Nomadic

Arman Faruqui

40181707

CART 451

Sabine Rosenberg

Concept

My project aims to propose an alternative way that social media algorithms can feed their users content. The social media landscape is currently being dominated by Instagram and Tiktok; 2 apps that can attribute a good portion of their success to the machine learning algorithms that determine what content they should serve to each individual user. Based on a variety of factors, these algorithms are able to infer what sort of content users are familiar and comfortable with. My application aims to do the same thing, but instead of using these insights to feed users with familiar content, I attempt to take them out of their comfort zone to have them interact with content they normally wouldn't. My app simply provides users with the functionality to flick through videos with the option of interacting with them by selecting one of 5 different emojis.

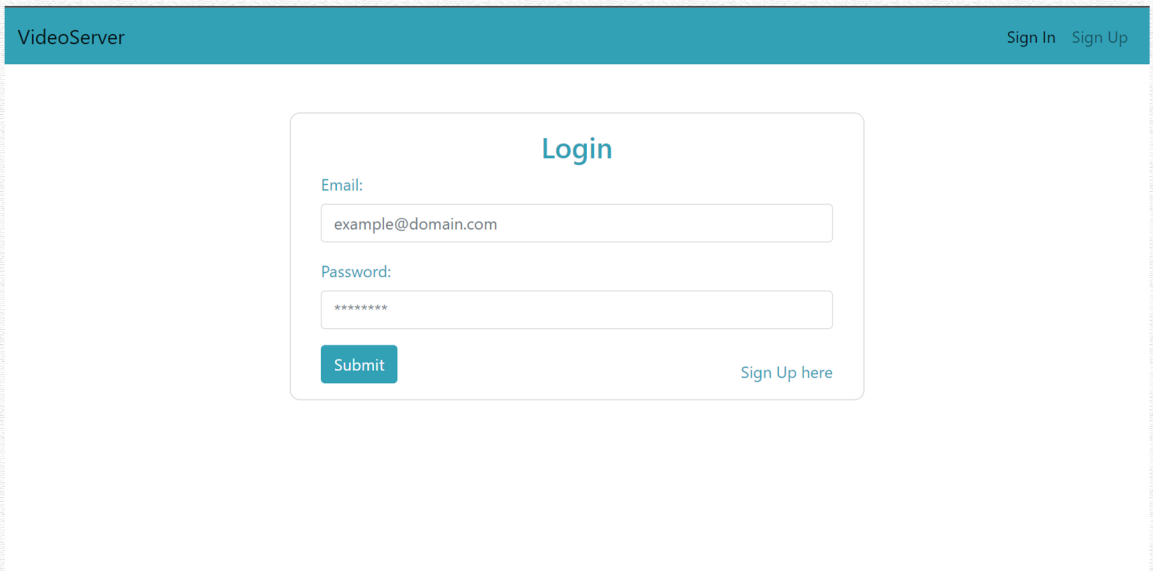
Inspiration

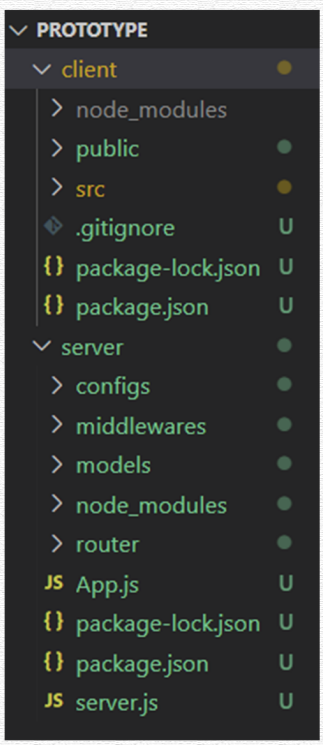
I have been heavily inspired both aesthetically and conceptually by the era of cable television being the primary source of information and entertainment within the household. I remember every time I'd use it, I'd first check the few chanels I was familair with to see if anything interesting was on. After that, I would explore the abyss of random Pakistani, Turkish, Iranian, and Western news, infortainment, and entertainment channels. I'd usually have no idea what was going on, but the active involvement I'd have to make in order to apply meaning and form contexts made the experience feel engaging and educational.

I want my application to feel engaging and entertaining to use purely through the novelty of exposing users to content they normally wouldn't see on their social media feed, whilst also critically engaging with the current landscape of machine-learning algorithms used for social media platforms.

Implementation

My first order of business was to build a framework upon which my code would run. I decided to work on a node-based server side that would connect to Mongo Database, along with a client-side react app. Users could sign up or sign in by posting data to the database. The passwords were encrypted using bcrypt. Upon a successful login, users would be provided with a unique token that would validate the validity of their session. After a few weeks of following tutorials and help from my professor, I finally got a functioning interface.






```
const express = require("express");
const mongoose = require("mongoose");
const app = express();

const morgan = require("morgan");
const cors = require("cors");
const bodyParser = require("body-parser");
const { application } = require("express");

mongoose.connect(
  "mongodb+srv://armanfaruqui:mediumalgorithm@atlascluster.kbarcly.mongodb.net/?retryWrites=true"
  { useUnifiedTopology: true, useNewUrlParser: true }
);
mongoose.Promise = global.Promise;

app.use(morgan("dev")); // HTTP Request middleware
app.use(cors()); // Prevents cross origin resource sharing errors. ie when client receives
//response from an incorrect port
app.use(bodyParser.urlencoded({ extended: false })); // Makes JSON data more readable
app.use(bodyParser.json());

// app.post("/api/signIn", function(req, res){
//   console.log(req.body)
// })

//Routes
app.use("/api/signUp", require("./router/signUp"));
app.use('/api/signIn', require('./router/signIn'));


module.exports = app;
```

Content Generation

News/InFortainment

Now it was time to start compiling the video content that would be displayed on my app. One of the primary reasons individuals consume content on social media platforms is to stay updated on current affairs around the world. I categorized this theme of content I would be collecting under the title of 'News and Infotainment.'

I started by creating a google docs file consisting of 6 headings related to different geographical regions of the world; North America, Latin America, Europe, Africa, Middle East, and East Asia. I, along with the help of a few of my friends, filled this document with Youtube URLs of news and infotainment content. We made an effort to actively try and have these videos be sourced directly from those regions. For example, we avoided having videos about Africa created by the BBC and CNN be put under the 'Africa' heading. Social media algorithms by design have a tendency to continously provide news content from sources that user's agree with and support, which is something I want my application to challenge. It becomes impossible for 2 people to agree on something when their truths are sourced from 2 different realities.

It was surprisingly challenging to find from certain regions of the world, such as China and Russia, which I found to be extremely problematic. The west has created the impression of supporting the freedom of access to information, yet it was a lot more challenging than it should have been to find news videos from China and Russia that did not automatically villanize them. I really hope to work on more projects in the future that examine the biases individuals from the west have towards these Eastern countries due to a lack of access to their perspectives and realities.

https://www.youtube.com/watch?v=_NBr1meMjLU&ab_channel=PakistanObserver
https://www.youtube.com/watch?v=0yJwGD4ic-c&t=16s

Europe
https://www.youtube.com/watch?v=IbdCaaW2cDc
https://www.youtube.com/watch?v=ChrOpdAr7fY
https://www.youtube.com/watch?v=DkAb7SmH-PY
https://www.youtube.com/watch?v=pbixAzhDLrU
https://youtu.be/qQUZf161raI
https://www.youtube.com/watch?v=N__aBAGEFdk
https://www.youtube.com/watch?v=37wCScaWO8Q
https://www.youtube.com/watch?v=_TJz-uSZs60
https://www.youtube.com/watch?v=ZthI4OvwE04&list=PLCUKIeZnrIUmLL9IWxWPo2OXYPk-xtDdO&index=41
https://www.youtube.com/watch?v=JJlHcXxuj-4&list=PLCUKIeZnrIUmLL9IWxWPo2OXYPk-xtDdO&index=71
https://www.youtube.com/watch?v=LHSsFqpd2ns&list=PLT6yxVwBEbi0Q8wAkcLI8eI1T3LLM41Yq&index=89
https://www.youtube.com/watch?v=n128SW1qPEM&list=PLT6yxVwBEbi0Q8wAkcLI8eI1T3LLM41Yq&index=99
https://www.youtube.com/watch?v=INX_3MiLB0s&list=PLT6yxVwBEbi0Q8wAkcLI8eI1T3LLM41Yq&index=109
https://www.youtube.com/watch?v=gIc0j9qUVAY
https://youtu.be/Gi3Pq8iMasA
https://youtu.be/zqlTBGKqD3M
https://youtu.be/pwga_ivwKLo

Af
ht
ht
ht
ht
ht
ht
ht

```
const links = {
    "northAmerica": [
        "https://www.youtube.com/watch?v=MAVfqeTLN94",
        "https://www.youtube.com/watch?v=FliLpEcYyeE",
        "https://www.youtube.com/watch?v=Nk6-M7v0y10",
        "https://www.youtube.com/watch?v=XaGZKuc0G4k",
        "https://www.youtube.com/watch?v=bOAuG0MAlOg",
        "https://www.youtube.com/watch?v=RY_NrjMUGfk",
        "https://www.youtube.com/watch?v=GAMoMFiTqtk",
        "https://www.youtube.com/watch?v=09Ui0F-tIVs",
        "https://www.youtube.com/watch?v=gIK8AMa4KDc",
        "https://www.youtube.com/watch?v=9358a-8eUXs",
        "https://www.youtube.com/watch?v=qJu6tDZmwds",
        "https://www.youtube.com/watch?v=3ju1MuBNCHg",
        "https://www.youtube.com/watch?v=pCn6_vMYUos",
        "https://youtu.be/UvIJmoMjI9c?t=65",
        "https://youtu.be/t_x2ey2IDDk?t=14",
        "https://youtu.be/I0Mcg1OO8yE?t=22",
        "https://www.youtube.com/watch?v=7x7fsPEivPU",
        "https://www.youtube.com/watch?v=wUoSbETkS2Q",
        "https://youtu.be/-_OzymdJ03c?t=105",
        "https://www.youtube.com/watch?v=40TA9OkML2k",
        "https://www.youtube.com/watch?v=jKE5Nv9CQwI",
        "https://youtu.be/B9SpZOP5UbQ"

    ],
    "latinAmerica": [
        "https://youtu.be/NkdUs2Dxv4s?t=17",
        "https://youtu.be/8IObz87Be9Q",
        "https://youtu.be/ihecR8gt0WE",
        "https://youtu.be/ah95W85hb5g",
        "https://www.youtube.com/watch?v=JWe9IigF5SU&ab_channel=BreakThroughNews",
        "https://www.youtube.com/watch?v=BEQWy8LLO88",
        "https://www.youtube.com/watch?v=qYxHx3OsD88",
        "https://www.youtube.com/watch?v=JvPE8jifIV0",
        "https://www.youtube.com/watch?v=n_o3u17HtAg",
        "https://www.youtube.com/watch?v=YcAqrRXRluU",
        "https://youtu.be/c4iTedP-7Ek?t=49",
        "https://www.youtube.com/watch?v=BKykW7gRFjc",
        "https://www.youtube.com/watch?v=FYM4eHH_oDw",
        "https://www.youtube.com/watch?v=KmgRXXNvokE",
        "https://www.youtube.com/watch?v=SEnehFGA6Gg",
        "https://www.youtube.com/watch?v=hUoqt5Goa1w"
```

Entertainment

When it came to finding a way to collect entertainment-based content for my app, I immediately realized that I could not rely on the same method. Entertainment and
humor are simply too broad and subjective of a concept to categorize into genres and collect videos for accordingly. Luckily, I had access to code for an application that employed a method that was also a major theme of this class; crowdsourcing.

This code would search through specified subreddits on Reddit, sort through them in a specified manner, grab a post, convert it to JSON data, and then extract the YouTube URL and video ID from it. That video ID would then be pushed into a Youtube iframe API, and be played on the screen.

This method provided me with a lot of important benefits. I was able to manipulate the iframe player so that the user could not pause and interact with the video as if it was a channel on television. Also, by crowdsourcing the videos, I had access to a seemingly unlimited, constantly updated pool of videos posted on those subreddits.

The only downside was that I was unable to get this code to work alongside the login and database framework I created earlier, which I then decided to abandon. Whilst this was extremely disheartening at the time, it forced me to think about how my algorithm should work in a different way, which lead to an experience that could be had in a lot shorter of a time.

Algorithm Time

With the time frame I had, along with the other projects I was working on, I realized that creating a machine-learning algorithm was too tall an order to do for this project. I instead decided to work on an if-statement based algorithm that employed a few of the principles we studied about fuzzy logic. I implemented this for the news and infotainment videos I had collected.

Each region array had a corresponding 'regionScore' variable attached to it which was assigned a value of 20. This score would determine the likelihood of a video from that region being selected. Based on the user's behavior these scores would be adjusted throughout the experience. The way it all would work was that a video would be selected by the scores all being totaled, followed by a random number between 0 and the total being generated to determine which range of scores that random videos fell under. The higher a region score, the higher the chance a video would play from that region, whilst there still would exist a chance that a video from an array with a low score could still be selected.

Once the array is selected, a URL is popped from it, eliminating the chance of it being played more than once, and then fed into the Youtube Iframe player.

```
// Scores used by algorithm to determine which content users are more comfortable with
var northAmericaScore = 20;
var latinAmericaScore = 20;
var europeScore = 20;
var africaScore = 20;
var eastAsiaScore = 20;
var middleEastScore = 20;

var count = 0; // Counts number of videos gone through
var totalScore = 60; // Algorithm scores
var currentScore = 0;
```

```
// Loads video IDs from array of youtube urls
var loadNews = function (){
    totalScore = northAmericaScore + latinAmericaScore + europeScore + eastAsiaScore + middleEastScore + africaScore;
    currentScore = Math.floor(Math.random() * totalScore);
    console.log(totalScore)
    console.log(currentScore)
    if (currentScore >= 0 && currentScore <= northAmericaScore){
        region = 'northAmerica';
        // Load video from North America array
        let url = links.northAmerica.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
    else if (currentScore > northAmericaScore && currentScore <= northAmericaScore + latinAmericaScore){
        region = 'latinAmerica';
        // Load video from Latin America array
        let url = links.latinAmerica.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
    else if (currentScore > northAmericaScore + latinAmericaScore && currentScore <= northAmericaScore + latinAmericaScore + europeScore){
        region = 'europe';
        // Load video from Europe array
        let url = links.europe.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
    else if (currentScore > northAmericaScore + latinAmericaScore + europeScore && currentScore <= northAmericaScore + latinAmericaScore + europeScore + africaScore){
        region = 'africa';
        // Load vide form Africa array
        let url = links.africa.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
    else if (currentScore > northAmericaScore + latinAmericaScore + europeScore && currentScore <= northAmericaScore + latinAmericaScore + europeScore + eastAsiaScore){
        region = 'eastAsia';
        // Load video from East Asia array
        let url = links.eastAsia.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
    else if (currentScore > northAmericaScore + latinAmericaScore + europeScore + eastAsiaScore && currentScore && currentScore <= totalScore){
        region = 'middleEast';
        // Load video from Middle East array
        let url = links.middleEast.randomPop();
        let id = youtube_parser(url)
        console.log(`id ${id}`)
        return {"link": url, "video": id }
    }
}
```

Yes, I could have made cleaner code by using a second function wiht paramters, but I thought that leaving like this would make the concept behind it seem a bit more obvious.

Time Spent

The time users spend on each video was one of the two factors that influenced the region's score for the algorithm. Upon the start of each video, a timer begins which is only stopped once the user changes the channel. Depending on what range the time spent falls under, a score is subracted from the region score that the video corresponds to.

If a user spends more time on videos from North America, I as the designer assume that this is the type of news content that this individual normally consumes, and I should therefore subtract a greater amount from this regions score to make it less likely in the rest of this algorithm cycle's duration for more of it to appear.

This is obviously flawed, as a user could also be interested in videos they normally do not engage with, which would then lead to less of this content shown to them, which essentially defeats the purpose of this app.

```
// Starts Timer
function startTimer(){
    sec = 0;
    var ele = document.getElementById("timer")
    timer = setInterval(() => {
        sec++
        ele.innerHTML = '00:' + sec;
    }, 1000);
}

// Ends timer and updates the algorithm
function endTimer(){
    var timeSpent = sec;
    console.log(`TIme score ${timerSpent}`)
    clearInterval(timer)

    if (timeSpent > 8 && timeSpent <= 30){
        timerScore = 1;

    }
    else if (timeSpent >30 && timeSpent <= 45 ){
        timerScore = 2;
    }
    else if (timeSpent > 45 && timeSpent <= 60){
        timerScore = 3;
    }
    else if (timeSpent > 60){
        timerScore = 4;
    }

    if (region === "northAmerica"){
        northAmericaScore = northAmericaScore - timerScore;
    }
    else if (region === "latinAmerica"){
        latinAmericaScore = latinAmericaScore - timerScore;
    }
    else if (region === "europe"){
        europeScore = europeScore - timerScore;
    }
    else if (region === "africa"){
        africaScore = africaScore - timerScore;
    }
    else if (region === "eastAsia"){
        eastAsiaScore = eastAsiaScore - timerScore;
    }
    else if (region === "middleEast"){
        middleEastScore = middleEastScore - timerScore;
    }
}
```

Emojis

This is the second factor that influences the algorithm. Users have the option to select an emoji out of 5 options displayed on their screen. Depending on the emoji selected and the region from which the current video is playing, an amount is either added or subtracted from the region score. Points are subtracted if the user selects the first 2, and added if they select the last 3.

This is based on the fact that I want users to be engaging with content they find intriguing or shocking. It's again a flawed system as a user can still find content they are unfamiliar with charming and funny

```
// Allows emoji selection to affect the algorithm
    function emojiSelect(){
        emojiClicked = false;

        if (emojiClicked === false){
            love.on("click", emojiAlgorithm(2))
            lol.on("click", emojiAlgorithm(2))
            mono.on("click", emojiAlgorithm(-1))
            wut.on("click", emojiAlgorithm(-2))
            whoa.on("click", emojiAlgorithm(-3))
        }
    }

    function emojiAlgorithm(score){
        if (state === "array" && region === "northAmerica"){
            emojiScore = score;
            northAmericaScore = northAmericaScore - emojiScore;
            emojiClicked = true;
        }
        if (state === "array" && region === "latinAmerica"){
            emojiScore = score;
            latinAmericaScore = latinAmericaScore - emojiScore;
            emojiClicked = true;
        }
        if (state === "array" && region === "europe"){
            emojiScore = score;
            europeScore = europeScore - emojiScore;
            emojiClicked = true;
        }
        if (state === "array" && region === "africa"){
            emojiScore = score;
            africaScore = africaScore - emojiScore;
            emojiClicked = true;
        }
        if (state === "array" && region === "eastAsia"){
            emojiScore = score;
            eastAsiaScore = eastAsiaScore - emojiScore;
            emojiClicked = true;
        }
        if (state === "array" && region === "middleEast"){
            emojiScore = score;
            middleEastScore = middleEastScore - emojiScore;
            emojiClicked = true;
        }

    }
```

Algorithm Reset

As I mentioned earlier, not being able to have users log in and have their data stored in a database meant I needed my algorithm to work in short bursts instead of over a longer duration. I created a function that resets the region scores each time the user flicks through 20 videos.

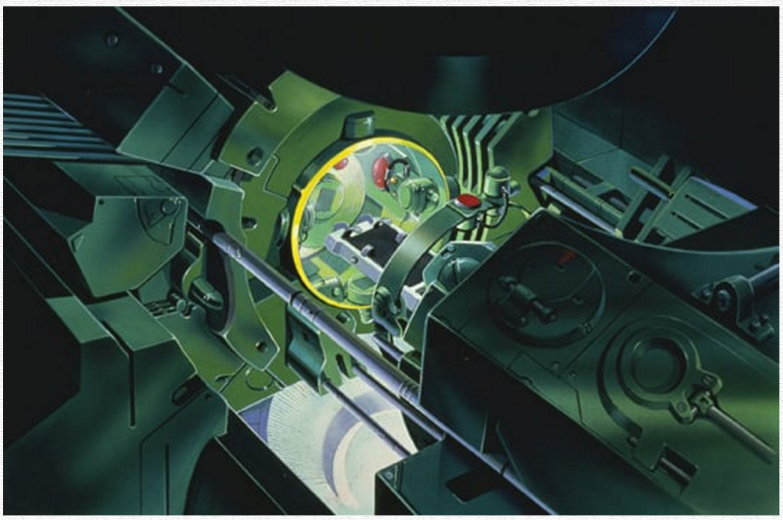
I really enjoyed how this allows the user to experience the different ways the algorithm can tailor their experience in one sit-down session. I feel it also helps make up for the slightly flawed nature of how users influence the algorithm in the first place.

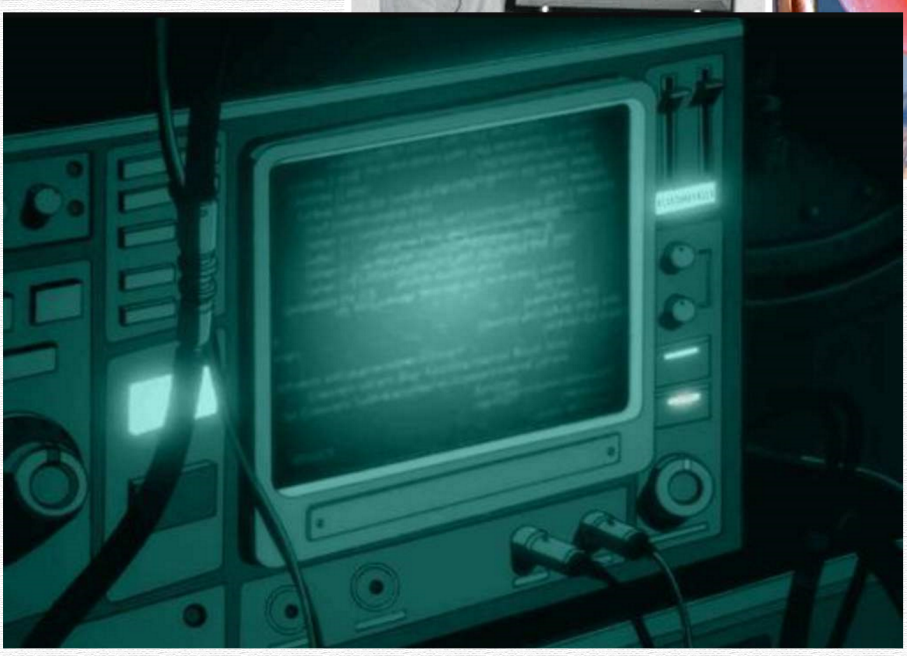
I think that if social media platforms also employed some sort of an algorithm reset every few weeks or so, their user's would be exposed to a lot more content they may find more engaging, and develop new interests and understanding they otherwise would be kept away from.

```
// Resets the variables that affect which video is likely to be displayed after 20 cycles
var resetAlgorithm = function(){
    if (count => 20){
        northAmericaScore = 20;
        latinAmericaScore = 20;
        europeScore = 20;
        africaScore = 20;
        eastAsiaScore = 20;
        middleEastScore = 20;
        count = 0;
    }
}
```

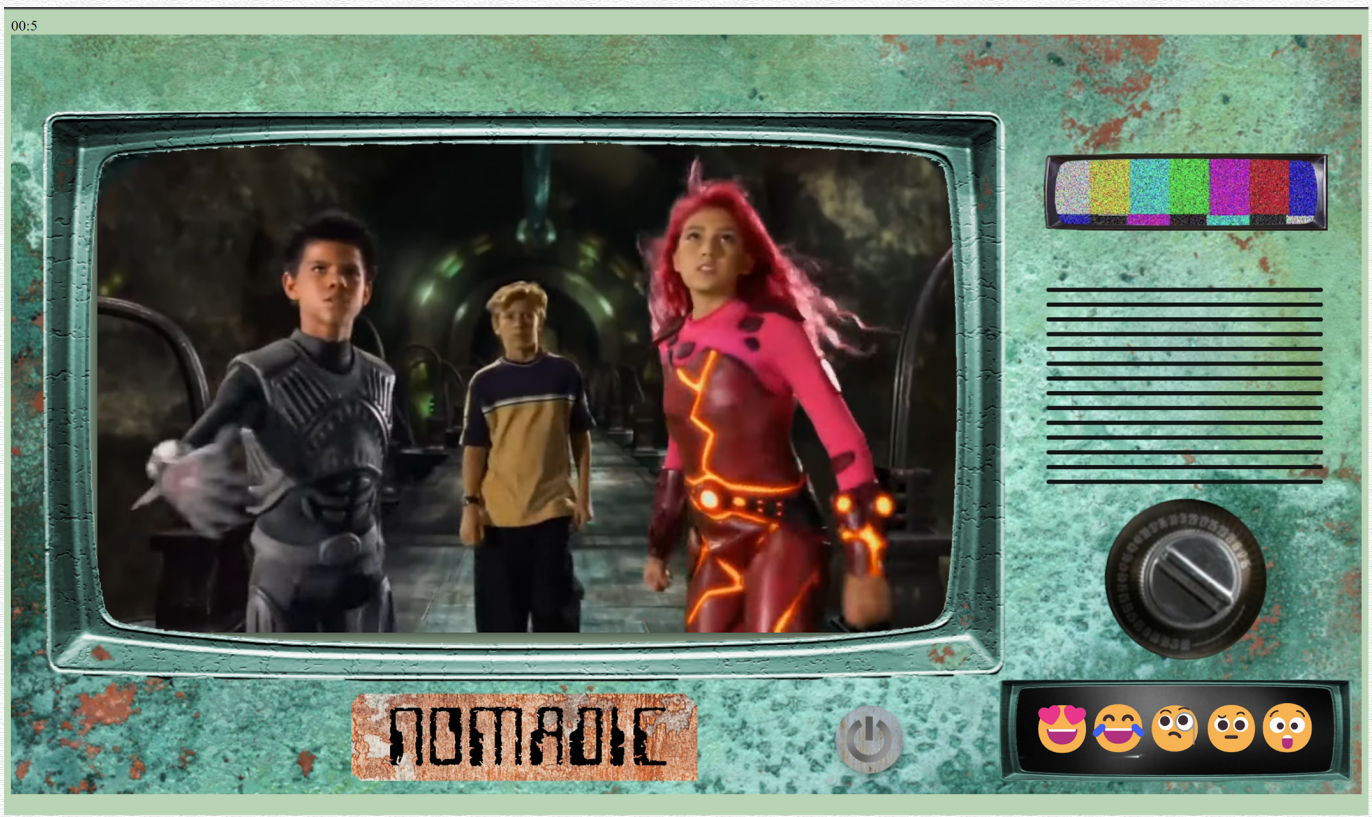
Aesthetic Inspiration

For the interface, I drew inspiration from cyberpunk technology as well from old-school retro televisions.






CGTN
Did the Western media get the Irani...
Watch later
Share
CGTN is funded in whole or in part by the Chinese...
But then, again I think if like for example,
when foreigners come inside Iran, they would be surprised.
WORLD INSIGHT
PROTESTS IN IRAN
Misconceptions about the compulsory wearing of the hi
YouTub
NOMADIC


00:5
NOMADIC

For the Future

Remote Functionality

To further push the atmosphere of feeling like you're flicking through cable television, I think it can be interesting to have the user control the TV with a remote they can download on their smartphone. I fel it would create a barrier between the user and the interface that used to exist with televisions, but no longer does with touch screens that feel like an extention of us.

User Feedback

I think it is important to have ways for users to give feedback on their experience and how they would prefer it to be tailored. This app is more of an experiment than a product, which is incomplete without user data and feedback.

A better experience

In the future, I would like the app's content to be sourced from more than just premade arrays and predefined subreddits. Perhaps sourcing video from facebok and twitter would allow me to have an algorithm that could infuence what sort of entertainment based content is fed to the users.