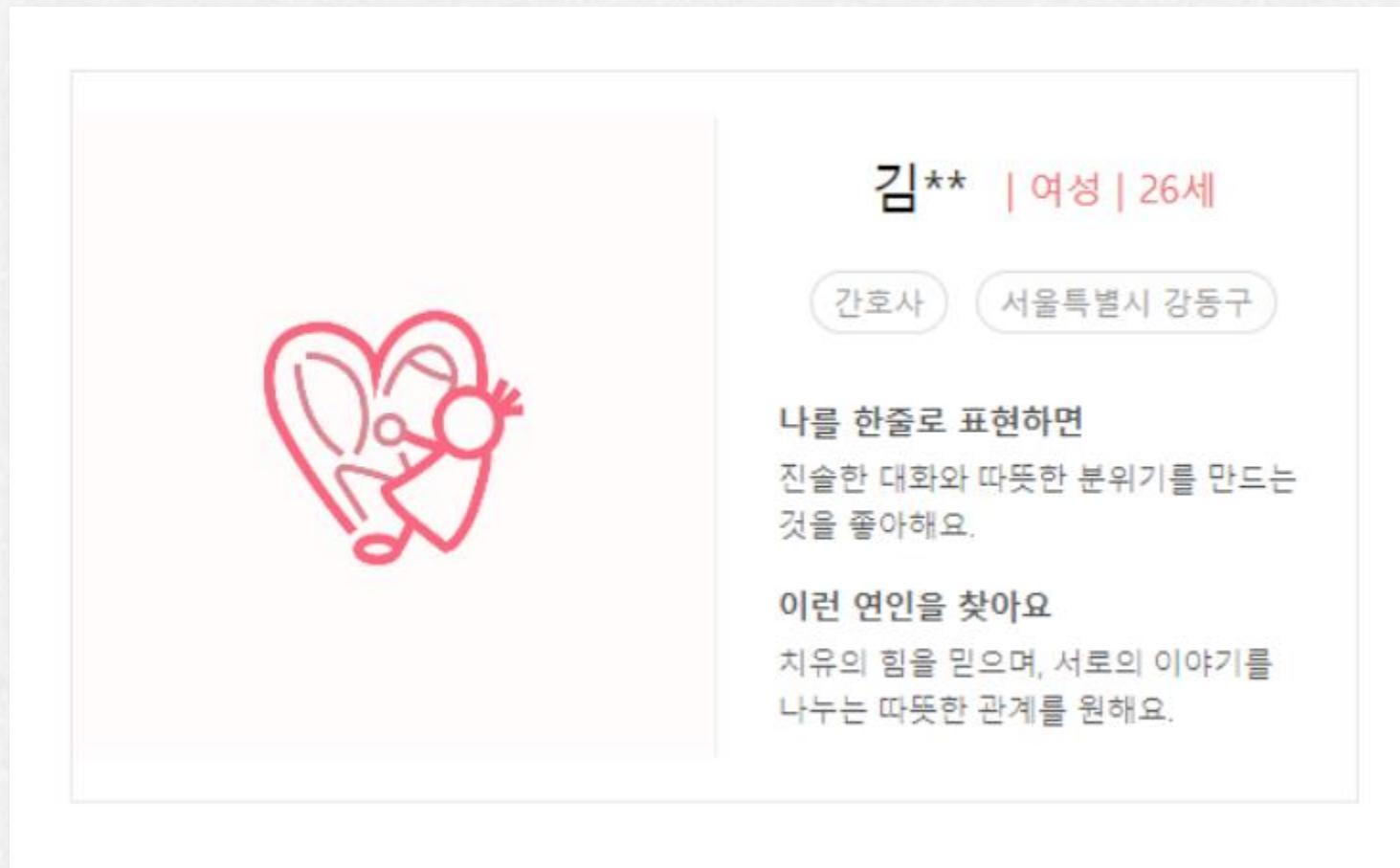
나를 한줄로 표현하면
심리학자

유 | 여성 | 30세**



나를 한줄로 표현하면
제작디자이너

회원 프로필 작성 *

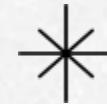


부분적인 리액트 사용

리액트를 사용하는 방식에는 2가지가 있습니다.

- 기존 HTML에 부분적으로 리액트를 사용하는 방식
 - 리액트를 중심으로 전체 웹 개발을 진행하는 방식
- 저희 조는 기존 HTML에 부분적으로 리액트를 적용해 보았습니다.

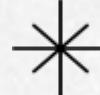
리액트 코드 설명



```
<div id="root"></div>
<script src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

<div id = "root"></div> : React 앱이 렌더링될 컨테이너 역할
React 및 ReactDOM 스크립트를 외부에서 불러옵니다.
Babel 스크립트를 불러와 JSX 문법을 브라우저에서 바로 사용할 수 있게 합니다.

리액트 코드 설명



Member는 회원 정보를 화면에 출력하는 개별 컴포넌트입니다.

매개변수는 props(속성)를 객체 구조 분해 할당으로 받아옵니다.

HTML 태그 구조:

<dd>: 각 회원을 표현하며, 클래스 이름 / gender를 사용합니다.

(man 또는 woman).

<h3>: 회원의 썸네일 (현재는 빈 태그).

<div className="info">: 회원 정보를 담는 박스.

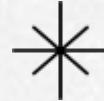
<h4>: 회원 이름, 성별, 나이.

<h5>: 직업과 지역.

<p>: 회원의 간단한 자기소개와 이상형.

```
<script type="text/babel">
  function Member({ name, s, age, gender, job, area, info1, info2 }) {
    return (
      <dd className={gender}>
        <h3 className="thum"></h3>
        <div className="info">
          <h4>
            <span>{name}</span>&ampnbsp&ampnbsp
            <em>| {s} | {age}<em>
          </h4>
          <h5>
            <span className = "mm">{job}</span>&ampnbsp
            <span className = "mm">{area}</span>
          </h5>
          <p>
            <em>나를 한줄로 표현하면</em>
            <span>{info1}</span>
          </p>
          <p>
            <em>이런 연인을 찾아요</em>
            <span>{info2}</span>
          </p>
        </div>
      </dd>
    );
  }
}
```

리액트 코드 설명



```
const members = [
  {
    name: "김**", s: "여성", age: 26, gender: "woman", job: "간호사", area: "서울특별시 강동구",
    info1: "진솔한 대화와 따뜻한 분위기를 만드는 것을 좋아해요.",
    info2: "치유의 힘을 믿으며, 서로의 이야기를 나누는 따뜻한 관계를 원해요."
  },
  {
    name: "조**", s: "여성", age: 36, gender: "woman", job: "약사", area: "충청남도 공주시",
    info1: "일상을 재미있게 만드는 작은 아이디어가 넘치는 장난꾸러기입니다.",
    info2: "진지함 속에서도 유머를 잃지 않는 사람을 기다립니다."
  },
  {
    name: "한**", s: "남성", age: 30, gender: "man", job: "심리학자", area: "경기도 용인시",
    info1: "심리학자라는 직업을 통해 사람을 이해하고 공감하는 걸 소중히 여깁니다.",
    info2: "속 깊고 따뜻한 대화를 나눌 수 있는 분을 기다립니다."
  }
];
```

members는 회원 데이터 객체의 배열
각 객체는 회원 1명의 정보를 담고 있으며,
총 10명의 회원 정보가 있습니다.

name : 이름

s : 성별

age : 나이

gender : CSS를 위한 성별 클래스 이름

job : 직업

area : 거주 지역

info1 : 본인을 표현하는 문구

info2 : 원하는 연인의 특성

리액트 코드 설명

ReactDOM.render: React 앱을 DOM의 root 엘리먼트에 렌더링
최종적으로 App 컴포넌트를 #root에 출력합니다.

```
function App() {
  return (
    <dl className="memList">
      {members.map((member, index) => (
        <Member key={index} {...member} />
      ))}
    </dl>
  );
}

ReactDOM.render(<App />, document.getElementById("root"));
```

App은 회원 목록을 출력하는 컴포넌트입니다.

members.map: 배열의 각 객체를 순회하며 Member 컴포넌트를 생성합니다.

key={index}: React에서 리스트 렌더링 시 고유 키로 사용.

{...member}: 스프레드 연산자를 사용하여 member 객체를 Member 컴포넌트에 전달.

The screenshot shows a user interface for a community platform. At the top left, there is a navigation bar with the number '5' and the text '안녕하세요' (Hello) next to a magnifying glass icon. The main title '커뮤니티 비밀글 가능' (Community Secret Message Available) is centered in a large, rounded rectangle with asterisks on either side. Below this, on the left, is a 'Q&A 작성' (Create Q&A) section. It includes fields for '제목' (Title) containing '안녕하세요' and '내용' (Content). At the bottom of this section are two red buttons: '작성' (Post) and '목록' (List). On the right, a message box displays the text '비밀글입니다. 작성자만 확인할 수 있습니다.' (It's a secret message. Only the author can verify it.) with a blue '확인' (Confirm) button. A callout box at the bottom right provides additional information: '커뮤니티에 비밀글 기능 추가' (Added secret message function to the community) and explains that while the community previously only supported public posts, it now includes a secret message feature. It notes that users can enable secret mode when creating a post by checking a box.

5 안녕하세요 🔎

* 커뮤니티 비밀글 가능 *

Q&A 작성

제목
안녕하세요

내용

작성 목록

비밀글입니다. 작성자만 확인할 수 있습니다.

확인

커뮤니티에 비밀글 기능 추가

기존 커뮤니티에서는 전체 공개 게시글만 작성했지만 현재는 비밀글 기능을 추가하였습니다.
게시글 작성 시 하단에 있는 비밀글 설정 체크를 누르게 되면 누구나 쉽게 비밀글을 작성할 수 있습니다.

커뮤니티 게시판 댓글 가능

안녕하세요

작성자 test101 | 작성일 2024-11-25

안녕 ~

댓글

작성

댓글

test1 2024-11-25

반가워요

작성

커뮤니티 사이트에 게시글을 작성한 후
작성된 게시글에 댓글을 달 수 있는
기능을 추가하였습니다.

댓글이 작성이 되면 DB 테이블에
저장된 댓글의 내용과 회원의 아이디가
화면에 표시되도록 구현하였습니다.

회원가입 이메일 인증 기능

```
import jakarta.mail.*;
import jakarta.mail.internet.*;
```

=> jakarta.mail이라는 java API를 이용

```
// 메일 작성
Message message = new MimeMessage(session);
message.setFrom(new InternetAddress(user));
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipientEmail));
message.setSubject(subject);
message.setText(content);

// 메일 전송
try {
    Transport.send(message);
    System.out.println("메일 전송 성공");
} catch (MessagingException e) {
    System.err.println("메일 전송 실패: " + e.getMessage());
    e.printStackTrace();
    throw e;
}
```



* 필수 입력

회원 등록	
아이디 *	<input type="text"/>
비밀번호 *	<input type="password"/> 8~20자 / 공백없이 만들어주세요.
이름 *	<input type="text"/>
성별 *	<input type="radio"/> 여자 <input type="radio"/> 남자
이메일 *	<input type="text"/> <input type="button" value="메일 인증"/>
인증번호 입력	<input type="text"/> <input type="button" value="인증"/>

회원 가입에 있는 이메일 인증을
받게 되면 사용자의 메일 주소로
인증번호가 발송됩니다.
받은 인증번호를 입력하면 해당 번호가
맞는 코드인지 확인하는 기능을
구현하였습니다.

매칭된 회원 간의 DM 가능 *

The first screenshot shows a conversation between two users. The user on the left sends "안녕하세요~~" (Hello~~) and the user on the right responds with "누구세요" (Who are you?). Both messages are highlighted in pink. The second screenshot shows a similar exchange where the user on the right's response is partially visible.

매칭된 상대가 없습니다.

확인

서로 매칭이 되었다는 전제하에 연락을 할 수 있는 DM 기능을 구현하였습니다.
매칭된 상대가 없는 경우에는 DM 창에 "메세지가 없습니다."라는 문구와 함께 매칭된 상대가 없다는 안내문구 창이 뜨게 됩니다.

