

팀 프로젝트 경진대회



프로그램 소개서

팀명: 명함소개팅

목차

I. 프로젝트 개요

- A. 프로젝트 명
- B. 구성명단
- C. 개요 및 목표
- D. 진행 방법 및 절차

II. 프로젝트 설계

- A. 기술 스택
- B. 시스템 구조
- C. 웹 아키텍처
- D. 데이터 흐름

III. 프로젝트 진행 내용

- A. 페이지별 세부 구현
- B. 주요 로직
- C. 개선해 나간 점

IV. 프로젝트 수행 결과

V. 설계 요소 평가

- A. 성능

- B. 안전성 및 보안성

- C. 구현 기간

VI. 추후 프로젝트 발전 방향

VII. 종합 토의

- A. 프로젝트 요약
- B. 개발 과장 평가
- C. 성과와 한계
- D. 개선 및 발전 가능성
- E. 개인 및 팀 학습 경험
- F. 결론

VIII. 참고문헌 및 첨부자료

- A. 백엔드 API 설계 문서
- B. SQL 쿼리
- C. 외부API 문서
- D. 배포 링크

I. 프로젝트 개요

A. 프로젝트 명

💌 디지털 명함 소개팅 💌

B. 구성명단

성명	소속	학년	연락처	e-mail
정다운	컴퓨터공학과	2	010-8868-4734	daun5535@gmail.com
문효재	컴퓨터공학과	2	010-5495-9050	dsdk1088@gmail.com
조유찬	컴퓨터공학과	2	010-6624-7408	yuchancho174@gamil.com
이서인	컴퓨터공학과	2	010-5243-3163	guapapamama@gmail.com

C. 개요 및 목표

디지털 명함 소개팅 사이트는 디지털 명함을 활용하여 자신을 소개하고 이성을 매칭시켜주는 혁신적인 대학생 소개팅 플랫폼으로 기존의 명함 소개팅의 단점인 단조로운 진행 방식 및 시공간의 제약을 보완하여 접근성을 높이는 방향으로 구현된 콘텐츠이다.

디지털 명함 소개팅의 차별성은 다음과 같다.

1. 동물 사진 및 명함 색상 선택 기능 : 기존 명함 소개팅에는 없던 항목으로, 16가지의 동물과 8가지 색상을 각각 선택하여 부담없이 개성을 드러내며 작성할 수 있고 뽑는 사람은 명함 선택 시에도 고려사항으로 참고가 가능하다.
2. 명함 추천 서비스 기능 : 랜덤으로 2장의 명함을 추천하여 뽑거나 사전 이상형 조사를 통해 개인화된 알고리즘을 통해 최상의 명함 1장을 제공한다.
3. 대학생 인증 API 사용 : 외부 대학생 API로 신분 인증 및 로그인을 함으로써 대학생이

아닌 외부인이 서비스를 이용하거나 악용하는 것을 막고 번거로운 회원가입 절차를 거치지 않고 인증코드로 간편하게 로그인 할 수 있다.

4. 횟수 제한 : 인증된 대학생은 1회만 참여할 수 있게 하여 중복 및 도배를 방지하고, 보다 더 신중한 명함 작성 및 뽑기를 유도한다. 다만 영구적으로 제한하진 않고 주기적으로 인증된 유저를 초기화한다.
5. AI 챗봇 시뮬레이션 : 이성과의 소통이 어색하거나 어려운 사용자를 위해 자연스러운 대화 연습 환경을 제공하는 AI 기반 챗봇을 지원한다.
6. 영어 지원: 각 페이지별로 중요한 공지사항은 영어멘트도 추가하여 외국인 학생들도 사용할 수 있도록 하였다.
7. 웹 접근성: 시맨틱 태그와 aria-label속성을 이용해 신체적으로 불편하신 분들을 위해 웹 접근성도 최대한 고려하여 제작하였다.

프로젝트 목적은 다음과 같다.

1. 디지털 명함을 활용하여 MZ 대학생의 라이프 스타일에 맞는 효율적이고 편리한 소개팅 플랫폼을 제공한다.
2. 관심사 공유를 통해 선택의 폭을 좁히고 대상 선택의 어려움을 해소한다.
3. 오프라인 명함 소개팅의 시간적, 공간적 제약을 극복하여 접근성을 확장한다.
4. 대면 만남의 부담감을 줄이고 디지털 기반의 안전하고 편안한 소통 환경을 제공한다.

타겟층은 다음과 같다.

- 대학생 누구나

D. 진행 방법 및 절차

사용자 페이지는 다음과 같은 순서로 진행된다.

- 홈 페이지(onboarding) - 로그인 페이지 - 명함 작성 페이지 - 이상형 조사 페이지 - 매칭 서비스 선택 페이지 - 랜덤 뽑기 페이지/추천 뽑기 페이지 - AI 챗봇 페이지

관리자 페이지는 다음과 같은 순서로 진행된다.

- 관리자 로그인 페이지 - 사용자 관리 및 모니터링 페이지

역할 분담 및 협업은 다음 방식과 같다.

- 각 팀원은 역할을 분담하여 한 페이지씩 개발을 진행한다.
- 수업 시간 외에도 추가 회의 시간을 정해 충분히 논의하고 피드백을 주고받으며 협업한다.
- 매주 최소 하루는 프로젝트 전용 회의 시간을 정해 다음 개발 계획과 수정을 논의한다.

프론트엔드 개발 전략은 다음과 같다.

- 사용자 중심 UI/UX : 사용자에게 깔끔하고 직관적인 인터페이스를 제공하고 시각적 요소와 동적 기능을 추가하여 사용자의 흥미를 유도한다.
- 디자인 품질 향상 : 전문적인 디자인 프로세스를 통해 사이트의 시각적 완성도를 높이고 반응형 웹 디자인을 적용하여 다양한 기기에서 최적의 사용성을 보장한다.

백엔드 및 데이터베이스 기술을 도입한다.

- 데이터 관리 시스템 : 사용자 데이터를 체계적으로 관리할 수 있는 데이터베이스 구조를 설계하여 데이터 처리 및 관리를 위한 안정적인 백엔드 서비스를 구축한다.
- 주요 기능 개발 : 명함 관리 시스템과 뽑기 기능을 통하여 사용자 정보를 저장하고 관리하는 기능과 사용자 참여를 유도하는 동적인 뽑기 시스템을 구현한다.

개발 프로세스 관리는 다음과 같다.

- 프로젝트 관리 도구를 활용하여 작업 상태와 진척도를 관리하고 코드 리뷰와 테스트 과정을 주기적으로 수행하여 품질을 유지한다.

II. 프로젝트 설계

I. 기술 스택

프론트엔드에 사용된 기술 스택은 다음과 같다.

- Vanilla JS: 클라이언트 측 동작을 제어하며, REST API 요청 및 DOM 조작 구현을 담당한다.
- CSS Modules: 사용자 인터페이스(UI)의 스타일링 및 반응형 디자인(Responsive Web)을 담당한다.

- HTML: 애플리케이션의 구조와 콘텐츠 작성을 담당한다.
- Figma: 클라우드 기반 협업 디자인 도구로, 와이어프레임, UI/UX 디자인, 프로토타이핑을 효율적으로 지원한다.

백엔드에 사용된 기술 스택은 다음과 같다.

- Flask: 경량 Python 웹 프레임워크로 서버 구축, API 설계, 라우팅을 간편하게 관리한다.
- SQLite: 경량화된 관계형 데이터베이스로 데이터 빠르고 독립적인 데이터 저장소를 제공한다.
- Python: 플라스크와 백엔드 조작에 쓰이는 주언어로, 객체지향적이고 간결하고 읽기 쉬운게 특징이다.

기타 사용된 기술 스택은 다음과 같다.

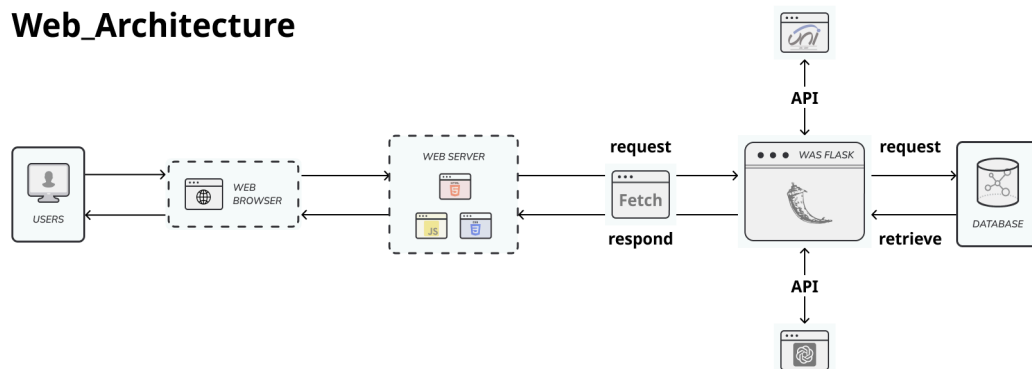
- PyCharm: 프로젝트별 독립적인 가상환경(Virtual Environment) 관리를 지원하는 Python 전용 IDE이다.
- Heroku: 클라우드 플랫폼으로 간편한 서버 배포와 확장성을 제공한다.

II. 시스템 구조

본 웹사이트는 클라이언트-서버 구조로, VanillaJS 기반 클라이언트와 Flask 기반 백엔드 간 REST AP 통신을 통해 유저와 서버간의 원활한 상호작용을 지원한다. 또한 SQLite로 데이터베이스를 설계하여 사용자 정보와 매칭 데이터를 효율적으로 관리한다.

III. 웹 아키텍처

Web_Architecture



IV. 데이터 흐름

데이터 흐름(flow)가 발생하는 동작은 다음과 같다.

- 사용자 프로필(명함) 등록

1. 사용자가 클라이언트에서 데이터를 입력한다.
2. API를 통해 서버가 데이터를 수신한다.
3. 서버가 받은 데이터를 DB를 저장한다.

- 추천 프로세스

1. 사용자가 이상형 설문조사를 통해 정보를 입력한다.
2. 입력된 정보가 브라우저 세션 스토리지에 저장된다.
3. 사용자가 추천 서비스를 요청하면 세션 스토리지에 데이터를 가져와 이상형과 가장 유사한 명함을 DB에서 조회한다.
4. 서버가 조회된 명함 데이터를 브라우저로 전송한다.
5. 사용자가 명함을 확정하면, 해당 데이터는 DB에서 삭제된다.
6. 이후 AI 챗봇 페이지에서 사용하기 위해 세션 스토리지에 명함 데이터를 저장한다.

- 랜덤 프로세스

1. 서버가 DB에서 랜덤으로 2개의 명함 데이터를 조회한다.
2. 조회된 데이터를 브라우저로 전송한다.
3. 사용자가 명함을 확정하면, 해당 데이터는 DB에서 삭제된다.
4. 이후 AI 챗봇 페이지에서 사용하기 위해 세션 스토리지에 명함 데이터를 저장한다.

- AI 챗봇

1. 세션 스토리지에서 확정된 명함에 대한 데이터를 가져온다.
2. 명함 데이터는 챗봇 캐릭터 구성을 위해 활용된다.

III. 프로젝트 진행 내용

A. 페이지별 세부 구현

본 페이지는 사용자 페이지와 관리자 페이지로 나뉘며, 각각 사용자 경험 향상과 안정적인 서비스 운영에 목표를 두어 설계되었다. 또한 핸드폰 화면크기에 대한 시장조사를 통해 최근 5개년간 출시된 핸드폰의 width를 분석했다. 그 결과 390px이라는 평균값을 도출할 수 있었고, 미디어 쿼리를 사용하여 태블릿과 스마트폰 환경에서도 사용가능한 UI/UX를 제공하였다. 태블릿 보다는 스마트폰을 많이 이용할 것을 고려했지만, 기존 스타일링이 태블릿 화면 크기(width: 834px)에서도 깨지는것을 감안하여 조건은 834px로 설정하였다.

사용자페이지

1. 홈 페이지(onboarding page)

해당 페이지는 플랫폼 소개, 사용자 환영 메시지, 간단한 사이트 이용 가이드를 제공한다.

세부 구현사항:

- 7개의 섹션으로 구성된 하향식 레이아웃을 제공한다.
- 사용자가 리프레쉬할 수 있는 포춘카드라는 미니 콘텐츠를 제공한다.
- 직접 디자인한 이미지를 활용하여 산뜻하고 직관적인 UI를 제공한다.
- 링크 공유시 사이트 제목과 이미지를 나타내주는 OpenGraph를 적용하였다.

모바일 구현사항: 기존 섹션들을 최대한 줄여 쾌적한 사용자 경험을 제공하였다. 3개의 섹션으로 제공되던 가이드 라인을 한 섹션안에 요약하였다. 그리고 home 섹션도 삭제하였다.

2. 로그인 페이지(login page)

해당 페이지는 'Univcert API'라는 외부 API를 활용하여 간편한 로그인 및 학교 인증 서비스를 제공한다.

세부 구현사항:

- 대학교 이름, 학교 이메일, 인증 코드를 통한 유효성 검사로 간단한 로그인

절차를 구현했다.

- b. Key값 보호를 위해 백엔드에서 API Key를 활용하여 안전한 인증 절차를 구현했다.
- c. 인증된 사용자에게 30분 동안 세션을 부여하고 만료 시 자동 로그아웃 후 로그인 페이지로 리다이렉션하게 설계했다.
- d. 무분별한 사이트 이용을 방지하기 위해 한번 인증 후 세션이 만료된 유저는 더이상 사이트를 이용할 수 없게 설계하였다.
- e. URL을 이용한 강제 사이트 이동을 막기 위해 checkAuth 함수로 세션이 부과되지 않은 이용자들을 로그인 페이지로 리다이렉션하도록 설계하였다.

모바일 구현사항: 기존 스타일을 최대한 유지하고, 모바일 환경에 맞게 flex-direction을 column으로 변경하였다.

3. 명함 작성 페이지(writing page)

해당 페이지는 사용자 맞춤형 디지털 명함을 작성하고 제출하는 기능을 제공한다.

세부 구현사항:

- a. 예시를 제공하여 올바른 양식을 작성하도록 유도하고 올바르지 않은 양식을 제출하였다면 양식검사 알고리즘을 통해 어느부분을 수정해야하는지 알림을 제공한다. 양식에는 성별, 이름, 학과, 학번(나이), MBTI, 취미, SNS ID가 있다.
- b. 기존 오프라인에는 없던 카드 색상 및 동물 이미지 선택 옵션을 추가하여 개성표현의 범위를 확장하였다.
- c. 백엔드와 통신하여 폼 data를 JSON형식으로 서버로 전송하고 서버에서 다시 DB에 저장한다.
- d. 성별에 관한 정보는 세션 스토리지에 저장하여 이성 명함만을 추출하기위한 프로세스에서 사용된다.

모바일 구현사항: 명함의 크기를 줄이고, 기존 UI의 틀을 유지하기 위해 여러 속성값들을 변경하였다.

4. 이상형 조사 페이지(Survey page)

해당 페이지는 이상형 데이터를 수집하여 추천 매칭 알고리즘에 반영하는 역할을 한다.

세부 구현사항:

- a. 나이, 학과, MBTI, 취미 등을 선택하고 입력받을 수 있는 데이터 수집 기능을 제공한다. 나이는 연상, 연하, 동갑으로 나뉘며 버튼 별 데이터 라벨링을 통해 특정 키워드를 string 형태로 서버에 전송하여 추천서비스를 처리할 수 있게 돕는다. 학과는 문과, 이과, 공대, 음대, 체대, 미대로 나뉘 각각의 분류에 맞게 미리 학과정보를 배열의 형태로 저장하여 전송한다. 이때 현재는 서울과학기술대학교를 서비스 대상으로 설정하여 음대는 존재하지 않는다. mbti는 선택한 문자들을 조합하여 4글자 형태의 string 형식으로 전송한다. 취미는 키워드별로 나누어 배열에 저장하여 전송한다.
- b. 선호 나이 및 학과는 복수 선택이 가능하도록 설계하였다.
- c. 선택 시 selected된 요소에 대한 즉각적인 시각적 피드백을 제공한다.
- d. 전송시 설문 데이터를 따로 데이터베이스에 저장할 필요없이 세션 스토리지에 저장하여 이후 추천알고리즘을 수행하기 위한 재료로 이용된다.

모바일 구현사항: flex-direction을 column으로 변경하여 모바일 환경에 맞게 최적화 하였다.

5. 매칭 서비스 선택 페이지(drawing page)

해당 페이지는 사용자 기호에 맞게 랜덤 또는 추천 명함 매칭을 선택하도록 하는 페이지이다.

세부 구현사항:

- a. 사용자에게 두 가지 옵션을 직관적으로 제공하도록 UI/UX를 설계하였다.
- b. 명함 뽑기 시 트리플7과 같은 애니메이션을 적용하여 역동적인 경험을 제공하도록 하였다.

모바일 구현사항: flex-direction을 column으로 변경하여 모바일 환경에 맞게 최적화 하였다.

6. 랜덤 뽑기 페이지(random page)

해당 페이지는 랜덤 명함 2장을 제공하고 선택할 수 있는 기능을 제공한다.

세부 구현사항:

- a. 서버에서 랜덤으로 가져와 명함을 교차하여 표시한다.
- b. 이름과 SNS ID는 사용자가 확정하기 버튼을 누를 때만 노출하도록하여 정보의 보안성을 높였다.
- c. 오픈 버튼을 누르기 전까진 카드를 볼 수 없고, 확정하기 버튼을 누르기 전까진 다음 페이지로 이동할 수 없게 하여 사용자의 흐름을 올바르게 제어하도록 설계하였다.
- d. 확정된 명함에 대한 데이터를 세션 스토리지에 저장하여 이후 AI 챗봇 페이지에서 사용할 수 있도록 하였다.

모바일 구현사항: 명함의 크기를 줄이고, 기존 UI의 틀을 유지하기 위해 여러 속성값들을 변경하였다.

7. 추천 뽑기 페이지(recommend page)

해당 페이지는 사용자 데이터를 기반으로 최적의 명함을 1장 추천해준다.

세부 구현사항:

- a. 작성된 명함과 이상형 조사 데이터를 기반으로 필터링 및 매칭 알고리즘을 실행하여 최적의 명함을 추천한다.
- b. SNS ID는 사용자가 확정하기 버튼을 누를 때만 노출시켜 정보의 보안성을 높였다.
- c. 오픈버튼을 누르기 전까진 카드를 볼 수 없고, 확정하기 버튼을 누르기 전까진 다음 페이지로 이동할 수없게 하여 사용자의 흐름을 올바르게 제어하도록 설계하였다.
- d. 확정된 명함에 대한 데이터를 세션 스토리지에 저장하여 이후 AI 챗봇 페이지에서 사용할 수 있도록 하였다.

모바일 구현사항: 명함의 크기를 줄이고, 기존 UI의 틀을 유지하기 위해 여러 속성값들을 변경하였다.

8. AI 챗봇 페이지(chatbot page)

해당 페이지는 이성과의 대화 연습을 위한 AI 챗봇, 격려 메시지, 이용자의 피드백을 수용하기 위한 구글 폼 링크를 제공한다.

세부 구현사항:

- a. ChatGPT API를 이용하여 인공지능을 활용한 자연스러운 대화 흐름을 구현하였다.
- b. 시스템 메시지로 캐릭터별 role을 부여한 후 전송하여 취미 성격에 따라 구별되는 7가지 캐릭터를 제공하고, 확정된 명함을 기반으로 구성된 커스터마이징 캐릭터도 제공한다.
- c. 사용자의 입력에 따라 즉각적인 응답을 생성할 수 있도록 설계하였다.

모바일 구현사항: 모바일 환경에 맞게 레이아웃 크기를 줄이고 폰트크기를 줄였다.

개발자 페이지

1. 관리자 로그인 페이지(adminLogin page)

해당 페이지는 관리자 전용 페이지 접근을 위한 안전한 로그인 환경을 제공한다.

세부 구현사항:

- a. 입력한 비밀번호에 대한 검증을 안전하게 백엔드에서 구현하였다.
- b. URL을 이용한 강제 사이트 이동을 막기 위해 checkAuth 함수로 세션이 부과되지 않은 이용자들은 로그인 페이지로 리다이렉션하도록 설계하였다.

모바일 구현사항: 모바일 환경에 맞도록 요소 크기를 줄였다.

2. 사용자 관리 및 모니터링 페이지(master page)

해당 페이지는 인증된 사용자를 관리하고, 초기화할 수 있는 기능을 제공하며, 사이트 상태를 모니터링할 수 있는 사이트 별 하이퍼링크를 제공하고, DB를 조회하고 삭제할 수 있는 기능을 제공한다.

세부 구현사항:

- a. 직관적인 디자인으로 손쉬운 사용편의를 제공한다.
- b. Univcert API를 이용하여 인증유저를 출력하고 삭제할 수 있는 버튼을 구현하여 개별 유저 초기화 및 전체 초기화 버튼을 제공한다.
- c. SQLite의 ORM(Object Relation Mapping)이용하여 DB정보를 웹 페이지로 쉽게 가져와 조회하고 삭제할 수 있게 설계하였다.

모바일 구현사항: 모바일 환경에 맞도록 요소 크기를 줄이고, 레이아웃 배치를 수정하였다.

B. 주요 로직

해당 웹 사이트에서 가장 중요한 로직은 사용자의 관심사와 선호 데이터를 기반으로 최적의 매칭을 수행하는 매칭 알고리즘이다. 코드는 다음 링크로 대체한다.

[추천 알고리즘](#)

1. 세션에서 사용자 정보를 가져와 데이터베이스에서 사용자 프로필을 조회.
2. 프로필이 없으면 404 에러 반환, 사용자의 성별을 기반으로 상대 성별 필터링.
3. 사용자 제외, 상대 성별, 나이 조건, 선호 전공/MBTI/취미를 기반으로 필터링.
4. 조건에 맞는 프로필 중 랜덤으로 1개를 조회해 필요한 정보를 반환.
5. 조회된 데이터가 없으면 기본 조건으로 이상형 데이터를 생성.
6. 모든 데이터를 벡터화 후 코사인 유사도를 계산해 이상형과 가장 유사한 프로필 선택.
7. 최종 선택된 프로필 데이터를 JSON 형식으로 반환한다.

C. 개선해 나간 점

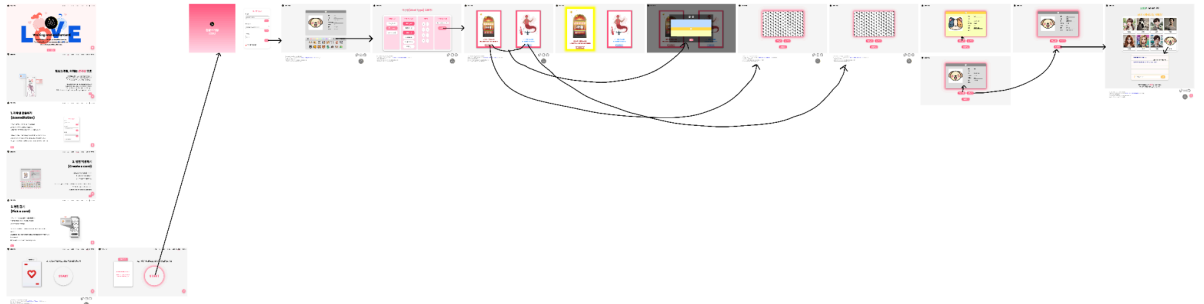
- 초기 문제: API키를 프론트엔드 JS 코드에 포함하여 노출 위험성이 있었음.
- 개선 방법: 프론트에서는 백엔드로 알맞는 동작요청을 패치하고, 백엔드에서 API 키를 처리하도록 변경하여, 브라우저에서 직접 외부 API와 통신하지 않도록 구조를 개선하여 보안성을 크게 향상시켰다.

IV. 프로젝트 수행 결과

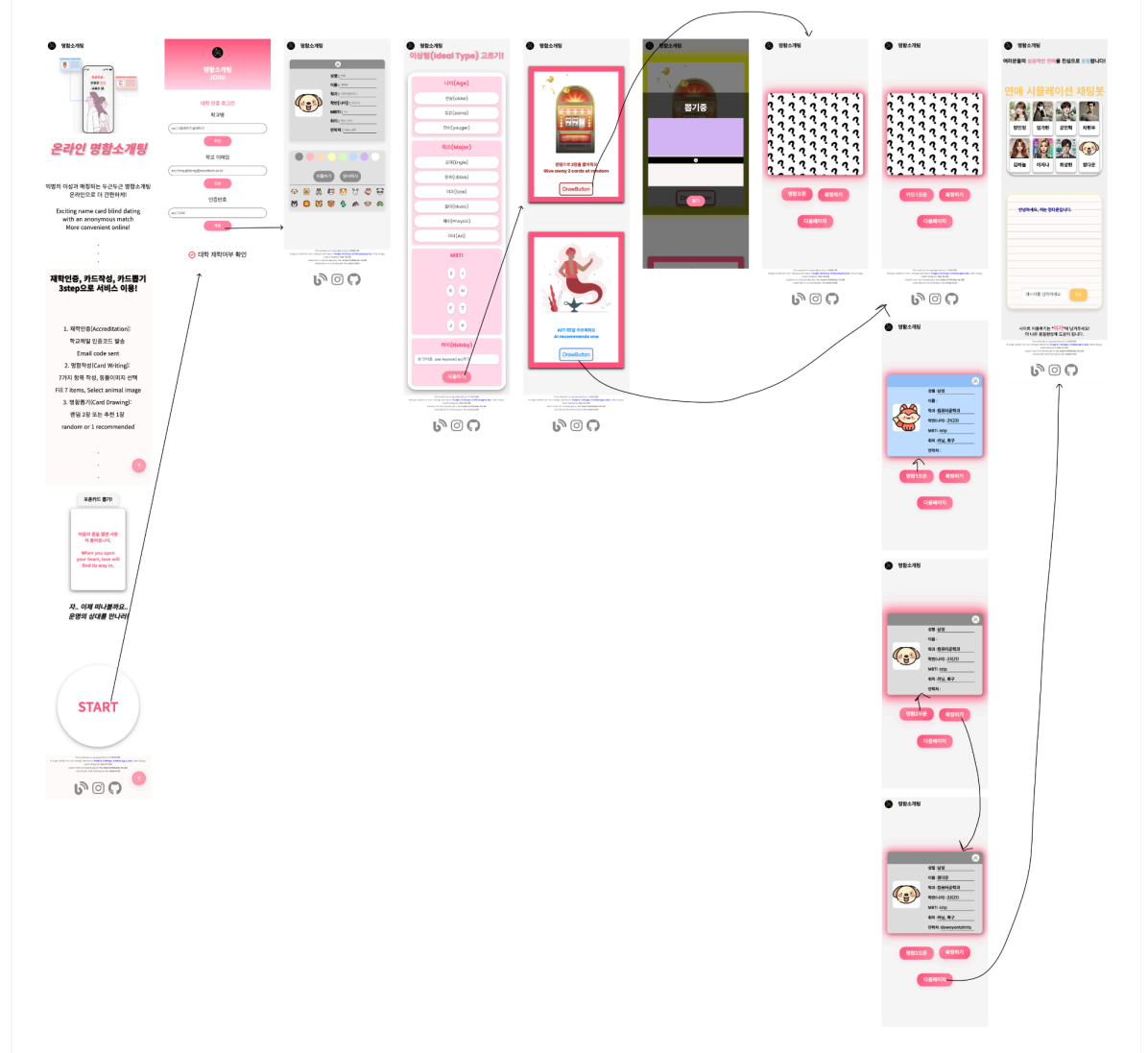
페이지별 결과물

User WireFrame

PC.ver



Mobile.ver



※ 이미지 클릭 후 다운로드 시 더 고화질의 이미지로 확인할 수 있음

Administrator WireFrame



※ 이미지 클릭 후 다운로드 시 더 고화질의 이미지로 확인할 수 있음

V. 설계 요소 평가

A. 성능

이번에 제작한 웹 사이트는 기능적으로는 의도한대로 작동하며, 다양한 예외상황을 고려하여 제작되었기 때문에 문제가 없다. 다만 앞서 말했듯이 실제로 서비스화를 한다면, SQLite의 단점을 극복하기 위해 DBMS를 변경하고, 보안 취약점은 없는지 한번 더 검토 후, 디자인을 더 세련되게 마감한다면, 이용하는데 실사용화하여도 충분히 괜찮은 성능을 보인다. 다만 배포시 외부 서버를 이용하기 때문에 안전성은 높지만, 약간의 로드 딜레이가 발생함을 확인할 수 있었다. 마지막으로 미디어 쿼리를 통해 모바일 환경에 알맞은 UI/UX를 제공하여 스마트폰 유저들의 편의성을 높였지만, 여전히 최적화는 조금 더 필요할 것으로 보인다.

B. 안전성 및 보안성

이미 다른 유저들도 많이 사용해본 Heroku를 사용하여 안정적인 서버를 제공하며, 다양한 예측 가능한 경우들을 고려하여 사용자 이탈(User Drop-off), 사용자 흐름 단절(Flow Interruption), 사용자 흐름 분산(Flow Diversion)등의 문제를 최대한 막고자 했다. 예를 들어 로그인시 시간제한이 존재하는 세션을 부과하여 과도한 사이트 이용을 막았다. 'checkAuth()' 함수를 활용하여 사용자의 악의적인 리다이렉팅을 막았다. 외부 API 통신을 할

때도 백엔드를 중간다리로 활용하였다. 이를 통해 프론트엔드에서 바로 통신함으로써 발생할 수 있는 여러 보안 취약점들을 보안하였다. 또한 관리자 페이지를 추가로 만들어 유저들을 효과적으로 관리할 수 있게 설계하였고, 관리자 페이지로 들어가기 위한 비밀번호 검증 절차 또한 백엔드에서 진행하여 비밀번호 유출을 최소화 하도록 설계하였다.

C. 구현기간

11/4~11/11: 온보딩 및 로그인 페이지 구현

11/11~11/18: 명함작성 페이지 구현

11/18~11/25: 서비스 선택 페이지 및 명함뽑기(랜덤, 추천) 구현

11/25~12/2: AI 챗봇 시뮬레이션 페이지 제작

12/2~12/9: 반응형 웹 구현 및 이상형 조사 페이지 제작, 로그인 페이지 디자인 수정

12/9~12/14: 관리자 페이지 제작, 디자인 최종 마감 및 백엔드 구현, 배포 및 디버깅

VI. 추후 프로젝트 발전 방향

해당 목차에서는 본 웹사이트의 한계와 개선 방안에 대해서 논의해보도록 하겠다.

A. 세션 정보의 신뢰성 부족

- 한계점: 세션 값은 클라이언트에서 쉽게 조작될 수 있어 신뢰도가 낮다.
- 개선 방안: 추가적인 서버 검증 및 암호화를 통해 세션 데이터의 안전성을 확보한다

B. UI/UX 최적화 문제

- 한계점: 모바일(폭 390px 이하) 및 노트북(16인치 이하) 환경에만 최적화 되어 있어 다른 크기의 디바이스에서 레이아웃이 깨져 사용자 경험이 저하되는 문제가 발생한다.
- 개선 방안: 반응형 디자인을 강화하여 다양한 화면 크기에서 최적화된 UI/UX를 제공한다.

C. 디자인 마감 문제

- 한계점: 최적화 작업 중 동적 요소의 움직임으로 인해 다른 요소 레이아웃이 변경되는

현상이 간혹 있다.

- 개선 방안: 피그마를 적극 활용하여 세밀한 디자인과 레이아웃을 설계하고, 팀원간의 디자인 설계 협업을 통해 레이아웃과 사용자 경험을 더욱 발전시킬 방향을 모색한다.

D. 데이터베이스의 한계

- 한계점: SQLite는 쓰기 작업에서 단일 스레드만 지원하여 다중 사용자 환경에 부적합하고, 파일 기반 DB로, 네트워크 접근, 사용자 권한 관리, 복제 기능 등이 제한적이어서 새로운 버전 배포 시 기존 데이터베이스 파일이 덮어쓰워져 데이터 손실 위험이 있다.
- 개선 방안: 동시성 처리가 가능한 클라우드 기반 데이터베이스로 전환하거나 데이터 접근을 관리할 수 있는 미들 웨어를 도입한다.

E. 협업 및 깃 커밋 문제

- 한계점: 프로젝트 협업 경험의 부족으로 잦은 코드 리뷰가 있었고, 깃허브 및 깃 경험 또한 부족하여 푸쉬오류와 충돌이 빈번했다.
- 개선 방안: 명확한 협업 프로세스를 수립하고 정기적인 소통 회의를 진행한다. 또한 브랜치 전략과 깃 사용 방법에 대한 사전 교육을 제공한다.

F. 추천 알고리즘 성능 문제

- 한계점: 현재 추천 로직은 모든 데이터를 벡터화하고 유사도를 계산하며, 데이터가 많아질수록 성능 저하 가능성이 크다.
- 개선 방안: 미리 계산된 유사도 데이터를 저장하고 캐싱을 활용하여 성능 저하를 방지한다.

G. 테스트 부재

- 한계점: 주요 로직과 에러 상황에 대한 테스트가 부족하다.
- 개선 방안: 유닛 테스트와 통합 테스트를 추가하여 주요 기능과 에러 상황을 점검하여 코드 품질을 유지하고 신뢰성을 향상시킨다.

VII. 종합 토의

A. 프로젝트 요약

디지털 명함 소개팅 프로젝트는 대학생들을 대상으로 한 새로운 소개팅 시스템을 구현하는데 목적을 둔 프로젝트로, 주요 기능으로는 명함 작성, 이상형 조사, 명함 추천 및 랜덤 추천 서비스, AI 챗봇 대화 등이 포함된다.. 개발 결과는 목표로 했던 모든 주요 기능을 성공적으로 구현하였으며, 사이트 전반의 성능과 안정성 면에서도 우수한 결과를 내었다.

B. 개발 과정 평가

개발 과정은 프론트엔드와 백엔드 간 협업과 지속적인 코드 리뷰를 통해 관리되었다. 프로젝트 초기에는 UI/UX 디자인과 기본 구조 설계를 마쳤고, 이후 주요 기능별로 작업을 부담하여 효율적인 개발이 가능했다. 주요 문제는 API 보안 관리 세션 유지 관리에서 발생했으며, 이를 백엔드 검증과 암호화를 통해 해결하였다.

C. 성과와 한계

성과:

- 명함 작성 및 추천 서비스, 사용자 관리 등 주요 기능이 정상적으로 동작한다.
- 웹 접근성과 모바일 대응을 고려한 반응형 웹 사이트를 구현한다.
- 대학생 인증 API를 통한 신뢰성 있는 사용자 인증 시스템을 구축한다.

한계:

- SQLite 기반 DB의 확장성이 부족하다.
- 일부 복잡한 레이아웃에서 디자인의 미세한 조정이 필요하다.
- 추천 알고리즘 성능이 데이터 양의 증가에 따라 저하될 가능성이 높다.

D. 개선 및 발전 가능성

- 데이터베이스 업그레이드: 더 큰 프로젝트 확장을 위해 MySQL 또는 PostgreSQL로 데이터베이스 이전을 고려한다.
- 디자인 최적화: 모바일과 태블릿, 데스크탑을 포함한 다양한 해상도 대응을 위한 반응형 웹 디자인의 최적화가 필요하다.
- 추천 알고리즘 고도화: 현재 유사도 계산 알고리즘을 개선하고, 미리 계산된 데이터를 캐싱하여 성능을 향상한다.
- 테스트 자동화: 주요 기능에 대한 자동화된 유닛 테스트와 통합 테스트를 추가한다.

E. 개인 및 팀 학습 경험

본 프로젝트를 통해 웹 개발의 중요한 부분인 프론트엔드와 백엔드 간의 협업, 데이터베이스의 설계, API 관리 등 다양한 기술적 요소를 터득할 수 있게 되었다. 또한, Git 협업과 코드 관리, 팀워크 능력을 키울 수 있게 되었다.

- 기술적 역량: 프로젝트를 통해 JavaScript와 Flask 기반 API 설계 및 데이터베이스 관리 능력이 강화되었다. 또한, 클라이언트-서버 간 통신과 데이터 흐름 관리, 반응형 웹 구현 등 실무적인 웹 개발 경험을 축적할 수 있었다.
- 협업 및 소통 능력: 프로젝트 진행 동안 팀원 간 역할을 분담하고 정기적인 코드 리뷰와 회의를 통해 피드백을 주고받으면서 협업 역량이 향상되었다. 특히, Git을 활용한 코드 관리와 충돌 해결, 브랜치 관리 등 협업 도구 사용법을 숙달할 수 있었다.
- 문제 해결 및 창의적 사고: 예상치 못한 오류와 보안 문제를 해결하는 과정에서 문제 해결 능력이 향상되었으며, 디자인 측면에서는 사용자가 직관적으로 사용할 수 있는 UI/UX를 구현하기 위해 창의적인 사고를 발휘하였다.
- 프로젝트 관리: 개발 일정 수립, 작업 우선순위 결정, 프로젝트 관리 도구 사용 등 프로젝트 관리 능력도 강화되었다. 일정 관리와 코드 품질 유지, 주요 기능의 테스트 계획 수립 등 개발 프로세스 전반을 경험할 수 있었다.

F. 결론

디지털 명함 소개팅 사이트는 대학생을 위한 창의적이고 혁신적인 플랫폼을 제공하는 것을 목표로 주요 기능을 성공적으로 구현하였으며, 프로젝트를 통해 다양한 웹 개발 기술과 협업 방식을 경험할 수 있었다. 추후 개선 사항을 반영하여 더욱 완성도 높은 서비스를 개발할 수 있는 기반을 마련하였다.

VIII. 참고문헌 및 첨부자료

A. 백엔드 API 설계 문서

[백엔드 API 설계 문서](#)

B. SQL쿼리

[SQL 쿼리 명세서](#)

C. 외부 API문서

[Univcert API 공식 사이트](#)

[ChatGPT API 공식 사이트](#)

[Jump to 플라스크](#)

D. 배포링크

[사이트 이용 링크](#)

[깃허브 링크](#)

[깃허브 페이지](#)

※ 원활한 이용을 위해 이용(배포)링크를 이용해 주세요. 다른 설치방법은 존재하지 않습니다. 또한 깃허브 폴더에서 파일을 참고하실 때, front_part가 아닌 back_part 폴더를 참고해주세요. 완성된 프론트 및 백엔드 코드가 해당 폴더안에 존재합니다.