

CAPSTONE 2022

# EYIING

아 이 트 래 킹   P P T   분 석   서 비 스   플 랫 폼

TEAM 44

# Team Introduction



**송영진**  
(\*\*\*\*0176)

**LEADER**  
FRONT-END

- 웹 사이트 기획 및 제작
- Eye Tracking 모델 적용



**고동훈**  
(\*\*\*\*3027)

FRONT-END

- 웹 사이트 제작
- Eye Tracking 모델 적용



**김창규**  
(\*\*\*\*2347)

BACK-END

- 서버 구축 및 배포
- 데이터베이스 모델링 및 API 개발



**채지윤**  
(\*\*\*\*1313)

BACK-END

- 서버 구축
- API 개발 및 데이터 시각화



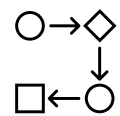
# Index



## Chap. 1

### 프로젝트 소개 (Project Introduction)

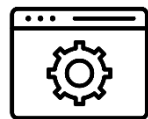
- 서비스 개요 (Service Summary)
- 서비스 목표 (Service Goals)
- 프로세스 (Process)



## Chap. 2

### 프로젝트 구조 (Project Architecture)

- 시스템 구조 (System Architecture)



### Chap. 3

## 핵심 기능 (Core Features)

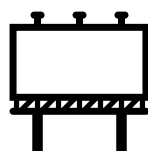
- 시선 추적 (Eye Tracking)
- 데이터 시각화 (Data Visualization)



### Chap. 4

## 서비스 소개 (Service Introduction)

- 웹 사이트 (Web Site)



### Chap. 5

## 시연 (Demonstration)

- 시연 영상 (Video, 46s)

# 01

## 프로젝트 소개

---

### 개요

어떤 서비스를 제공하나요?

---

### 목표

무엇을 얻을 수 있나요?

---

### 프로세스

어떻게 제공하나요?



CAPSTONE 2022

## 프로젝트 소개

# PPT ANALYSIS

PPT 시청자 시선을 분석하여  
피드백을 제공하는 웹 서비스



## 프로젝트 소개



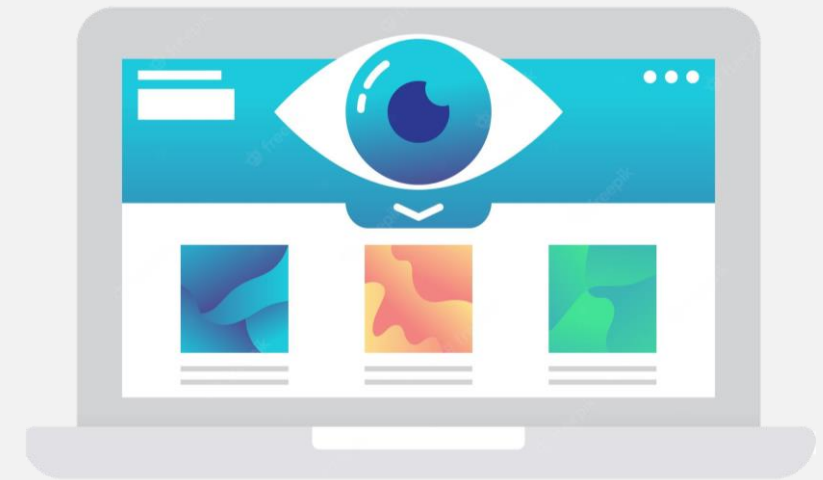
01

사용자가 분석을  
원하는 자료 업로드



02

다른 사용자  
시선 데이터 수집



03

데이터 시각화로  
자료 피드백 제공

# 프로젝트 소개



발표 전 분석 데이터를 통해 자료 수정에 도움  
효과적인 발표 자료 제작 가능



# 02

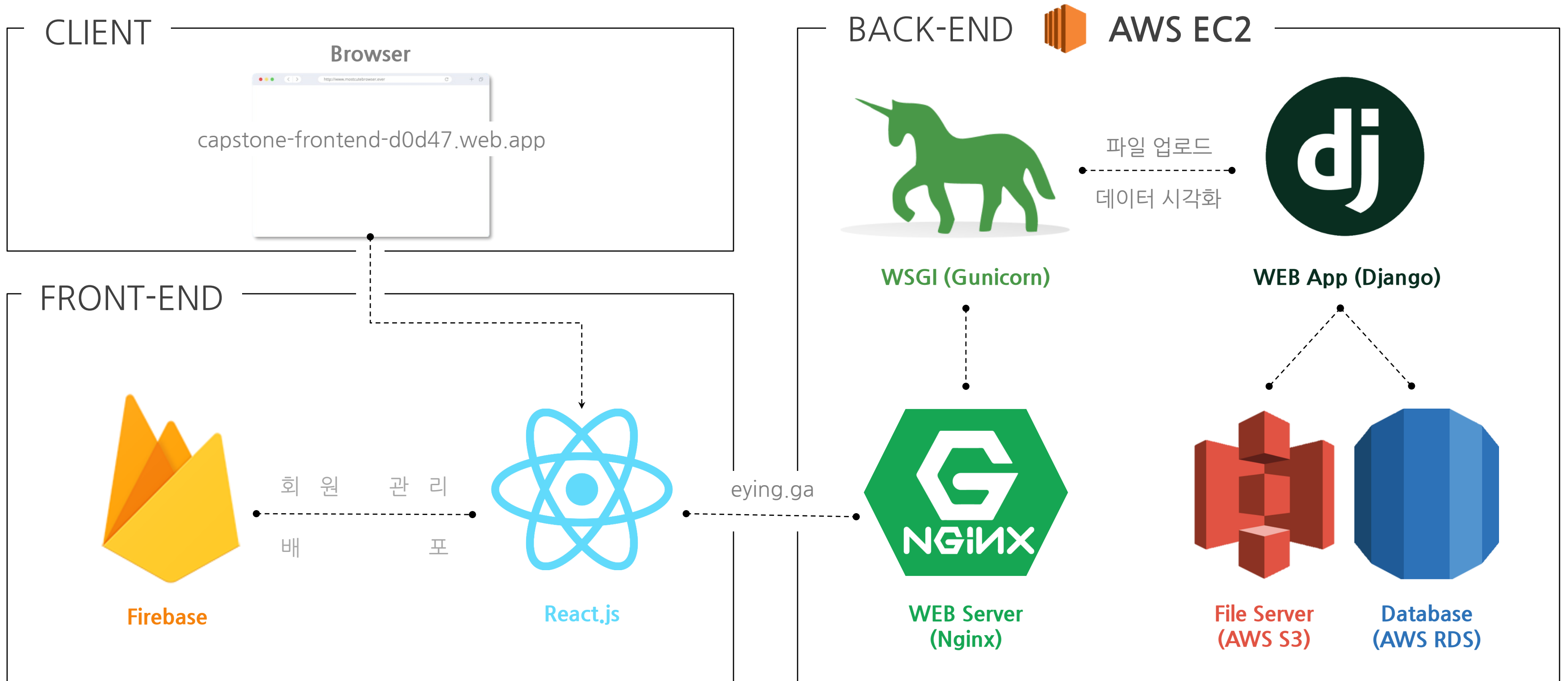
## 프로젝트 구조

---

### 시스템 구조

시스템은 어떻게 구성되었나요?

# 프로젝트 구조



# 03

---

## 시선 추적 (아이트래킹)

어떻게 사용자 시선을 추적하나요?

---

## 데이터 시각화

데이터를 어떻게 보여줄 건가요?

## 핵심 기능

# 핵심 기능



## 시선 추적

Webcam 시선 추적 모듈

정확도 차이

정확도 향상 방법

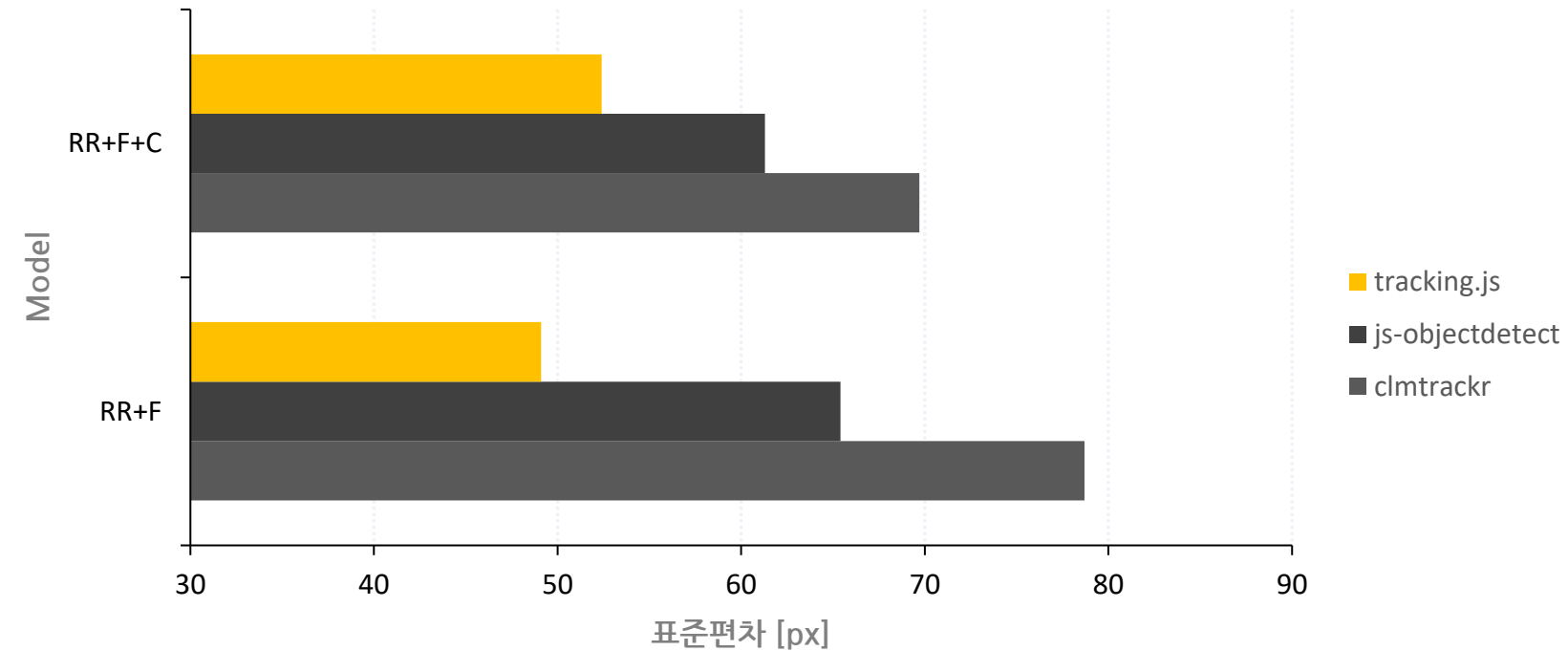


## 데이터 시각화

시선 분포

시각 흐름

# 핵심 기능



## 적용 모델

Javascript 얼굴 인식 라이브러리 중  
가장 표준편차가 적은 **tracking.js** 를  
WebGazer.js 에 적용

WebGazer.js 에서 선형(Linear) 회귀 모델보다  
**능형(Ridge) 회귀 모델**에서 높은 성능



# 핵심 기능

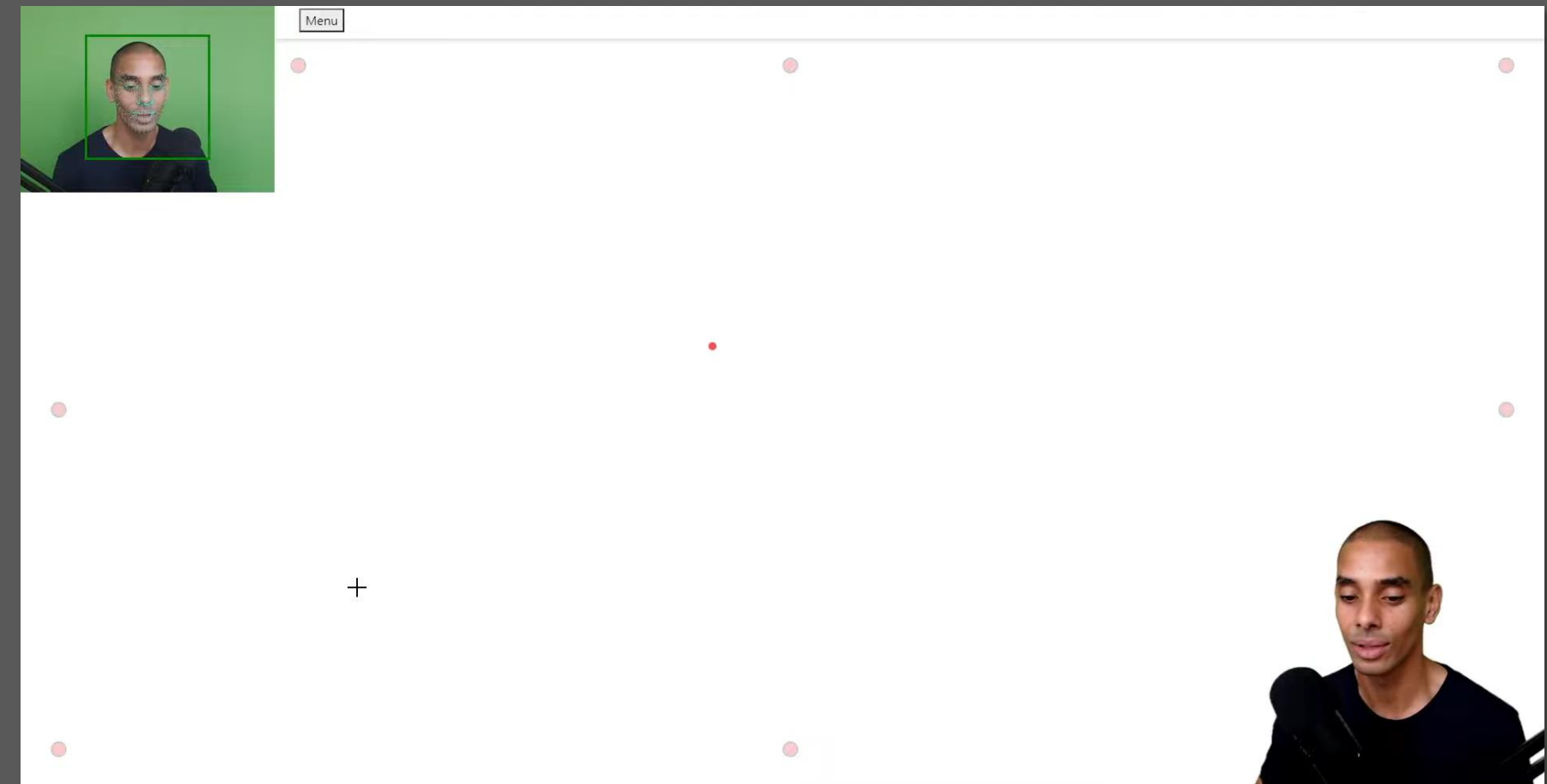
## 정확도 개선

### Calibration (교정)

사용자 별 눈의 특성을 측정하고  
3D 눈 모델과 함께 시선 데이터를 계산

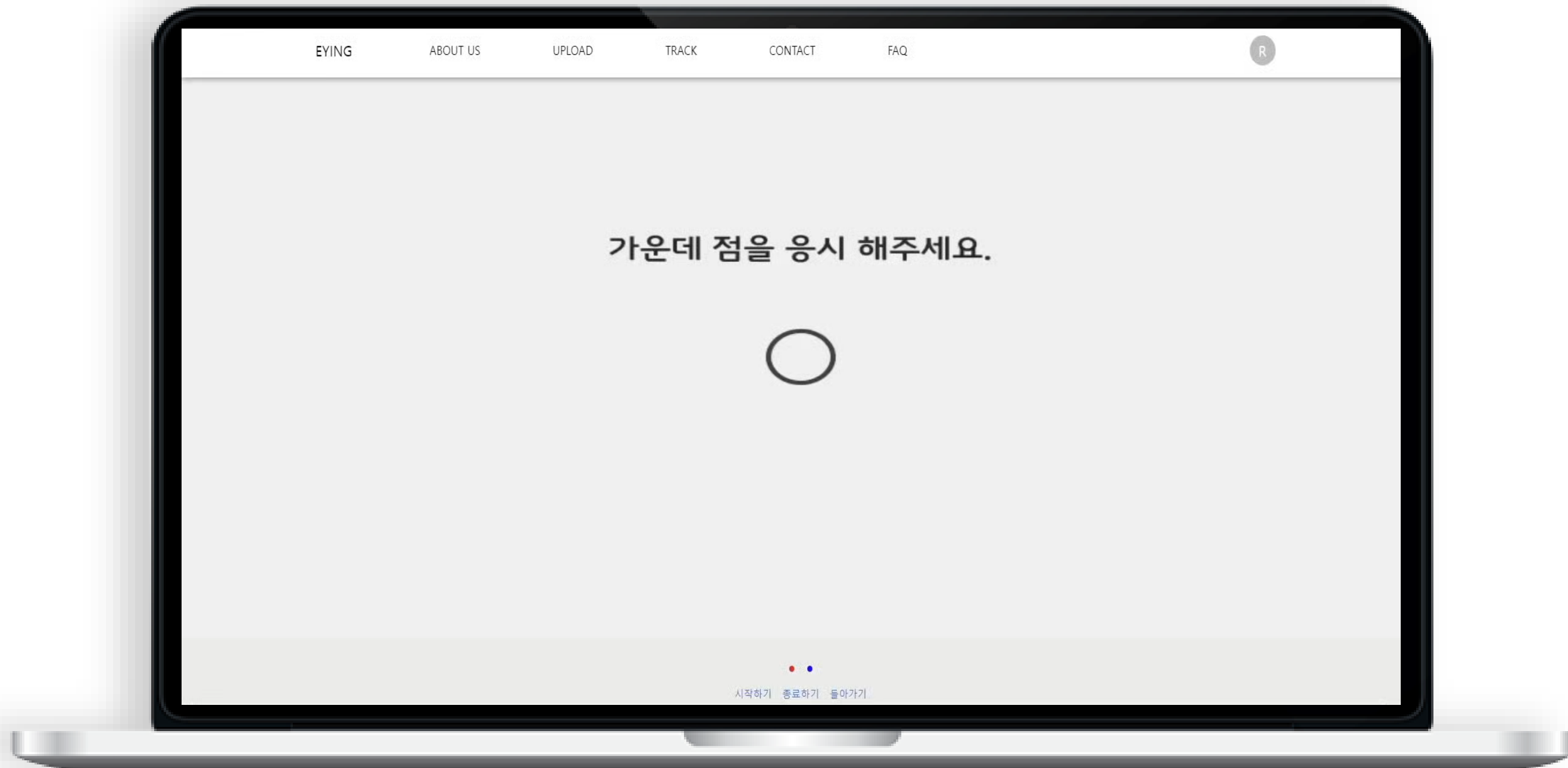
### Kalman Filter

잡음이 포함된 관측치를 바탕으로 상태 추정 및 제어



# 핵심 기능

시 선 추 적 정 확 도 영 상



# 핵심 기능



## 시선 추적

Webcam 시선 추적 모듈  
각 모델 성능 차이  
정확도 향상 방법



## 데이터 시각화

시선 분포  
시각 흐름

# 핵심 기능

## 응용 기술

LumenResearch/  
**heatmappy**

Draw image and video heatmaps in python



*heatmappy*

데이터 밀집도에 따라 Heatmapping



많음

적음

## 구현 화면

I N P U T

좌표

$[[x1, y1], [x2, y2], \dots]$

이미지

PNG, JPEG, ...



# 핵심 기능

응용 기술



*Pillow*

데이터 순서에 따라 Line Drawing



시작

끝

구현 화면

I N P U T

좌표

$[[x1, y1], [x2, y2], \dots]$

이미지

PNG, JPEG, ...





# 04

## 서비스 소개

---

웹 사이트

실제 구현된 서비스가 궁금해요

CAPSTONE 2022

서 비 스 소 개


회 원 가 입  
로 그 인

Email-Address	
Password	
Repeat Password	
User Name	
Age	
Gender ▼	
Job Field ▼	
Job ▼	Position ▼
명함 업로드	

회 원 가 입 페 이 지

평가자 데이터 제공을 위한 항목

나이 / 성별 / 분야 / 직무 / 직책 / 명함

 Sign In
Email Address
Password
로그인
Continue With Google
회원가입

로 그 인 페 이 지

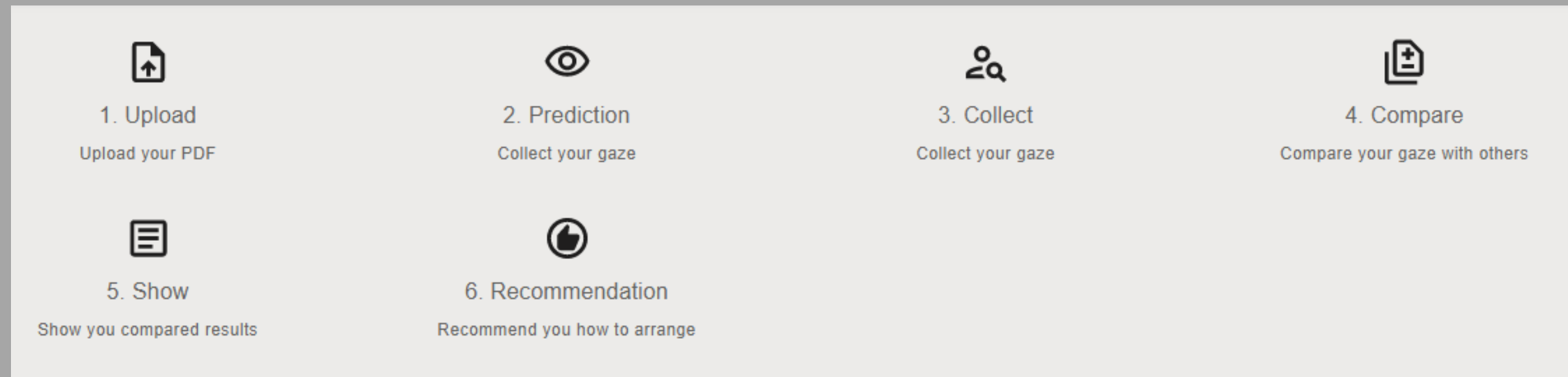
간편 로그인 가능

사설 이메일 또는 구글 이메일

CAPSTONE 2022

# 서비스 소개

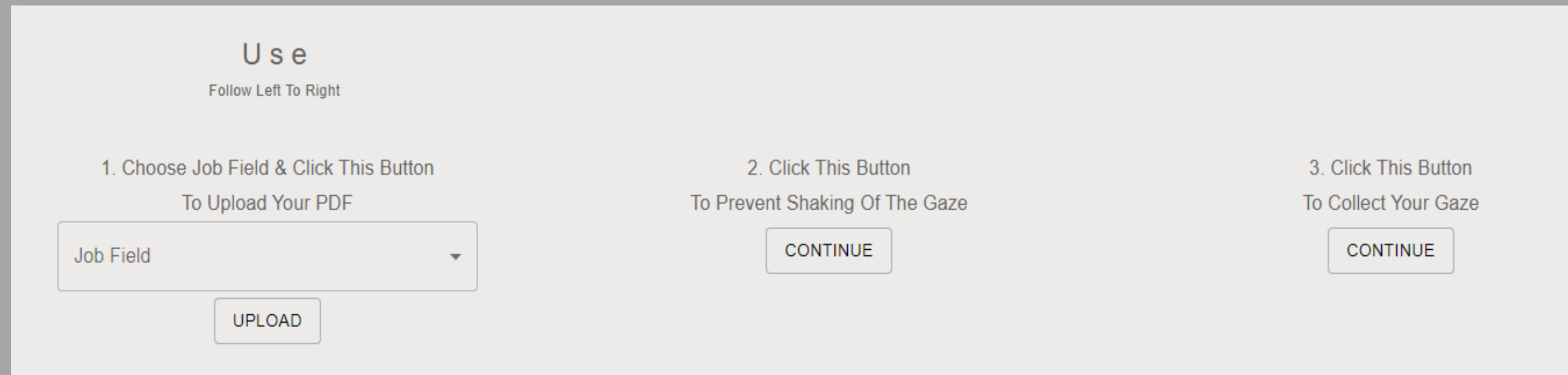
# U P L O A D



가이드라인

## 전체적인 서비스 사용법 설명

업로드 ~ 평가 데이터 조회



업로드 섹션

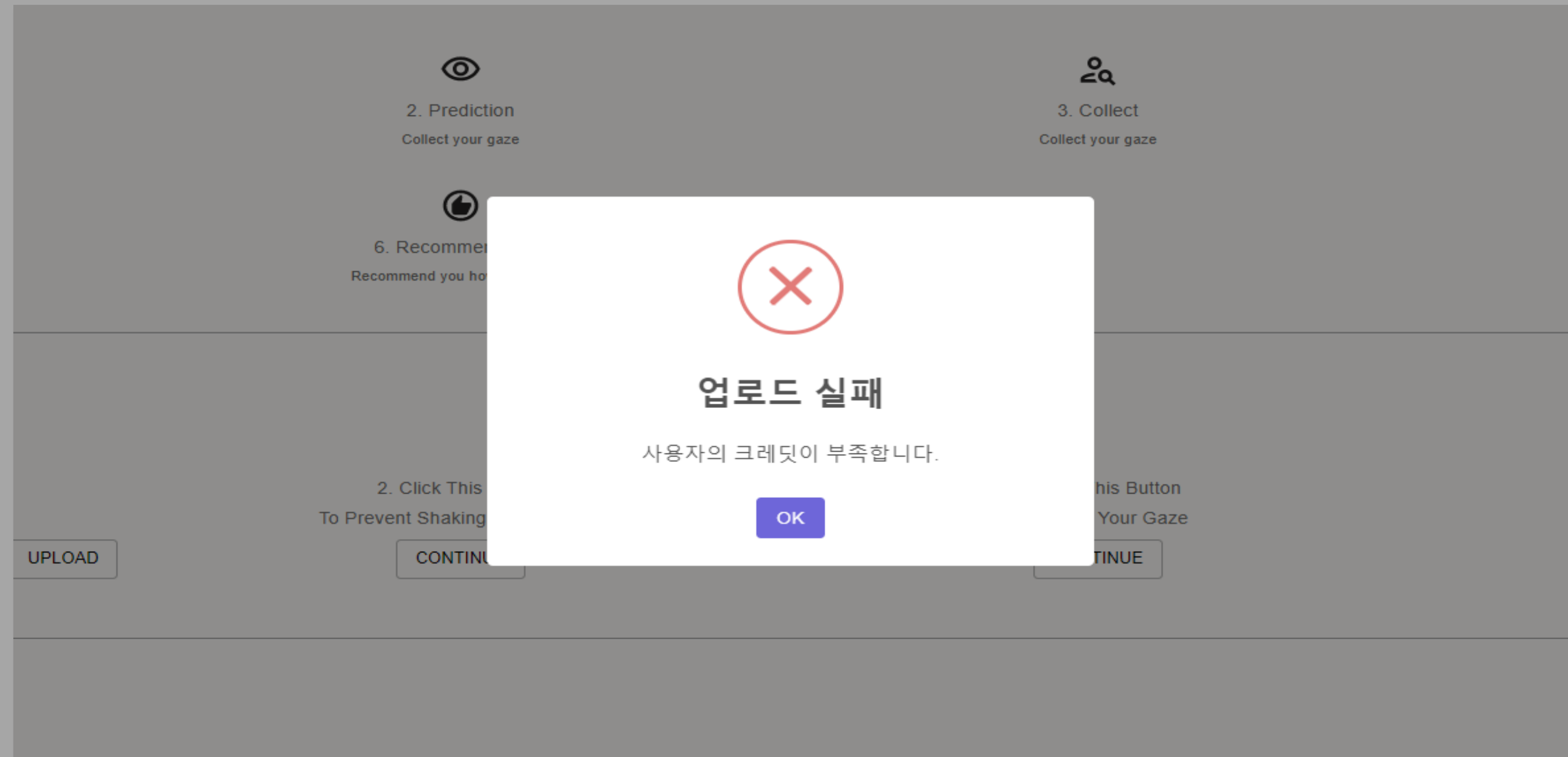
## 업로드 - 시선 교정 - 시청

자료 사용 분야 활용

CAPSTONE 2022

# 서비스 소개

# U P L O A D



크레딧 부족 화면

UPLOAD를 위한 크레딧 부족  
가입 시 크레딧 지급 또는 TRACK 리워드로 수급

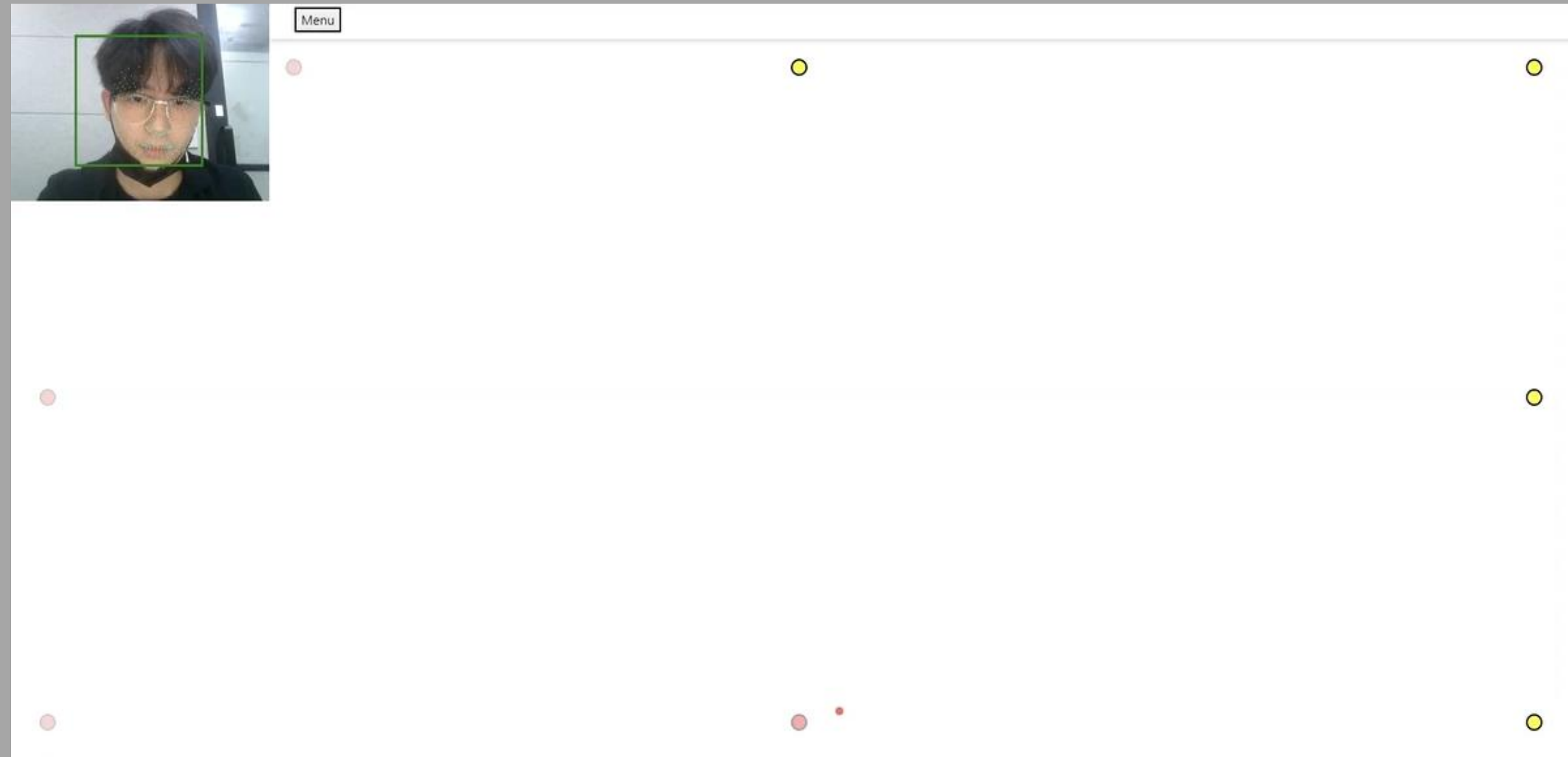
CAPSTONE 2022

서비스 소개

CALIBRATION

교

정



캘리브레이션

각 점을 응시하며 노란색이 될 때까지 클릭  
가운데 점을 마지막으로 정확도 측정이 끝나면  
시선 교정 완료



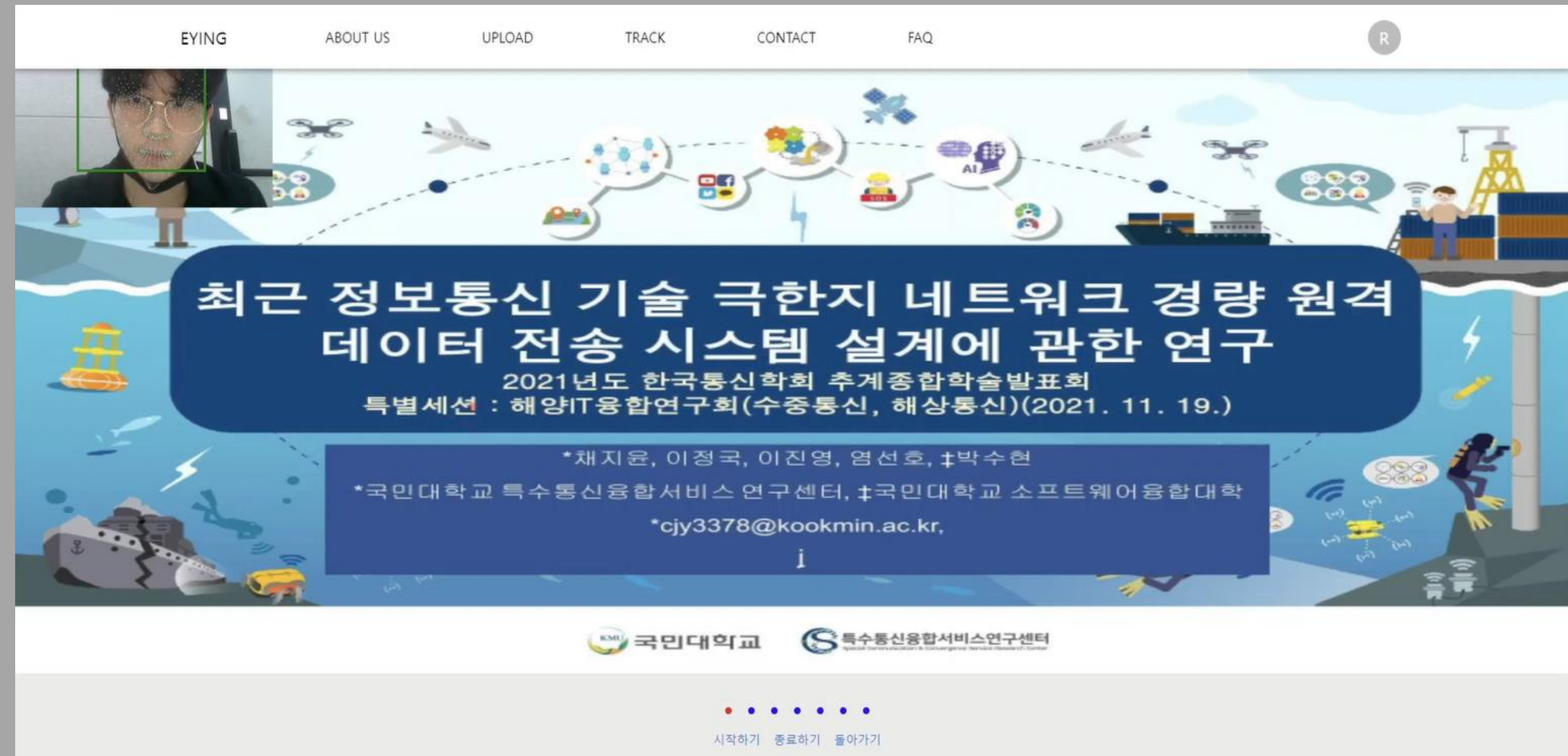


CAPSTONE 2022

# 서비스 소개

# TRACK

# 시선추적



EYE TRACKING

특정 자료 시청 및 시선 추적

카메라 권한 허용 必

## 서비스 소개

## TRACK

## 시선추적

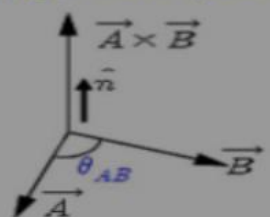
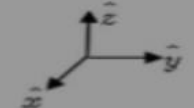
EYING
ABOUT US
UPLOAD
TRACK
CONTACT
FAQ
R

이차원에서의 운동

● 벡터 곱 ( Vector Product, Cross Product, Outer Product

• 두 벡터 A와 B의 벡터 곱
$$\mathbf{A} \times \mathbf{B} \equiv AB \sin \theta \hat{n} \quad (\text{정의})$$

벡터곱(vector product, cross product) :  $\vec{A} \times \vec{B} = AB \sin \theta_{AB} \hat{n}$ ,  $\hat{n} \cdot \vec{A} = 0$  &  $\hat{n} \cdot \vec{B} = 0$

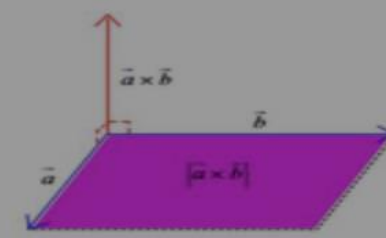
따라서,  $\vec{A} = A_x \hat{x} + A_y \hat{y} + A_z \hat{z}$ , 그리고  $\vec{B} = B_x \hat{x} + B_y \hat{y} + B_z \hat{z}$   

$$\vec{A} \times \vec{B} = (A_x \hat{x} + A_y \hat{y} + A_z \hat{z}) \times (B_x \hat{x} + B_y \hat{y} + B_z \hat{z})$$


$$\vec{A} \times \vec{B} = A_x B_x \hat{x} \times \hat{x} + A_x B_y \hat{x} \times \hat{y} + A_x B_z \hat{x} \times \hat{z} + A_y B_x \hat{y} \times \hat{x} + A_y B_y \hat{y} \times \hat{y} + A_y B_z \hat{y} \times \hat{z} + A_z B_x \hat{z} \times \hat{x} + A_z B_y \hat{z} \times \hat{y} + A_z B_z \hat{z} \times \hat{z}$$

$$\vec{A} \times \vec{B} = A_x B_y \hat{z} - A_x B_z \hat{y} - A_y B_x \hat{z} + A_y B_z \hat{x} + A_z B_x \hat{y} - A_z B_y \hat{x}$$

$$\vec{A} \times \vec{B} = (A_y B_z - A_z B_y) \hat{x} + (A_z B_x - A_x B_z) \hat{y} + (A_x B_y - A_y B_x) \hat{z} \text{로 벡터가 된다.}$$



보기: 두 벡터가 이루는 평행사변형의 넓이)



방향: 두 벡터가 포함된 평면에 수직, 오른손 법칙

$$\mathbf{A} \times \mathbf{B} = -\mathbf{B} \times \mathbf{A} \quad (\text{반교환법칙})$$

$$\mathbf{A} \times (\mathbf{B} + \mathbf{C}) = \mathbf{A} \times \mathbf{B} + \mathbf{A} \times \mathbf{C} \quad (\text{배분법칙})$$

15

시작하기

종료하기

돌아가기

평가 리워드 지급 메시지

다른 사용자의 자료를 평가하는 사용자에게 리워드 지급


업로드 크레딧 50% 수준


CAPSTONE 2022

## 서비스 소개

## MY PAGE

## 사용자 정보

이메일 : capstone@capstone.cap
이름 : 아잉
명함
 Need Technical Support Contact anytime
나이 : 25
성별 : Male
분야 : IT
직업 : 대학생
계급 : 4학년
PDF 업로드 수 : 4
포인트 : 50
포인트 구매

이메일	capstone@capstone.cap	Email	변경하기
변경할 이메일 주소를 입력해주세요			
이름	아잉	Name	변경하기
변경할 이름을 입력해주세요			
분야	IT	Job Field	변경하기
직업	대학생	Job	
계급	4학년	Position	변경하기
직업을 먼저 선택하세요			
 Need Technical Support Contact anytime		새 명함 업로드	
비밀번호 재설정			

내 정보 조회 및 변경

사용자 정보 조회 및 변경

PDF 업로드 수 / 포인트

# 서비스 소개

# MY PAGE

LH P D F

내 PDF 목록 조회

## 사용자가 업로드 한 자료 목록 조회

• **등록일 / 평가 마감일 / 조회수 정보 제공**



# 서비스 소개

# MY PAGE

# 시청자

# 시청 사용자 데이터 및 시선 데이터 시각화

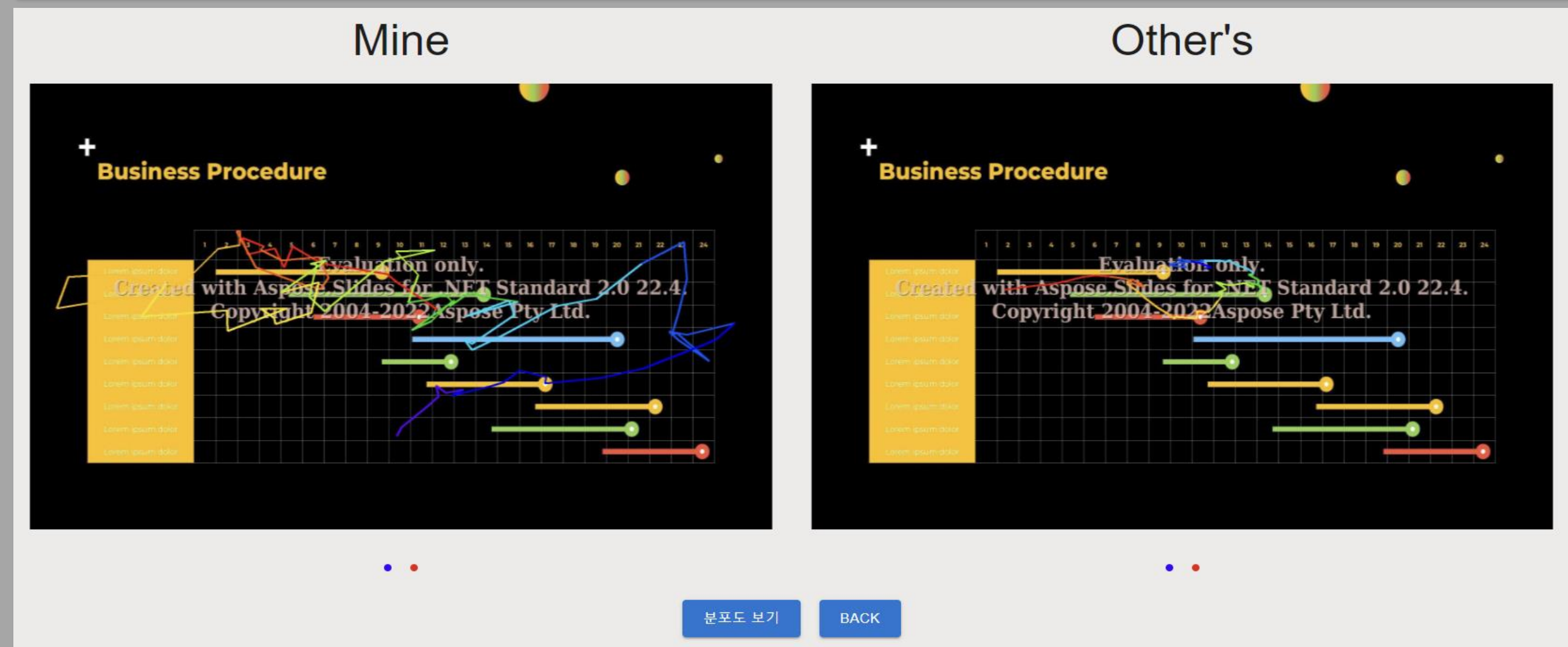
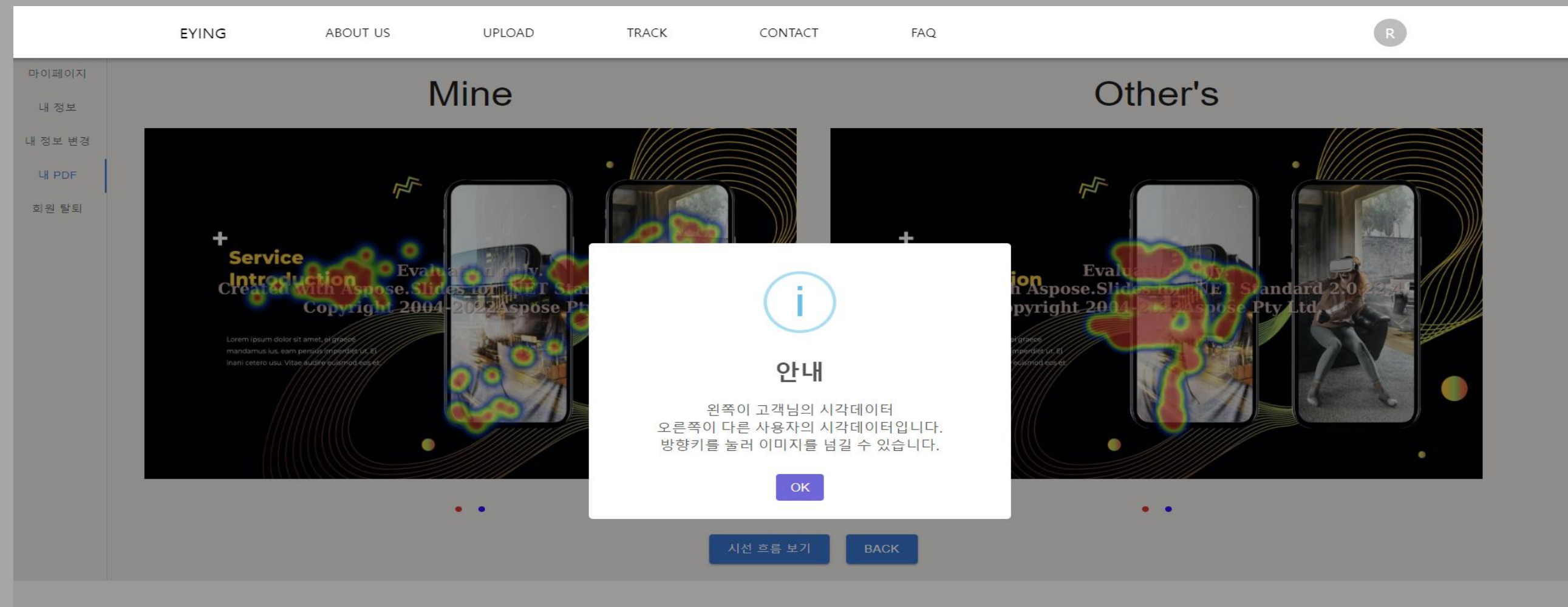
분야 / 직무 / 직책 / 성별 / 날짜 별 필터링 가능

CAPSTONE 2022

# 서비스 소개

# MY PAGE

# 시각화



05

시

연

---

시연 동영상

서비스 소개 영상

# 시 연

서 비 스 실 제 시 연 영 상





T H A N K S

CAPSTONE 2022-44

06

보 조 자 료

---

표준 포맷

왜 PDF를 사용하나요?

---

WebGazer 모델

사용 모델 성능표

## 보 조 자 료

# Why PDF?

PPTX 파일도 지원

무료 라이선스 워터마크 제거 불가

## 표 준 형 식 보 안 이 슈

ISO 32000 표준 형식

장치 및 해상도 독립성

실행 환경 제한 無

관련 모듈 보안 이슈

원격 코드 실행 취약



보 조 자 료

Model	Quiz/ Cimtrackr		Target/ Cimtrackr		Quiz/ Js-object.		Target/ Js-object.		Quiz/ Tracking.js		Target/ Tracking.js		All Tasks/ Libraries	
	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
Linear	245.4	82.5	227.9	37.5	271.2	82.8	230.0	36.1	311.9	112.2	260.7	29.6	256.9	75.0
RR	207.6	87.5	165.8	71.7	248.5	80.8	197.3	58.8	309.8	111.7	275.1	47.1	232.4	92.3
RR+F	247.4	91.3	207.4	78.7	257.8	87.1	218.6	65.4	308.1	111.9	275.7	49.1	251.5	89.0
RR+C	118.7	55.4	104.5	55.1	167.3	67.8	160.7	54.6	258.7	115.8	251.8	55.8	174.9	91.6
RR+F+C	180.8	75.5	157.0	69.7	208.7	82.5	206.1	61.3	263.0	114.2	255.4	52.4	210.6	86.3

Table 1: Mean (*M*) and standard deviation (*SD*) of prediction errors measured in pixels across all regression models and combinations of tasks and libraries. The mean is obtained from averaging across the 76 online participants.

Simple Linear vs. Ridge Regression

First, we compare the accuracy of the simple linear regression model that maps the pupil location to display coordinates against the ridge regression model that maps the 120D eye feature vector. Across all clicks, the mean error from the linear regression model is 256.9 pixels (*SD* = 75.0). Similarly, the mean distance between the location of a click and the prediction made by the ridge regression model is 233.4 pixels (*SD* = 92.3). The means and standard deviations of each combination of task type and eye detection library are reported in Table 1.

We average the error across all clicks for each participant. A Mann-Whitney U test showed that the mean error was greater for the simple linear regression ( $p < 0.005$ ).

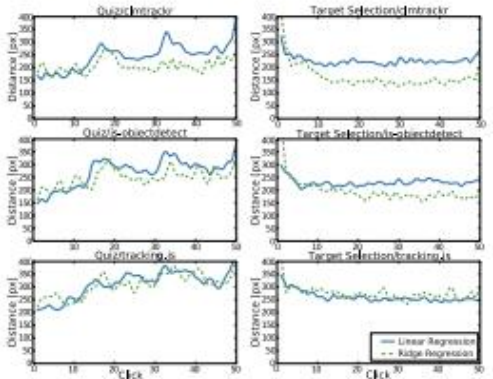


Figure 2: Average Euclidean distance in pixels between the click location and the predictions made by the simple linear (solid blue) and the ridge regression model (dashed green).

Figure 2 shows the average Euclidean distance in pixels across all 50 normalized clicks. We observe different error trends across the two task types. Filling the quiz seems to introduce more error with more clicks, perhaps because users need to scroll to reach all questions and thus they move more. On the other hand, when selecting targets, the error drops until it stabilizes—no scrolling happens in this type of task.

As ridge regression has generally a lower error, we base our subsequent analysis only on the ridge regression model that matches eye feature vectors to display coordinates.

Comparison of All Ridge Regression Models

We compare the accuracy of all prediction models that use ridge regression: the simple ridge regression (RR), the regression when adding extra samples within the fixation buffer (RR+F), when sampling cursor activity outside of clicks (RR+C), and when combining all the above (RR+F+C). Figure 3 shows the average Euclidean distance in pixels across all clicks for all combinations of tasks, libraries, and regression models. Again we observe the same upward trend for the quiz task across all prediction models. On the other hand, for the target selection task, we notice that for the cimtrackr and js-objectdetect detection libraries, the error decreases during the first half of the task and increases during the second half. Performing the study online leaves room for the interpretation of the observed variability. The speed of the external libraries can have a significant effect on WebGazer’s ability to match correctly frames and locations on screen. Head movement can also affect the detected images that correspond to eye patches.

Note that WebGazer achieves a significant increase in the accuracy of the predicted gaze. Sampling cursor activity (RR+C) has the smallest average error of 174.9 pixels, followed by the model that combines fixations and cursor activity (RR+F+C) with an error of 210.6 pixels. Contrary to our expectations, the error of WebGazer using extra samples within a fixation buffer (RR+F) increased ( $M = 251.5$  pixels). There are a couple of factors that could have an effect, e.g. blinks within the fixation buffer or temporal and spatial ranges being too lenient. The number of extra samples within a fixation buffer also depended on the performance of the eye detection library in conjunction with the increased cost of extra training data.

5.2 In-Person Study Results

The in-person user study data was captured via log files from the Tobii EyeX eye tracker and server logs for WebGazer. Both datasets were parsed down to a time series of predictions and grouped into 10 millisecond bins. The error was computed as the average distance between each regression and Tobii EyeX for the equivalent bin. Tracking.js data were excluded due to performance issues running both eye trackers simultaneously.

Table 2 contains the mean prediction errors and standard deviations measured in pixels for all 4 participants. We note that the mean errors reported here are comparable with the ones in Table 1 from the online study. The errors are naturally higher as the population size differs significantly. The mean errors are computed even when the user is not clicking. That further supports our assumption for matching gaze and click coordinates.