DOAA CA2

Emotive

Name: Muhammad Faqih Akmal **&** Oh Tien Cheng

ID: P2012030 **&** P2012072

Class: DAAA/FT/2B/01

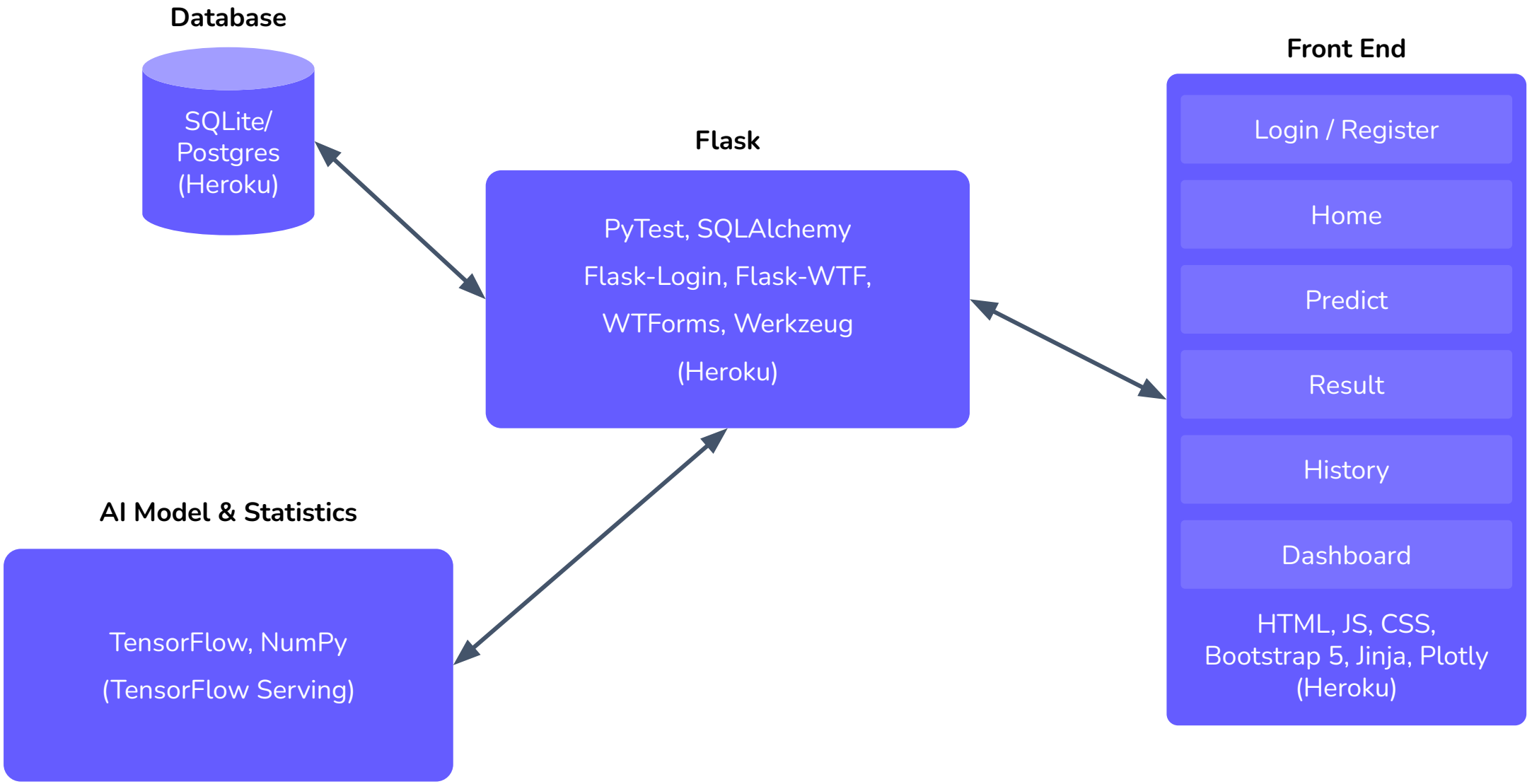
Agenda

- **Introduction**
- **System Architecture & Libraries**
- **Pages & Features**
  - Login/Register
  - Home
  - Predict
  - Result
  - History
  - Dashboard
- **APIs**
- **Model**
- **Testing**

Introduction

- **Purpose of Project**
  - To create a Flask Web Application that uses a Deep Learning model to classify images fed from a Web Camera
  - To store the images and predictions based on user
- **Scope of Project**
  - Face emotion detection using a Camera

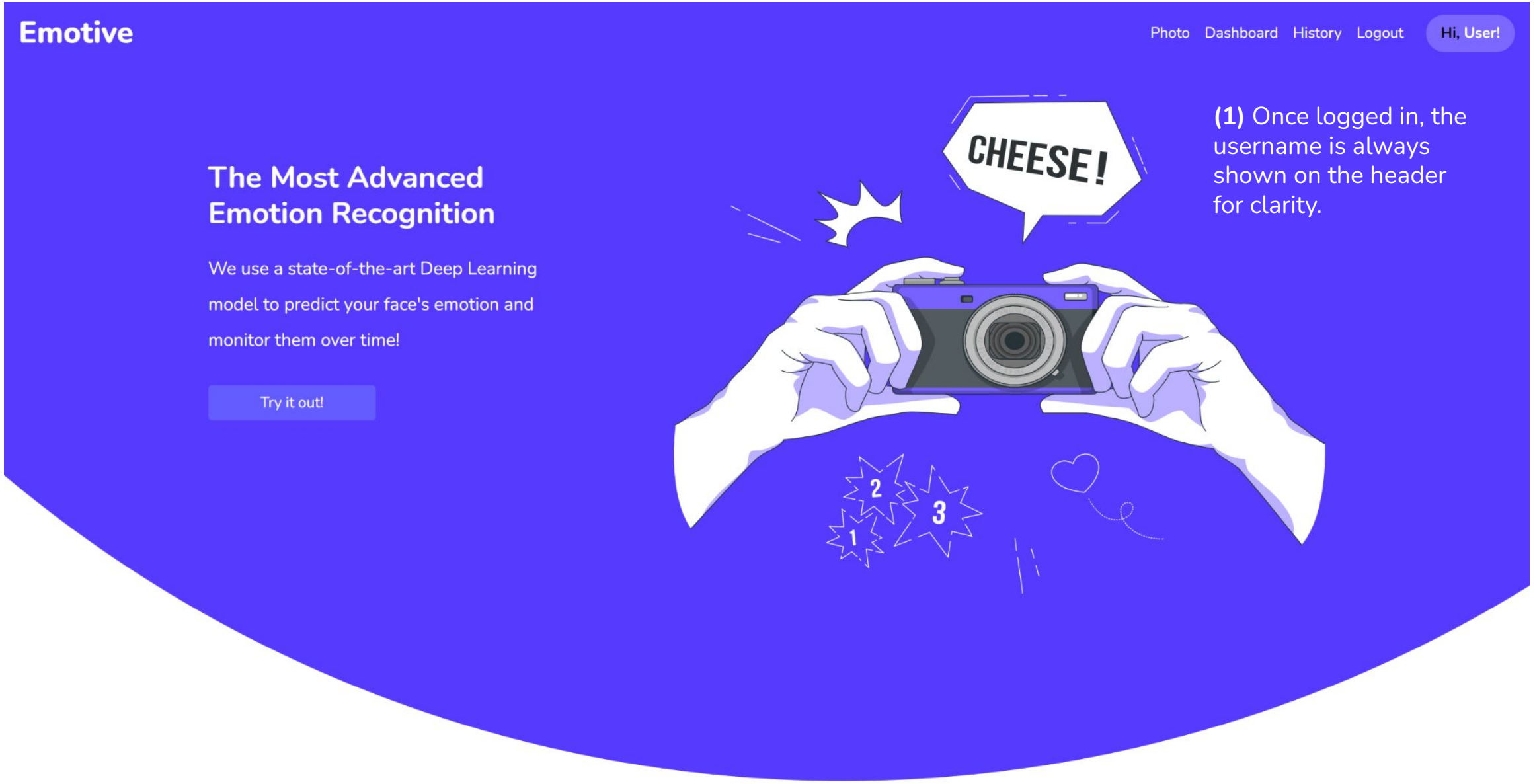
Architecture & Libraries



General Features

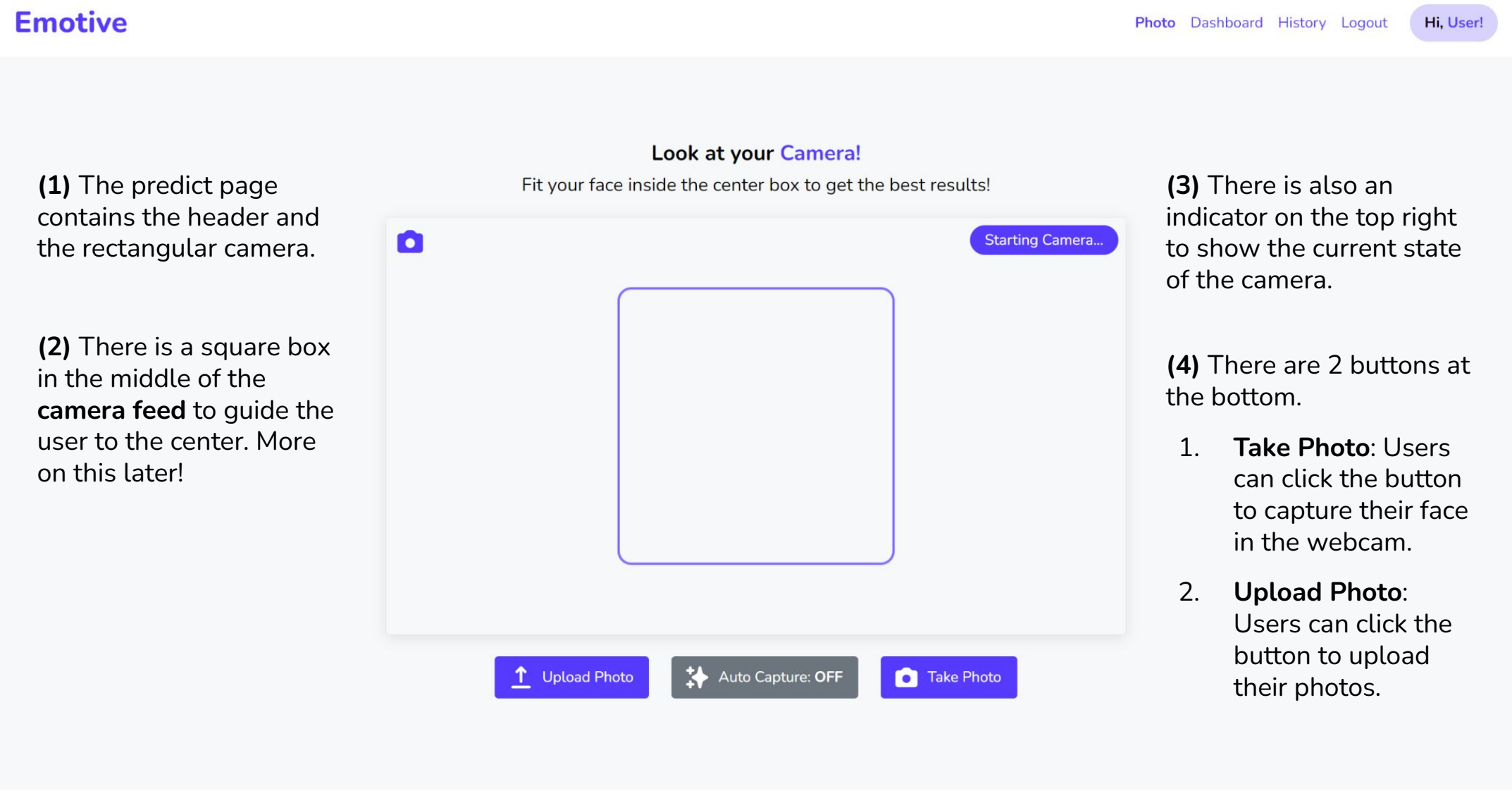
- **Authentication**
  - All pages except the home page requires authentication
  - Predictions are stored based on User
  - Different user sees different prediction history
- **Mobile Friendly Interface**
  - All pages are viewable on mobile devices
  - These pages include Camera, Result, History & Dashboard

Home Page



(1) Once logged in, the username is always shown on the header for clarity.

Predict Page (1)



(1) The predict page contains the header and the rectangular camera.

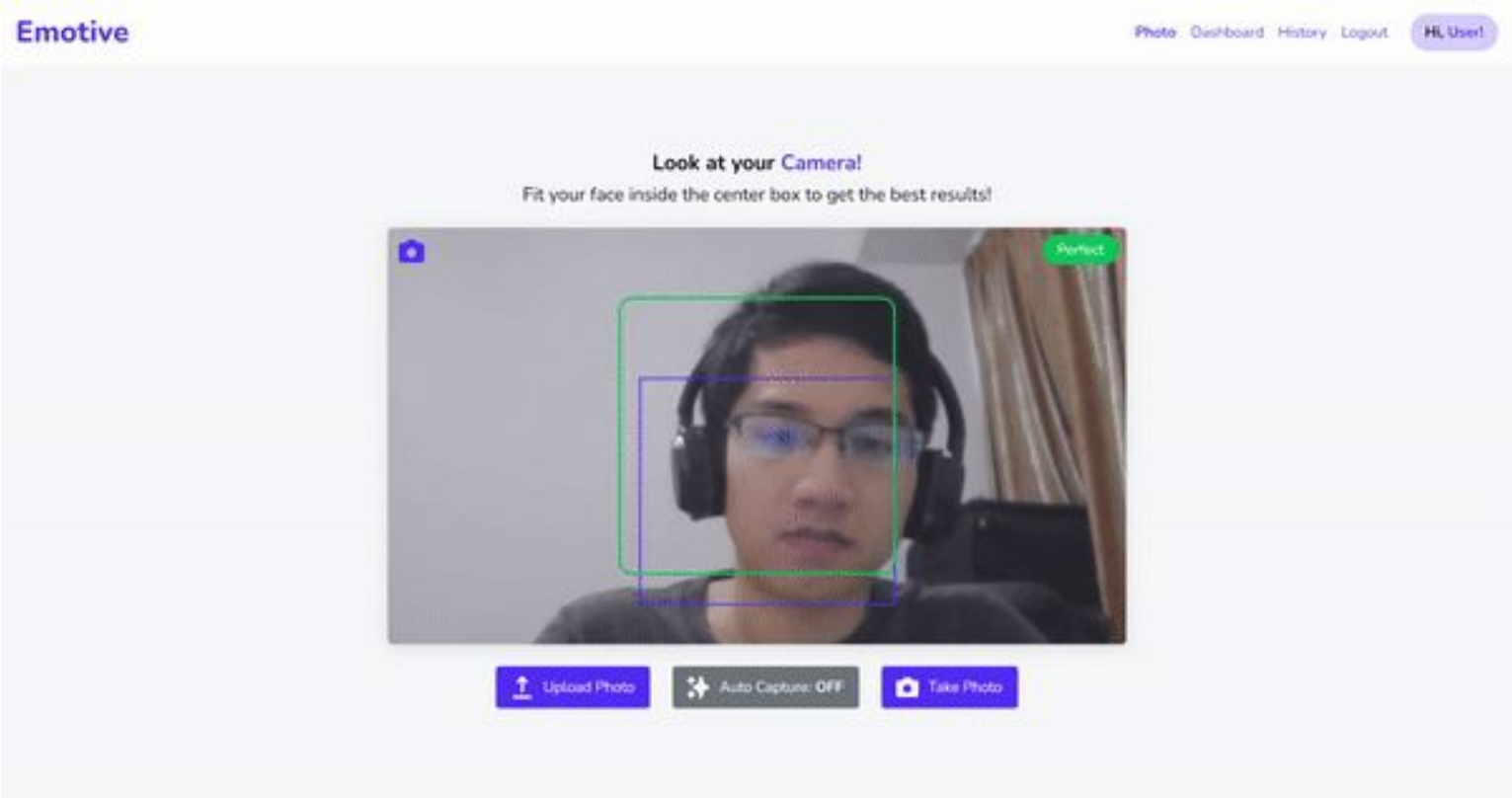
(2) There is a square box in the middle of the **camera feed** to guide the user to the center. More on this later!

(3) There is also an indicator on the top right to show the current state of the camera.

(4) There are 2 buttons at the bottom.

1. **Take Photo**: Users can click the button to capture their face in the webcam.
2. **Upload Photo**: Users can click the button to upload their photos.

Predict Page (2)



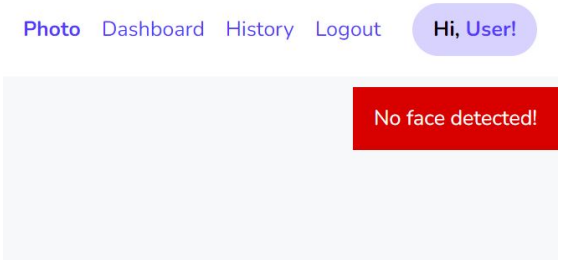
Face Tracking: If a face is detected, the user's face will be tracked with a purple box.

3 Live Camera States:

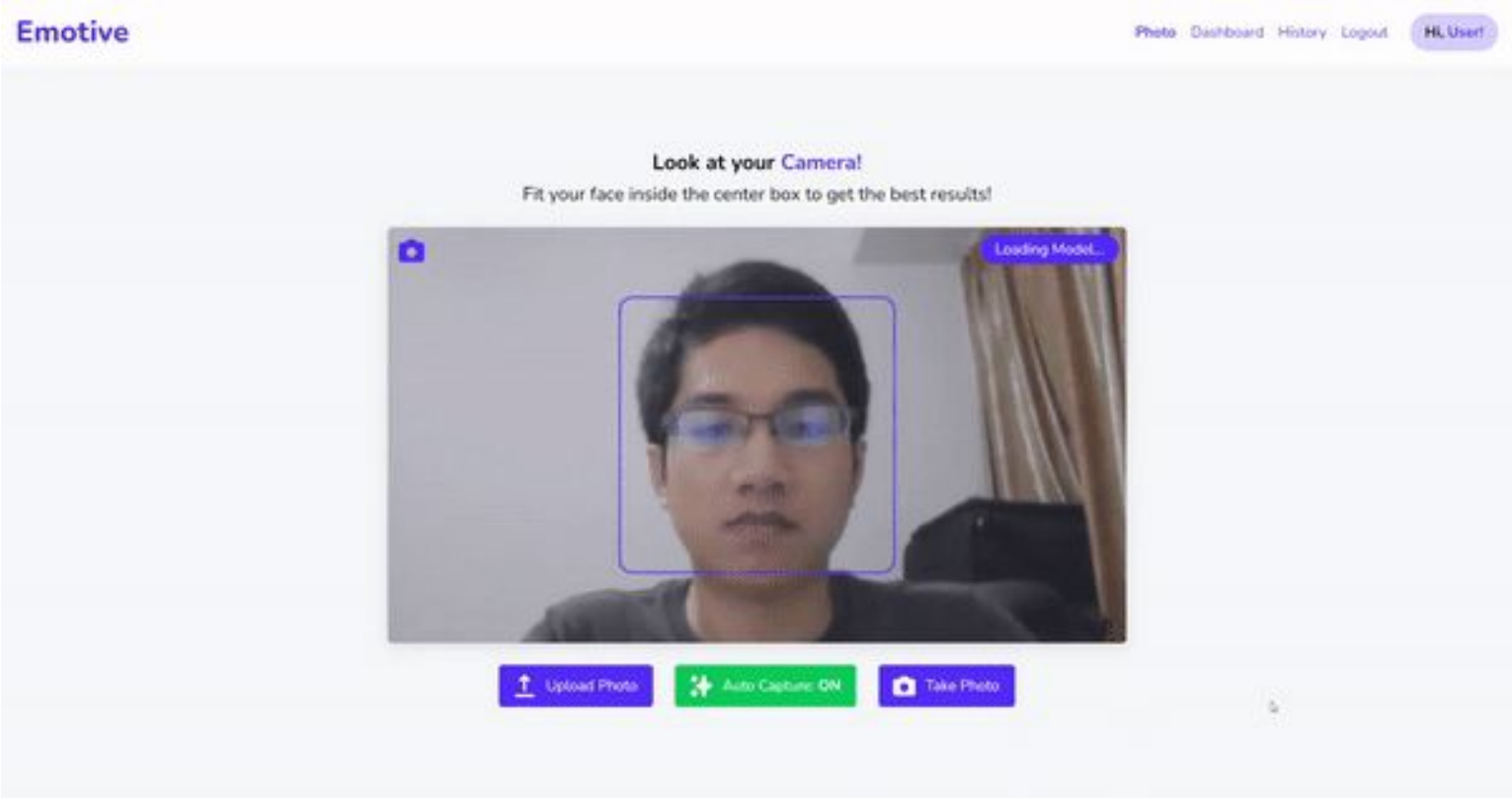
1. **Perfect** (Green)
   The user is perfectly centered
2. **Move to the box** (Orange)
   The user is not centered
3. **No face detected** (Red)
   No face is detected

This ensures the user's face is centred when the photo is taken. Even if the face is not centered, we have an additional model in the backend that crops the image to the user's face.

Once the image is sent, if no face is detected, the page will show a warning that "**No face is detected**!" before redirecting back.



Predict Page (3)



Auto Capture

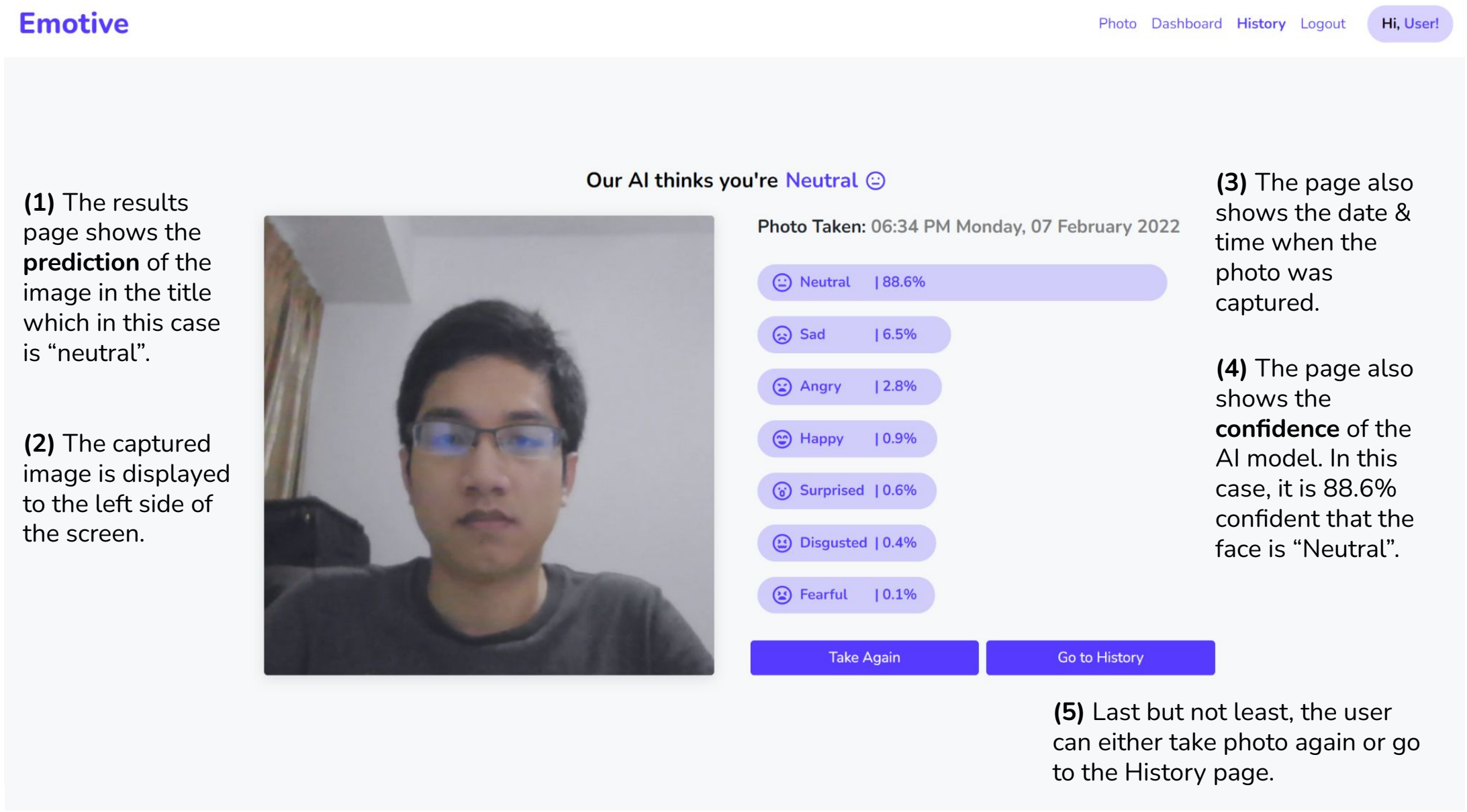
Auto Capture is a feature that is **enabled** by default and can be turned on or off in the Predict page as well as the Dashboard page.

If a user's face is centred for 3 seconds, then the app will **automatically capture** the image and move on to prediction.

Auto Capture ensures **effortless** photo taking without pressing any button.

The feature is **disabled** on mobile devices due to insufficient computing power compared to most laptops or PCs.

Results Page



(1) The results page shows the **prediction** of the image in the title which in this case is “neutral”.

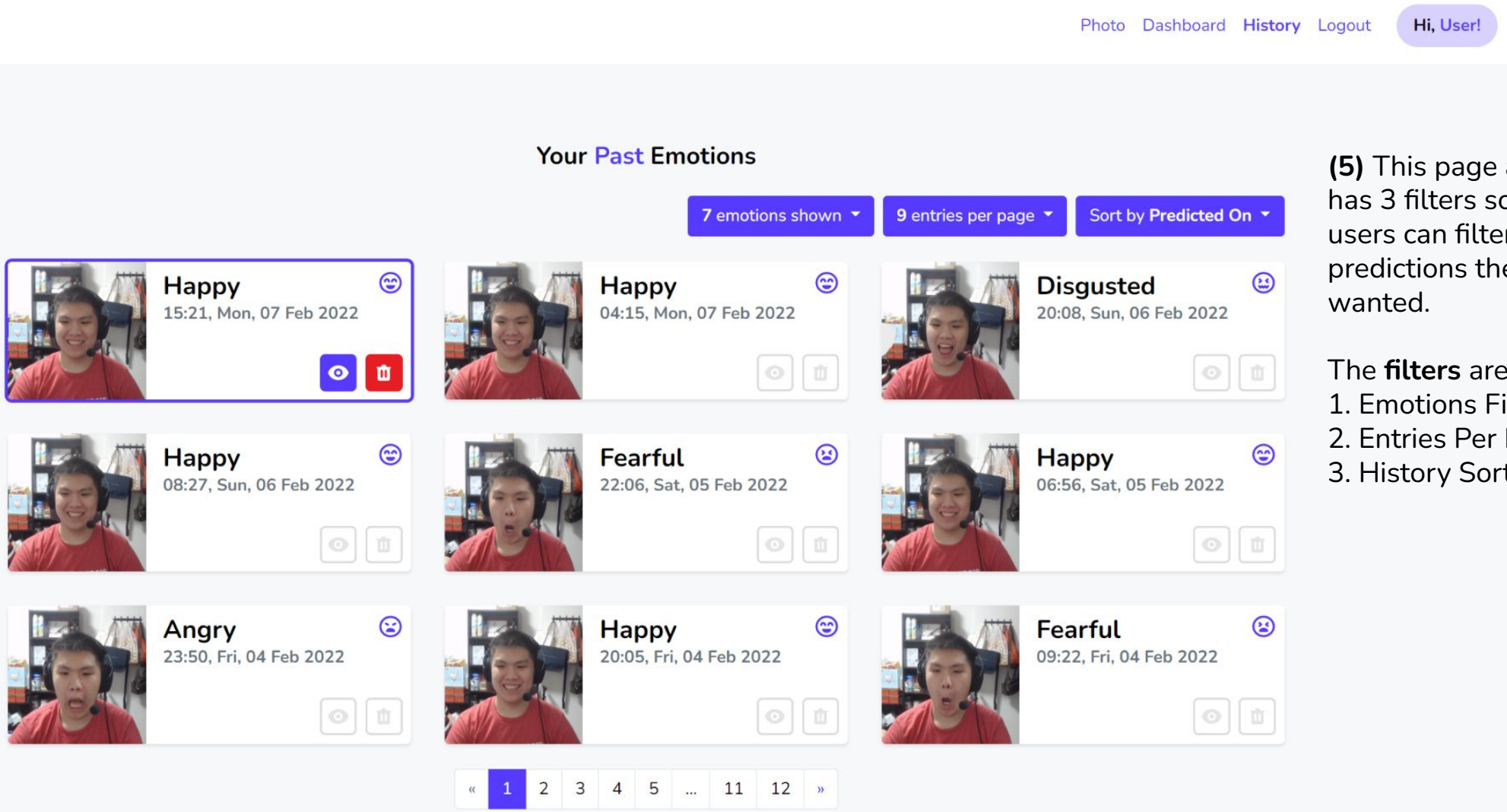
(2) The captured image is displayed to the left side of the screen.

(3) The page also shows the date & time when the photo was captured.

(4) The page also shows the **confidence** of the AI model. In this case, it is 88.6% confident that the face is “Neutral”.

(5) Last but not least, the user can either take photo again or go to the History page.

History Page



(1) All of the predictions of the user will be shown in this history page.

(2) Each card is a prediction that contains the photo, the predicted emotion and the date taken.

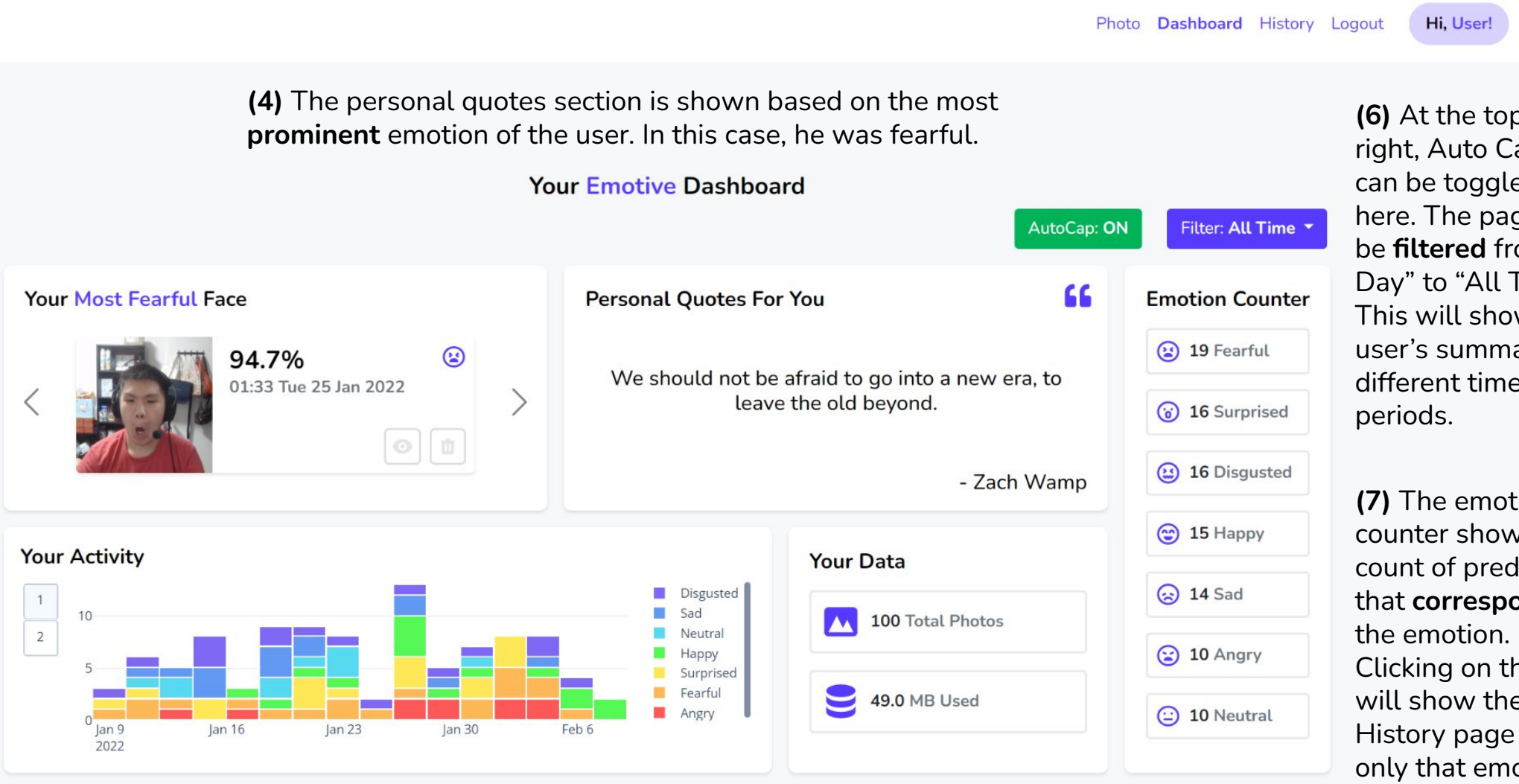
(3) Each card also has 2 buttons. One to **view** back the prediction & the other to **delete**.

(4) This page has a **pagination** feature so that users do not have to see their history all at once.

(5) This page also has 3 filters so that users can filter the predictions they wanted.

The **filters** are:
1. Emotions Filter
2. Entries Per Page
3. History Sorting

Dashboard Page (1)



(1) The dashboard is a **statistical and personal** summary of the user's usage on the web application.

(2) The top left section is a sliding **carousel** that shows the face with the highest confidence of the AI model for the particular emotion.

(3) The bottom left shows the user's **emotion** over a period of time. More about this in the next slide.

(4) The personal quotes section is shown based on the most **prominent** emotion of the user. In this case, he was fearful.

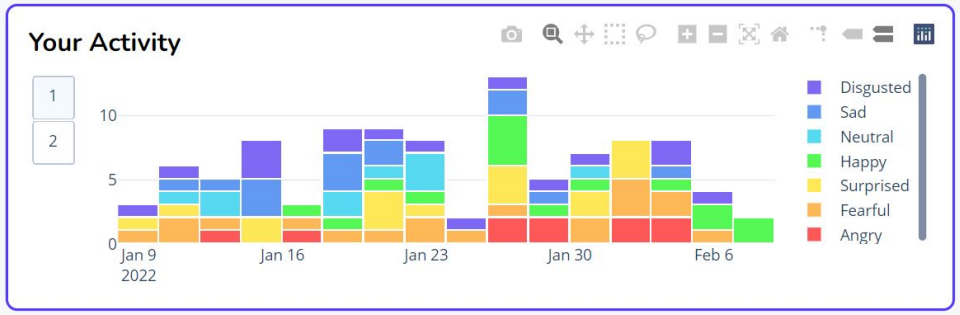
(5) "Your Data" section shows the total number of photos and the amount of **storage** used.

(6) At the top right, Auto Capture can be toggled here. The page can be **filtered** from "1 Day" to "All Time". This will show the user's summary for different time periods.

(7) The emotion counter shows the count of predictions that **corresponds** to the emotion. Clicking on them will show the History page with only that emotion.

Dashboard Page (2)

(1) Under “Your Activity” Section, there are 2 different graphs to choose from.



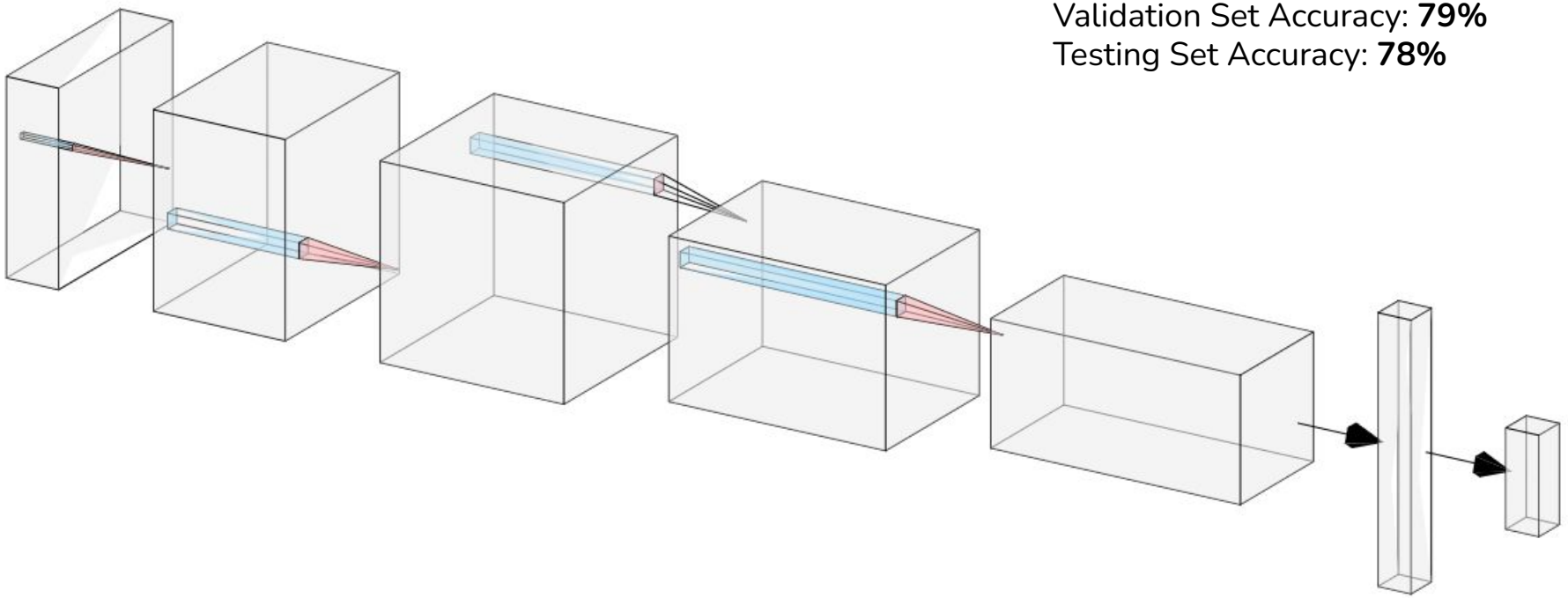
(2) The default graph is a histogram that shows the total number of **emotions predicted** for every period of time. They are separated by the different emotion types. Clicking on the legend on the right can even filter out or in the emotions of interest.

(3) The second graph can be accessed by clicking the “2” button on the left. It is a bar chart that shows the **net positive and negative emotions** over certain periods of time. For example, the above figure shows that the user has a net emotion of 0.9 around 22 Jan.

APIs

- **POST /api/predict:** Receives an image and returns the prediction
- **POST /api/history/add:** Adding new history to the database
- **GET /api/history/get/<history_id>:** Getting the history by its ID
- **GET /api/history/<user_id>:** Getting all history of a particular user (with pagination & filters)
- **DELETE /api/history/delete/<history_id>:** Delete a history with its ID
- **POST /api/user/add:** Register users to the database
- **GET /api/user/<user_id>:** Get the user's information
- **GET /api/user/all:** Get all users' information
- **DELETE /api/user/delete/<user_id>:** Delete user with his/her ID

About the Model



Validation Set Accuracy: **79%**
Testing Set Accuracy: **78%**

Model Name: Wide Residual Network
Number of Parameters: 3.3 Million

(1) Trained from scratch using 28K labelled images
(2) Validated on 3.5K images
(3) Tested on 3.5K images.

	precision	recall	f1-score	support
Anger	0.66	0.76	0.71	335
Disgust	0.00	0.00	0.00	23
Fear	0.49	0.44	0.47	90
Happiness	0.93	0.88	0.90	921
Neutral	0.76	0.90	0.82	1277
Sadness	0.68	0.50	0.58	439
Surprise	0.89	0.68	0.77	459
accuracy			0.79	3544
macro avg	0.63	0.60	0.61	3544
weighted avg	0.79	0.79	0.78	3544

Testing

Testing Summary

- **Total Number of Tests:** > 1600
- **Types of Tests:**
  - Validity Testing
  - Range Testing
  - Consistency Testing
  - Expected Failure Testing
  - Unexpected Failure Testing

Validity Testing

Test that the application works as intended on real data

Example:

- Test on real data and check that output is correct
- Test that model results are added to user history

Range Testing

Inputs or data that are not in the range of accepted values should be rejected.

Example:

- Ensure that predictions probabilities are between 0 and 1
- Ensure that the date stored not before the creation of the application

Consistency Testing

Application should respond identically for the same inputs or data sent.

Example:

- Ensure probabilities for each class is the same when same data is sent multiple times
- Ensure that non-faces are rejected by the model as input

Expected Failure Testing

Send invalid data where the application is anticipated to return an error.

Example:

- Ensure that when sending a non image file format, the app returns an error
- Ensure app returns error when sending images with no face

Unexpected Testing

Application ability to handle unexpected behaviour (e.g. corrupted data, broken disk)

Example:

- Ensure app returns appropriate error on accessing non-existent or unauthorized endpoints
- Ensure app returns appropriate error on receiving corrupted data

DOAA CA2

Thank You

Project Name: Emotive