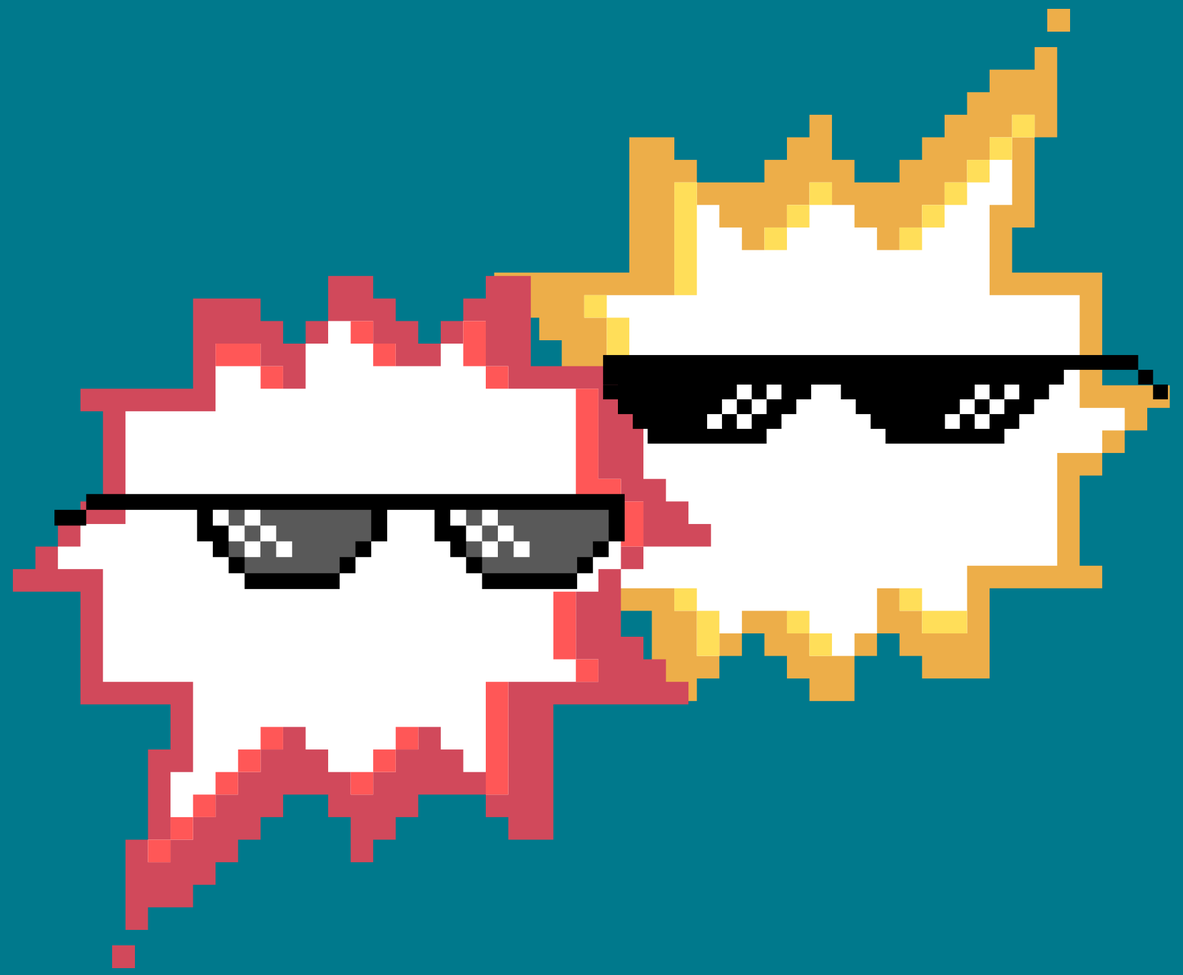


TALKTAGGER

Can you guess who said that?

B. Nil Atabey

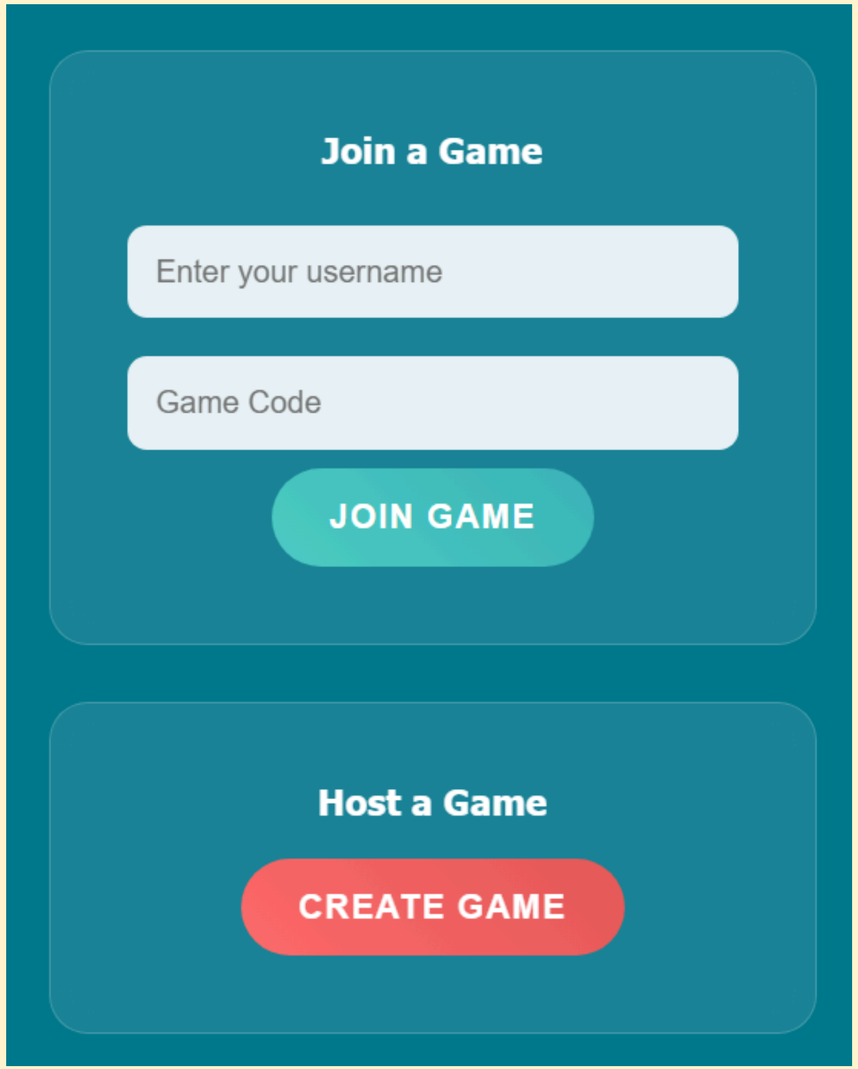
AI Lab: Computer Vision and NLP
Applied Computer Science and Artificial Intelligence
Sapienza University of Rome

Introduction & Motivation

What is TalkTagger?

A chat-based guessing game challenging players to identify speakers and styles from real and AI-generated messages.

💡Jackbox/Kahoot but with a more customized, socially relevant experience using real-world chat data



Data & Preprocessing

Data Sources: Exported chat logs from WhatsApp or Discord

Preprocessing

Strips emojis, timestamps, and metadata to focus on linguistic content

Standardizes text and tags messages with speakers' names

Returns a CSV file with two columns (Name, Message)

Challenges

Handling informal, noisy real-world chat data (typos, abbreviations) is hard

Usernames are retained for gameplay, but future iterations may include pseudonymization for privacy

Message Selection

Selected based on the following properties:

- Signature words (words that are unique/frequent for the user)
- Signature phrases (same as above)
- Average message length of user
- Capitalization Patterns
- Punctuation Patterns
- All Caps Words

Remove messages that contain the users' names!

```
"selected_messages": {
  "niltheoverkill": [
    {
      "message": "Do not even breathe the same air as them",
      "distinctiveness_score": 4.97,
      "bert_similarity": 24.8,
      "is_synthetic": false
    },
```

Distinctiveness Score
how much a message matches the unique texting habits of a specific user

BERT Similarity Score
how semantically and stylistically similar a message is to user's style

User Profiles

We can use the same features to build user profiles
But why?


```
{
  "niltheoverkill": {
    "message_count": 763,
    "total_words": 5417,
    "avg_message_length_words": 7.1,
    "avg_word_length": 5.43,
    "raw_vocab_size": 990,
    "most_common_words": [
      {
        "word": "think",
        "count": 36
      },
```

```
"signature_words": [
  {
    "word": "kinda",
    "score": 5.0
  },
  {
    "word": "honestly",
    "score": 5.0
  },
```

```
"signature_phrases": [
  {
    "phrase": "make sense",
    "count": 3
  }
```

```
"capitalized_sentence_start_ratio": 0.394,
"lowercase_only_message_ratio": 0.537,
"all_caps_word_ratio": 0.05,
"proper_punctuation_ratio": 0.054,
"exclamation_count": 8,
"question_mark_count": 43,
"emoji_count": 6,
"sample_messages": [
  "ITS THE SMACK DAB MIDDLE OF ITALY",
  "did you know?",
  "Yeah??",
```

Personality Modeling

(and Text Generation)

Version 1 - Algorithmic NLP Approaches

- Tried RAKE and TextRank etc.
- Generated messages felt robotic (not so human)
- Sentences didn't make sense (no understanding of context)

Version 2 - Fine-tuned Model

- Much smarter: learns to tag based on patterns, context, tone, etc.
- Training on the spot takes sooo much time
- A pre-fine-tuned model → time + different tones (ESL)

Version 3 - Works!

- Mistral API (transformer-based LLM) in a zero-shot setting
- Prompts are engineered using user profiles (statistical features and sample messages)
- Faster gameplay by avoiding local fine-tuning on user data

Almost there...

Prep everything for gameplay

- generate superlatives ❓
  - shown at the end of the game
- everything in game_data.json
  - randomly select some real msgs
  - randomly select some synthetic msgs
  - insert superlatives

Backend done✅

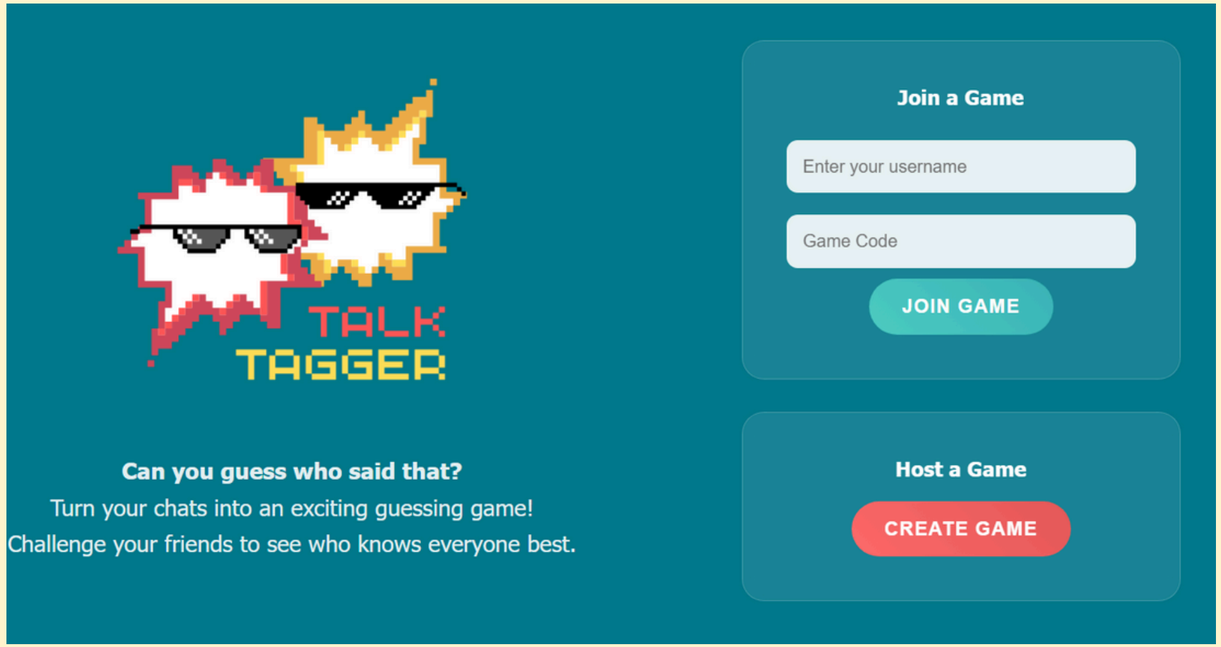
Frontend

Why a web-based game?
→ Wanted multiple people to play, and everyone has a phone/device

This required **WebSockets** to implement (host screen and player screens operate differently)

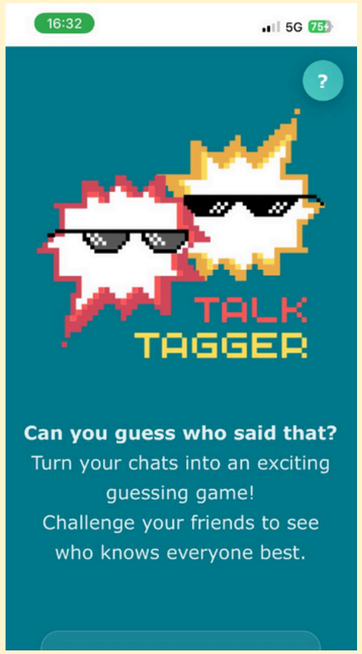
A **Flask** app was used for the game logic

And the formatting/styling was done with **HTML and CSS**



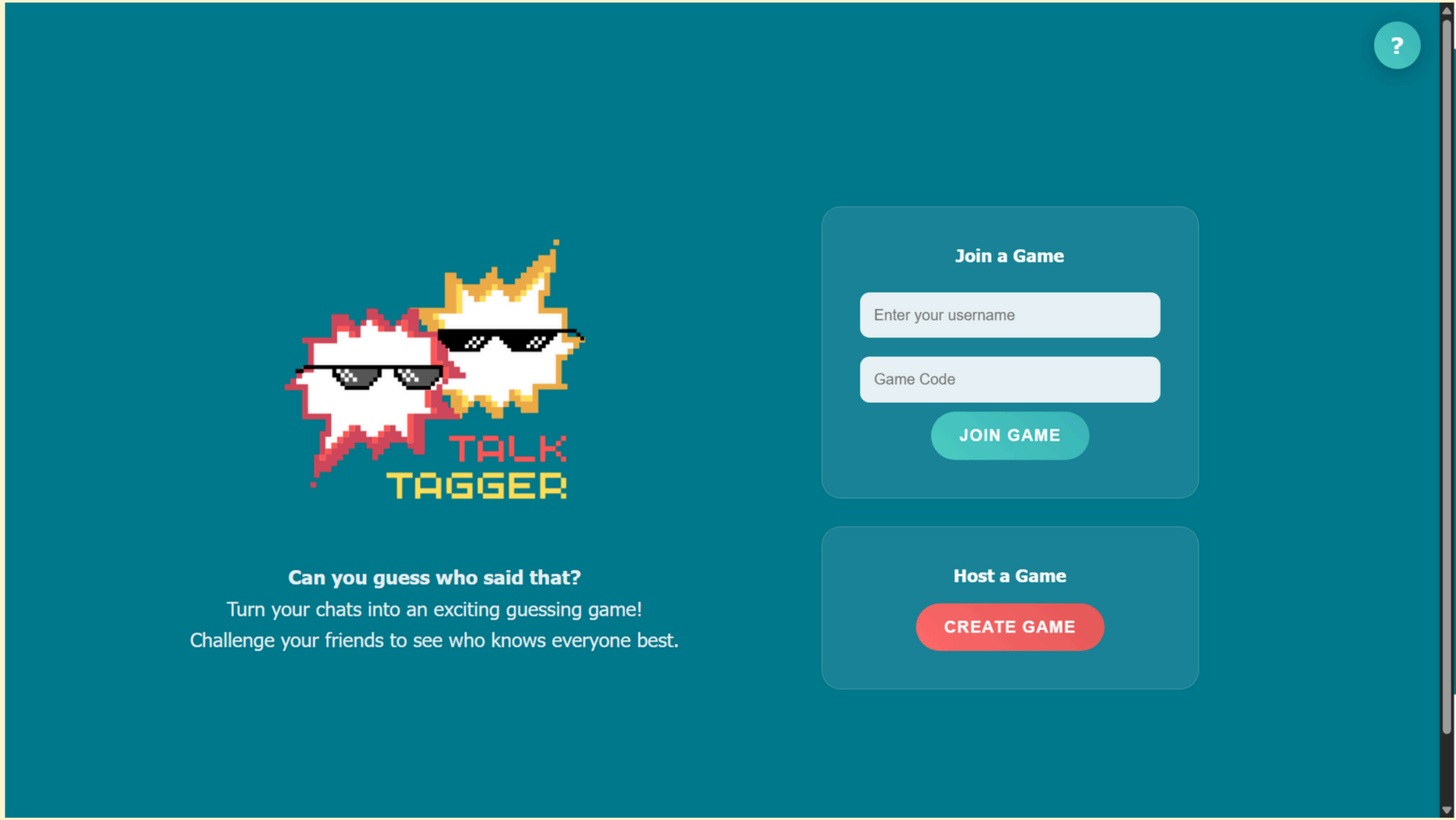
Host Screen

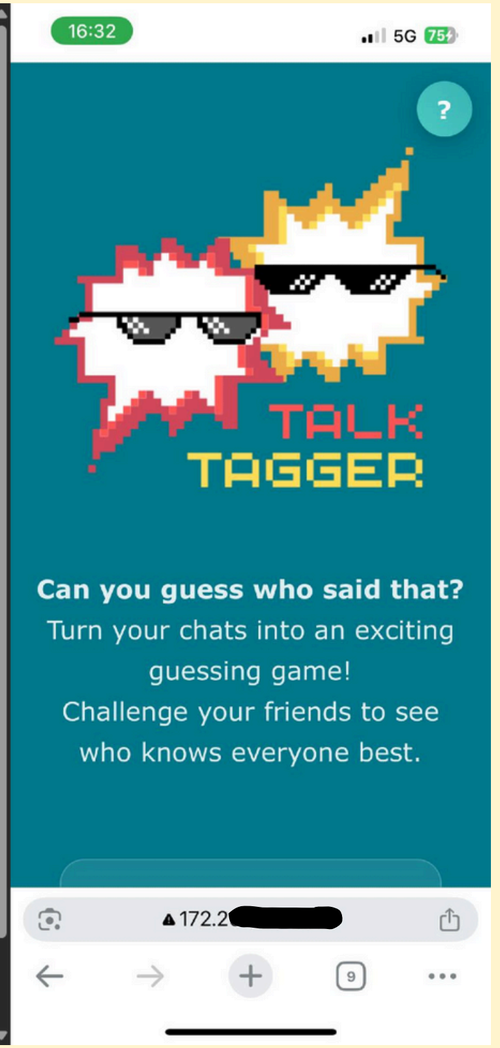
Now let's see how the game works



Player Screen

Gameplay Screenshots - 1

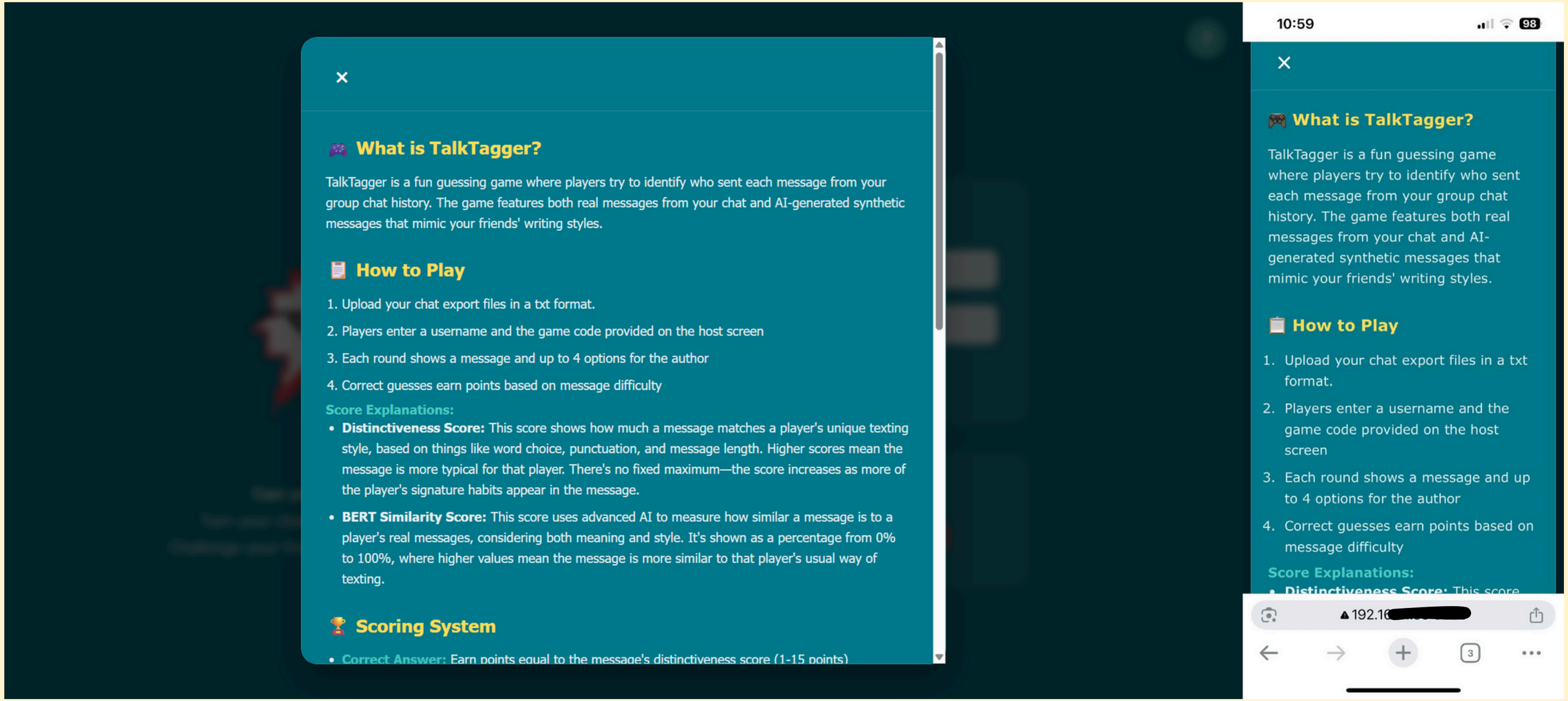




Host Screen

Player Screen

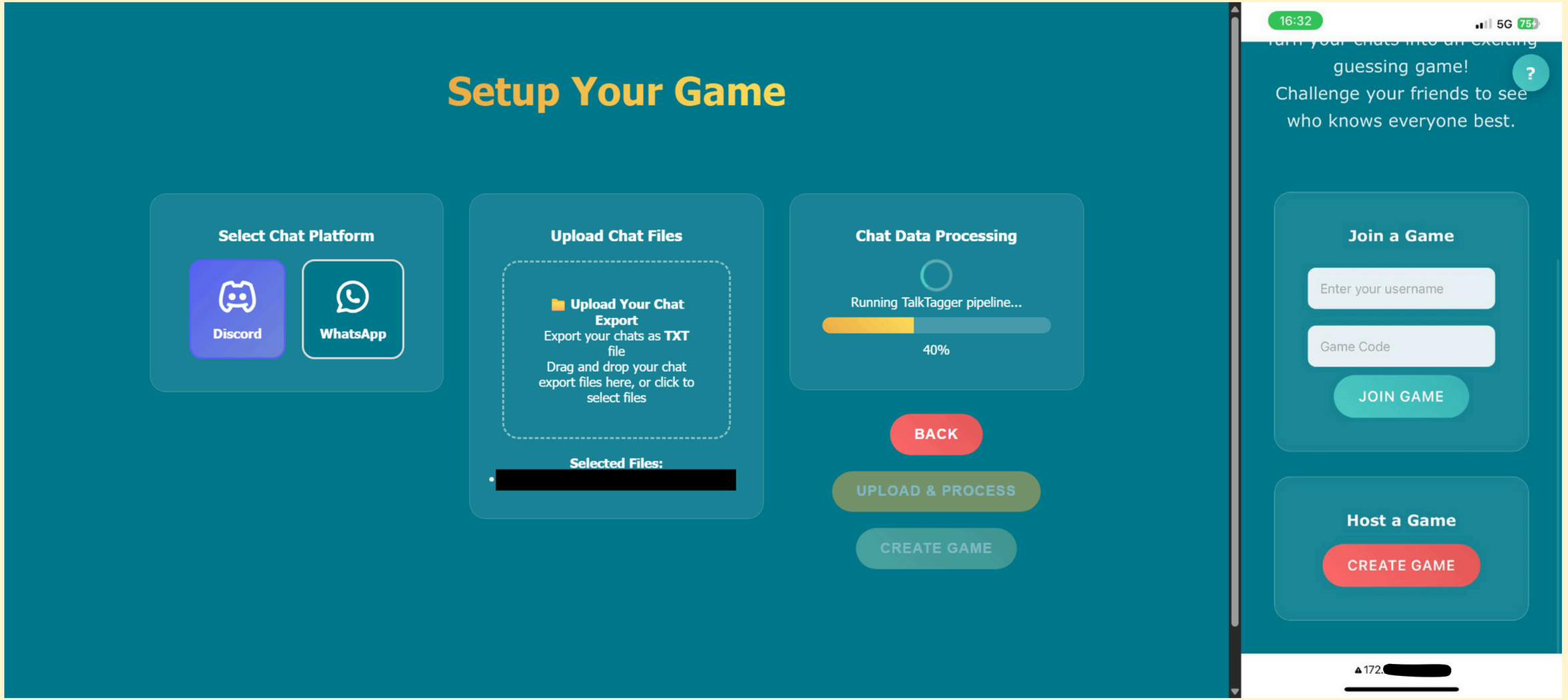
Gameplay Screenshots – 2



Host Screen

Player Screen

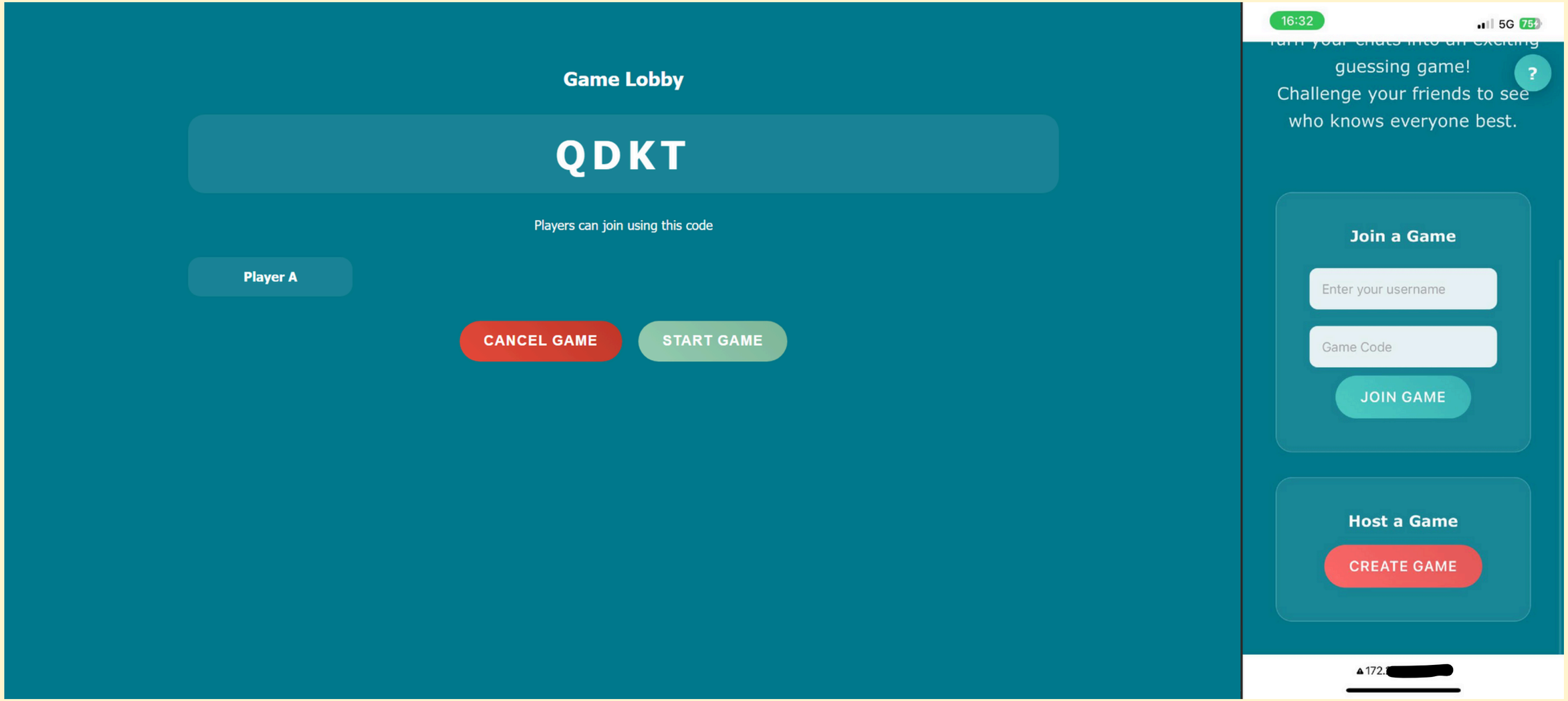
Gameplay Screenshots – 3



Host Screen

Player Screen

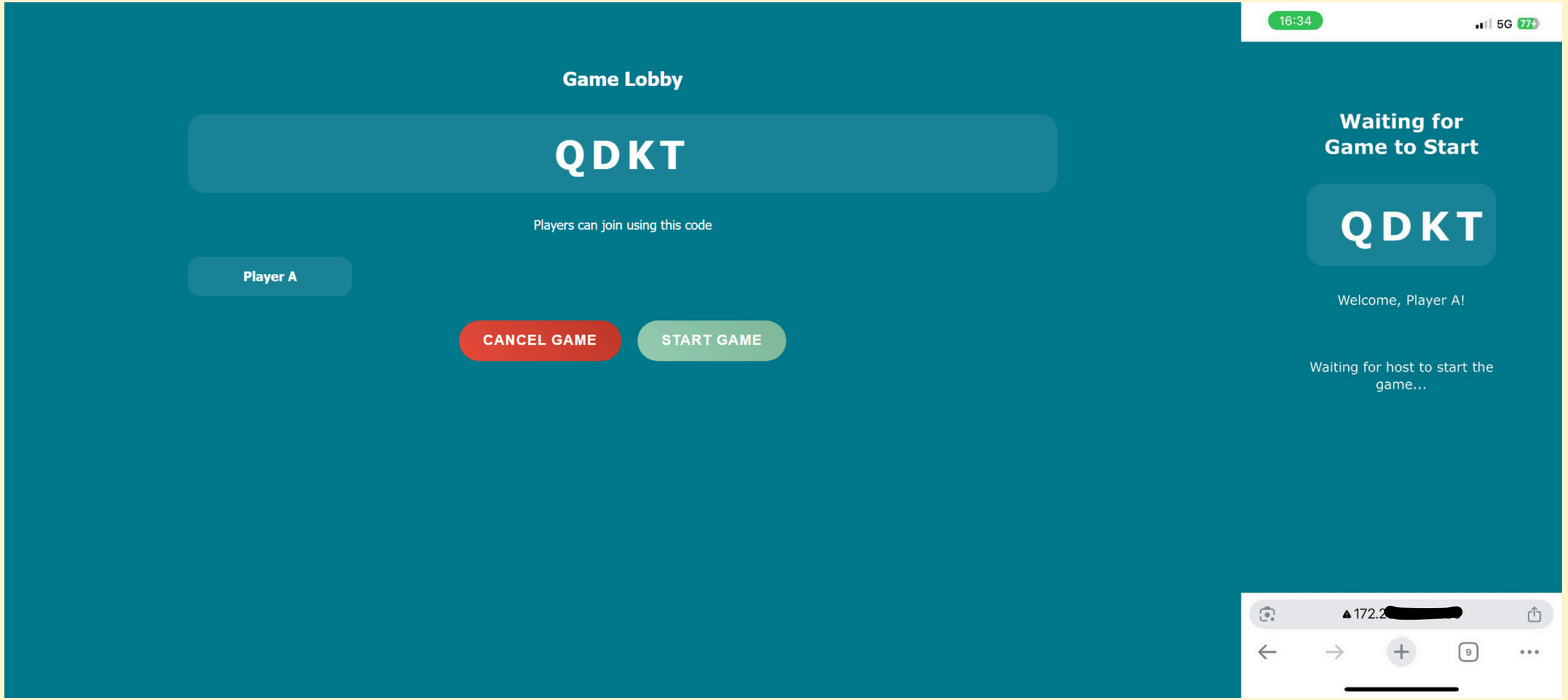
Gameplay Screenshots – 4



Host Screen

Player Screen

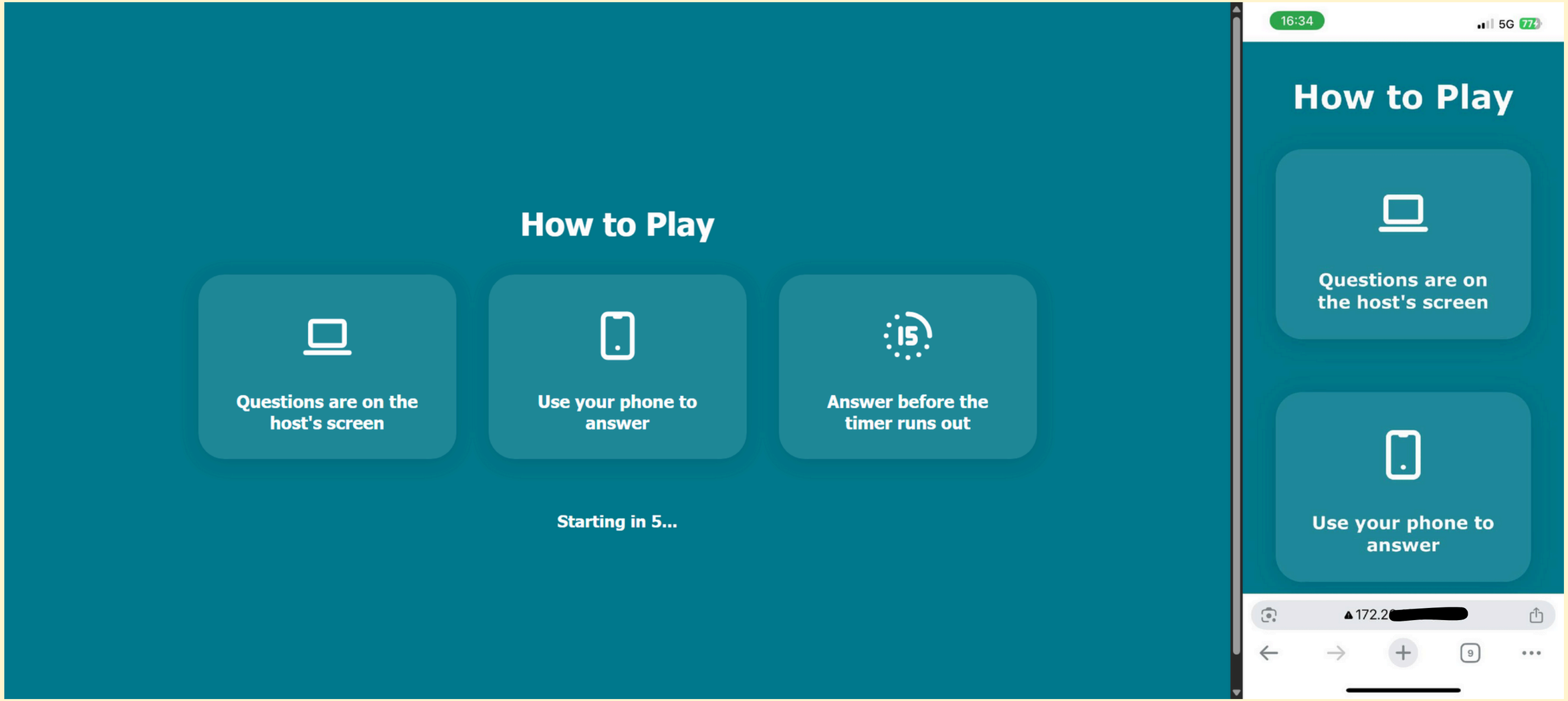
Gameplay Screenshots – 4



Host Screen

Player Screen

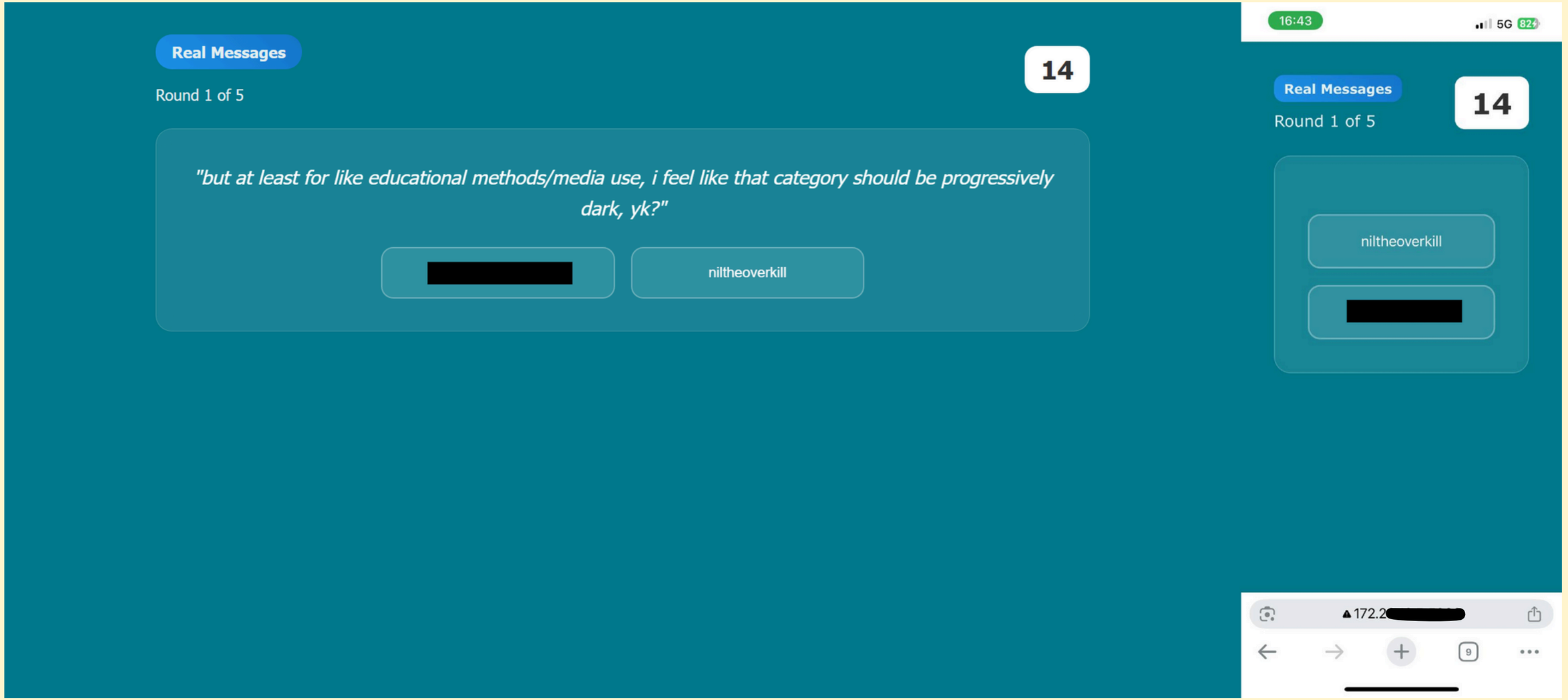
Gameplay Screenshots – 5



Host Screen

Player Screen

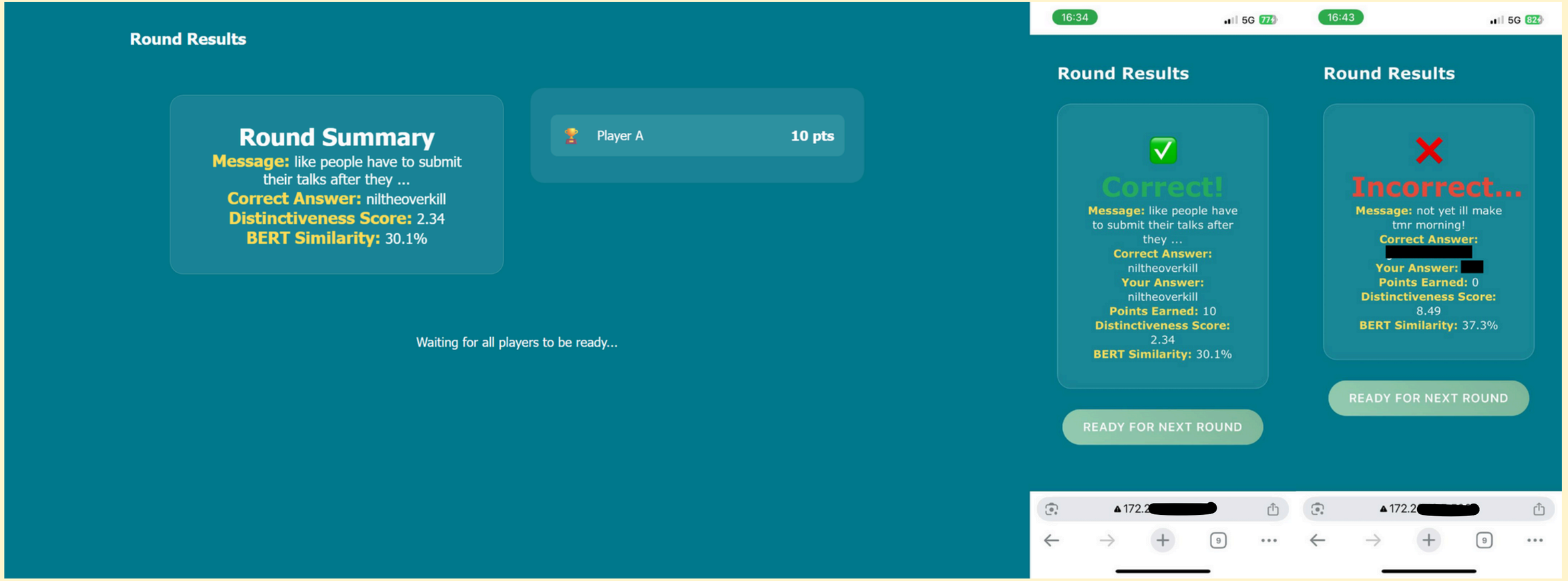
Gameplay Screenshots – 6



Host Screen

Player Screen

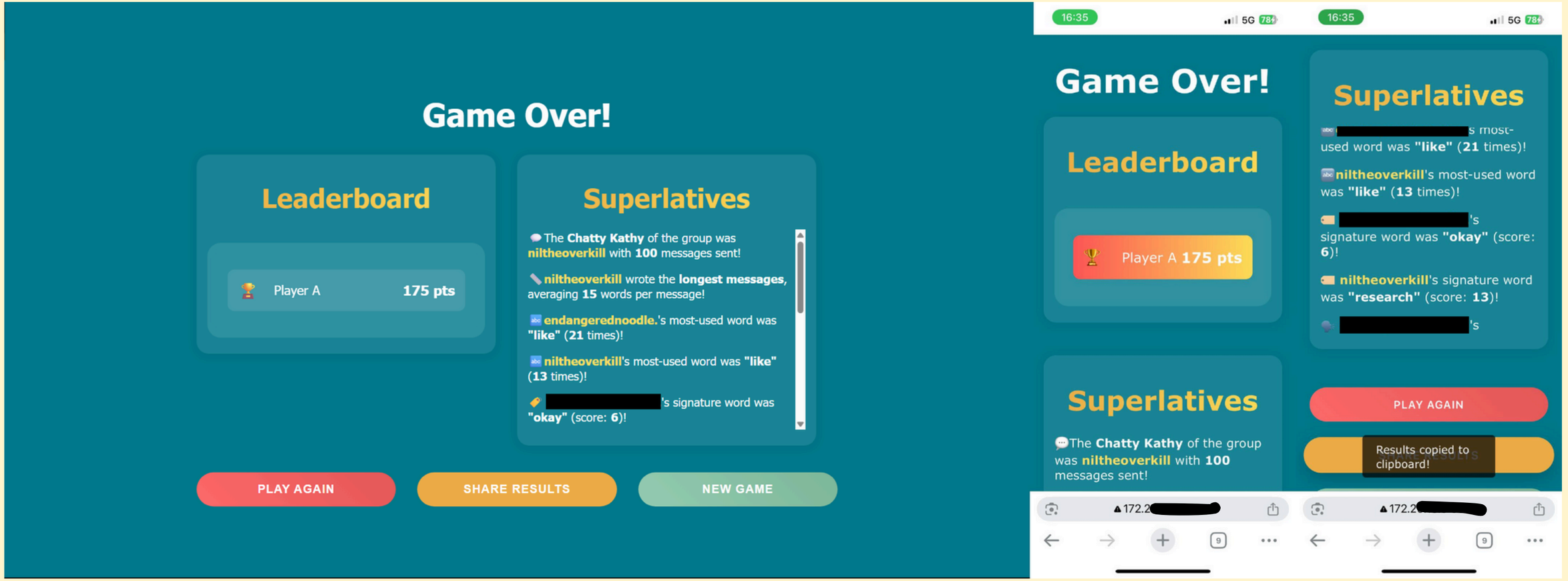
Gameplay Screenshots – 7



Host Screen

Player Screen(s)

Gameplay Screenshots – 8



Host Screen

Player Screen(s)

Thank You

Everything is available on
github.com/nilatabey/TALKTAGGER