

# **IDEA9101 - ASSESSMENT 2:**

## **AI-BASED DISTRIBUTED RESPONSIVE ENVIRONMENT**

### **PROCESS DOCUMENTATION**

Yiyang Ren \_yren4072 Ruijuan Mao \_rmao8944

Yiyun Zhang \_yzha6851 Jing Chen\_jche5588 Yuqi Zhou\_yzho6208

# INDEX

- **Phase 1: Technology Choice Analysis**
- **Phase 2: UI design process**
- **Phase 3: Coding stage**
- **Phase 4: What's next**
- **Phase 5: Appendix**

## PHASE 1: TECHNOLOGY CHOICE ANALYSIS

To learn about target users' needs, preferences and feedback, we have conducted a survey online and user interviews.

In talking with tutors, we find that it is challenging to design a touching screen as our interaction mode. Due to the musical atmosphere in bars, guests are usually distracted by the performance, resulting in the wrong click. Besides, camera tracking requires guests to do particular poses in particular areas, which makes guests feel uncomfortable and unfree.

Taking user-friendliness, sociality and convenience into consideration, we finally find that semantic analysis and text input best suit our design. Through semantic analysis, we can analyze users' emotions in real time and then recommend corresponding cocktails. In this way, instead of doing a particular pose or touching the screen, users can engage in the interaction by simply entering text, which can avoid them miss live music shows. In other words, this kind of interaction mode is not only convenient for users, but also keeps their attention focused on the music and performance.

- DESIGN SOLUTION

- INPUT MECHANISMS:

- Touch
- Typing text
- Semantic analysis

- OUTPUT MECHANISMS:

- Visual graphics
- Sound

- SYSTEM ARCHITECTURE

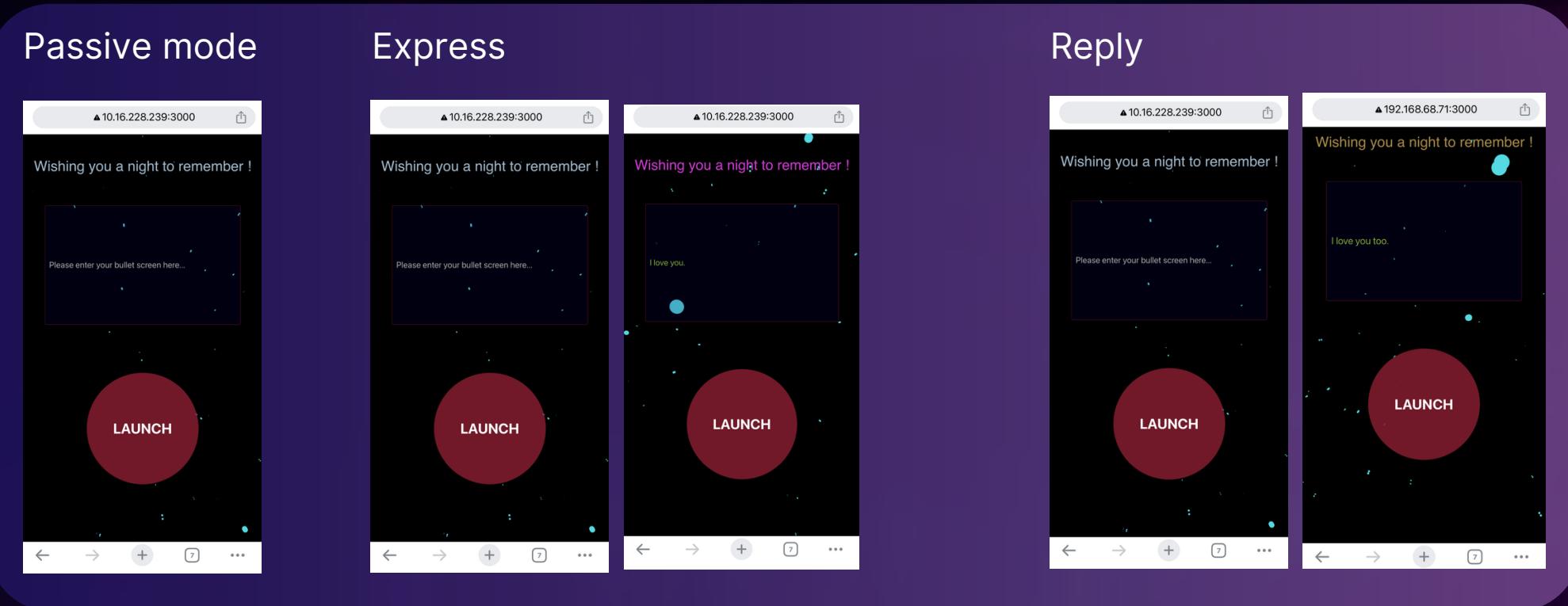
- Client-server
- Peer-to-peer
- The client should be a p5js app to be loaded on a smartphone, running on the computer

- DESIGN SYSTEM

- Interface Design:
- Input text and send message button
- Visual communication

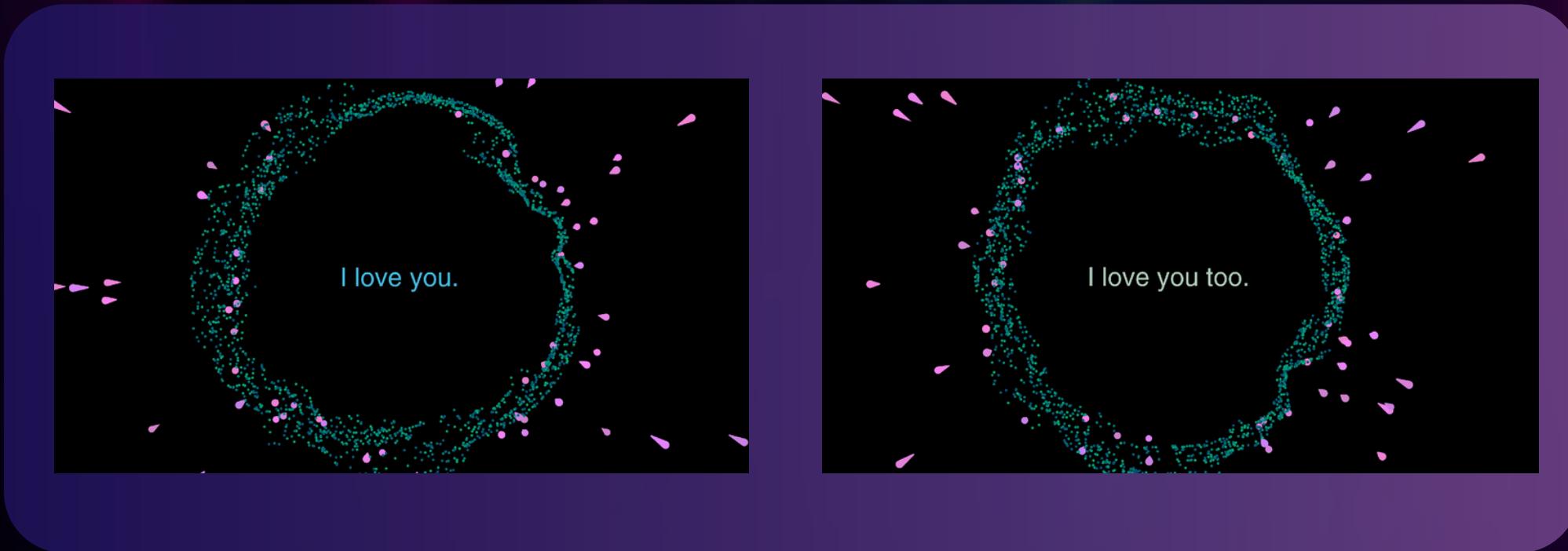
# PHASE 2: UI DESIGN PROCESS INTERATIONI

## CLIENT

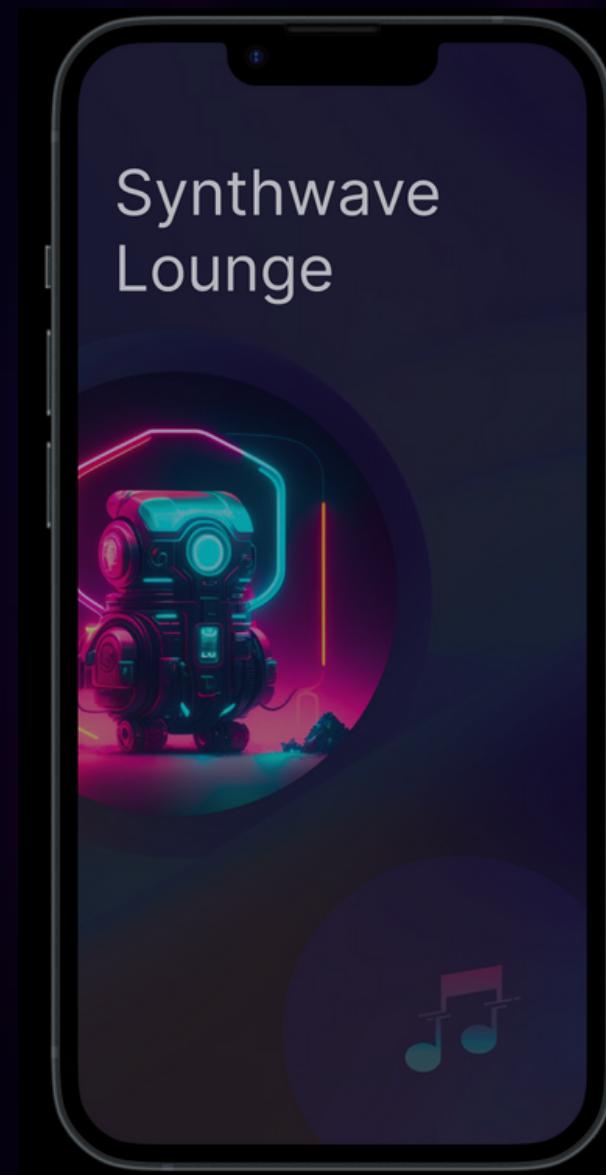


Based on the UI design in the Assignment, we kept the function to send bullet comments

## SERVER



# PHASE 2: UI DESIGN PROCESS



Initial Aesthetic

Inter  
Semi Bold ▾ 48

Font Styles



Product Logo

<span style="color: #EC71A4;">■</span>	EC71A4	70%
<span style="color: #09081C;">■</span>	09081C	100%

Color

# PHASE 2: UI DESIGN PROCESS--CLIENT INTERACTION2

**Cocktail recommendation**

Mood Range	Cocktail Name	Description
Sadness: 0-20%	Manhattan	Dear guest, your mood today is suitable for drinking Combine dry and sweet vermouth to make the perfect Manhattan cocktail. Drink it and let everything sad go with the wind!
Anxiety: 20%-40%	Bloody Mary	Dear guest, your mood today is suitable for drinking It is one of the world's best known cocktails, prized for its ability to cheer you up. With the addition of heat sources, the overall flavor is savory and spicy.
Calm: 40%-60%	Screwdriver	Dear guest, your mood today is suitable for drinking Mix freshly squeezed orange juice with vodka and Angostura bitters for a classic screwdriver cocktail. Bring the taste of summer holidays to your smooth life!
Relax: 60%-80%	Mojito	Dear guest, your mood today call for Mojito Mix this classic cocktail for a party using fresh mint, white rum, sugar, zesty lime and cooling soda water. Enjoy more leisure time!
Happy: 80%-100%	Alexander	Dear guest, your mood today is suitable for drinking It's a blend of amaretto, cream, lemon curd and vanilla, finished with flowers. You can enjoy a springtime twist on a brandy alexander with Alexander cocktail!

**Start screen**

**Into screen**

**Selection Page**

**Write mood**

**Bullet comment**

**Sadness: 0-20%**

**Anxiety: 20%-40%**

**Calm: 40%-60%**

**Relax: 60%-80%**

**Happy: 80%-100%**

Add mood cocktail recommendation function to the original barrage sending function

According to the semantic analysis of emotions, each mood range value corresponds to a wine of a taste

## PHASE 2: UI DESIGN PROCESS-- SERVER INTERACTION2

screen

At the Serve side of the music barrage interactive bar, we combined the two elements of water ripples and coconut trees to create a refreshing visual effect. When the music plays, water ripples appear on the screen. The wavy form in nature represents the change of emotions, and the coconut tree, as an element symbolizing leisure and relaxation, makes the audience feel relaxed and happy. The interaction of immersive pattern design and music barrage creates a unique music experience.



# PHASE 3: CODING STAGE

## TECHNICAL DECISION - SENTIMENT ANALYSIS

### STUDY

In the learning segment of Sentiment Analysis, we initially gathered some fundamental understanding from our classroom study, but our main reference source comes from The Coding Train's introduction to Sentiment Analysis.

In the learning process, we have the following thoughts: Sentiment Analysis is a relatively outdated technology. Dictionary-based Sentiment Analysis is far less intelligent than Machine Learning-based (LSTM) Sentiment Analysis, and it's nowhere near as efficient and practical as the state of art NLP. At the same time, dataset training requires a lot of time, like AFINN-165, and we also need to consider a lot of bias in natural language. For this project, we chose to try our best to develop a Sentiment Analysis suitable for music bars, but actually using some advanced open-source APIs would be better ([https://moderationapi.com/sentiment-analysis-api?utm\\_source=google&utm\\_medium=cpc&utm\\_id=20162073988&utm\\_content=sentiment&gclid=Cj0KCQjwmtGjBhARIsAEqfDEdopsjErpcpoQRdcBSjEvlZQ7oSuwrJnehU-zgBU6hFdH3G1aaIlgQaAtXEEALw\\_wcB](https://moderationapi.com/sentiment-analysis-api?utm_source=google&utm_medium=cpc&utm_id=20162073988&utm_content=sentiment&gclid=Cj0KCQjwmtGjBhARIsAEqfDEdopsjErpcpoQRdcBSjEvlZQ7oSuwrJnehU-zgBU6hFdH3G1aaIlgQaAtXEEALw_wcB)).

### DATASET

We compared different datasets and ultimately chose AFINN-165, because it is the most recent and relatively more comprehensive.

# PHASE 3: CODING STAGE

## TECHNICAL DECISION - SENTIMENT ANALYSIS

### BIAS PROCESSING

- The first bias we handled is the issue with negation words like "not", "no", etc. We created an array for these words. When an emotion word is detected, the system will check for any preceding negation words to ensure correct understanding of the sentence;
- The second bias we dealt with is the issue with adverbs. We considered that the application of adverbs would have different effects. For instance, "super happy" and "happy" definitely differ in degree. We created an array for adverbs to magnify the degree of emotion;
- The third bias we addressed is the issue of contrast. Generally, when people use words like "but", "however" and other contrasting terms, the focus of the sentiment is likely on the latter part. Hence, we made some adjustments here to better capture the sentiment.
- There are many biases that we did not consider, such as the use of slang and the sarcastic meaning of sentences. Of course, Natural Language Processing (NLP) is a very large project and we can't delve into too much depth. It requires the involvement of techniques like transfer learning.

# PHASE 3: CODING STAGE

## TECHNICAL DECISION - SENTIMENT ANALYSIS

```

//Variables relevant to Wine page
let wine_poor = [];
let wine_poor_name = ['Manhattan', 'Gin_Tonic', 'Absinthe'];
let wine_poor_desc = [
    "Combine dry and sweet vermouth to make the perfect Manhattan cocktail. Drink it and let everything bad go with the wind!", 
    "Classic and easy, the gin and tonic is light and refreshing. It is a simple mixed drink that requires just the two named ingredients and a hint of lime. Drink it and start afresh!"]
let wine_fair = [];
let wine_fair_name = ['Bloody_Mary', 'Tequila', 'Dry_Martini'];
let wine_fair_desc = [
    "It is one of the world best known cocktails, prized for its ability to cheer you up. With the addition of heat sources, the overall flavor is savory and spicy.", 
    "Tequila is smooth, sweet, and fruity. Its unique mild spicy taste can release you!", 
    "A chilled gin martini served up in a graceful cocktail glass is one of the most elegant and sophisticated drinks around. Spicy taste, high degree of wine, have the effect of relaxation."];
let wine_average = [];
let wine_average_name = ['Screwdriver', 'Michelada', 'Margarita'];
let wine_average_desc = [
    "Mix freshly squeezed orange juice with vodka and Angostura bitters for a classic screwdriver cocktail. Bring the taste of summer holidays to your smooth life!", 
    "Cold lager, chilli powder, pepper and lime to spice up your lager with this Mexican cocktail, great for a summer party!", 
    "Cool off this summer with our simple frozen margarita recipe. This refreshing tequila cocktail can make you feel good and comfortable!"];
let wine_good = [];
let wine_good_name = ['Mojito', 'Red_Wine', 'Cosmopolitan'];
let wine_good_desc = [
    "Mix this classic cocktail for a party using fresh mint, white rum, sugar, zesty lime and cooling soda water. Enjoy more leisure time!", 
    "Red wine has a sweet, dark flavor. It has the perfect balance of sour, bitter, and sweet elements. You can enjoy it at anytime!", 
    "Lipsmackingly sweet and sour, the Cosmopolitan cocktail of vodka, cranberry, orange liqueur and citrus is a good-time in a glass. Perfect for relax!"];
let wine_excellent = [];
let wine_excellent_name = ['Alesander', 'Baileys', 'Pina_Colada'];
let wine_excellent_desc = [
    "It is a blend of amaretto, cream, lemon curd and vanilla, finished with flowers. You can enjoy a springtime twist on a brandy alexander with Alexander cocktail!", 
    "It is a creamy liqueur, made with Irish whiskey and cream. It is rich and sweet. It is able to bring much more joy to your life!", 
    "With rum, coconut milk and pineapple juice as the main ingredients, add ice to mix and stir, taste rich and sweet. You can have a wonderful time!"];
let wine_image_all = [wine_poor_name, wine_fair_name, wine_average_name, wine_good_name, wine_excellent_name];
let wine_name_all = [wine_poor_desc, wine_fair_desc, wine_average_desc, wine_good_desc, wine_excellent_desc];
let wine_desc_all = [wine_poor_desc, wine_fair_desc, wine_average_desc, wine_good_desc, wine_excellent_desc];
let idx = 0;
let saveButton;
let arrow_left;
let arrow_right;
let byeTextarea;
let desc_God;

```

```

function page_4_furnish(){
    let image_God;
    let nameGod;

    if (averageScore <= 0.2) {
        image_God = wine_image_all[0];
        nameGod = wine_name_all[0];
        desc_God = wine_desc_all[0];
    } else if (averageScore > 0.2 && averageScore <= 0.4) {
        image_God = wine_image_all[1];
        nameGod = wine_name_all[1];
        desc_God = wine_desc_all[1];
    } else if (averageScore > 0.4 && averageScore <= 0.6) {
        image_God = wine_image_all[2];
        nameGod = wine_name_all[2];
        desc_God = wine_desc_all[2];
    } else if (averageScore > 0.6 && averageScore <= 0.8) {
        image_God = wine_image_all[3];
        nameGod = wine_name_all[3];
        desc_God = wine_desc_all[3];
    } else if (averageScore > 0.8) {
        image_God = wine_image_all[4];
        nameGod = wine_name_all[4];
        desc_God = wine_desc_all[4];
    }
}

```

```

/*
Code Referencing and Credit: (ChatGPT,2023)
"barMoodAdverbs" and "barMoodConjunctions" are word arrays
we created for the SA model to address biases and better detect
the semantic meaning of users. These phrases were derived from user testing
of our simplified music bar scenario simulation and ChatGPT's
generated responses.

Here barMoodNegatives didn't write words like doesn't,
don't, the reason is because there is no time left and it needs more process
*/
let barMoodAdverbs = [
    "absolutely", "amazingly", "astonishingly", "awfully", "completely",
    "deeply", "enormously", "especially", "exceptionally", "extraordinarily",
    "extremely", "fairly", "fantastically", "fully", "greatly",
    "highly", "hugely", "immensely", "incredibly", "intensely",
    "particularly", "perfectly", "positively", "pretty", "profoundly",
    "quite", "rather", "really", "remarkably", "seriously",
    "significantly", "so", "super", "totally", "truly",
    "unusually", "utterly", "very", "wonderfully", "madly",
    "wildly", "insanely", "crazily", "ridiculously", "freakishly",
    "hysterically", "dramatically", "outrageously", "excessively", "hellishly",
    "majorly", "unbelievably", "staggeringly", "mind-blowingly", "blindingly",
    "breathtakingly", "sickeningly", "alarmingly", "soberingly", "drunkenly",
    "tipsily", "buzzingly", "booze-fueled", "liquor-lubricated", "beer-soaked"
];

let barMoodConjunctions = [
    "however", "but", "yet", "although", "though", "tho",
    "nevertheless", "nonetheless", "still", "despite", "in contrast",
    "on the other hand", "conversely", "whereas", "instead", "in spite of",
    "conversely", "on the contrary", "nonetheless", "regardless", "even so"
];

let barMoodNegatives = ["never", "no", "not", "neither"];

```

```

function heyPage4(){
    currentPage = 4;

    scoredwords = [];
    total_score = 0;
    let inputText = sa_input.value();
    wordsArray = inputText.split(/\W/);

    // 首先找带转折词的句子
    for (let i = 0; i < wordsArray.length; i++) {
        var word = wordsArray[i].toLowerCase();
        if (barMoodConjunctions.includes(word)) {
            // 如果找到带转折词，就只保留转折词及其后面的部分
            wordsArray = wordsArray.slice(i);
            break; // 找到转折词后就结束遍历
        }
    }

    for (let i=0; i < wordsArray.length; i+=1){
        var word = wordsArray[i].toLowerCase();
        if (afinn165.hasOwnProperty(word)) {
            var score = map(afinn165[word],-5, 5, 0, 1);
            // 检查前一个单词是否存在
            if(i > 0) {
                var prevword = wordsArray[i-1].toLowerCase();
                // 第一个条件，检查前一个单词是否在barMoodAdverbs列表中
                if(barMoodAdverbs.includes(prevword)) {
                    score *= 2; // 加倍
                }
                // 第二个条件，检查前一个单词是否是 'not' 或者 'no'
                if(barMoodNegatives.includes(prevword)) {
                    score *= -1; // 变为相反数
                }
            }
            total_score += Number(score);
            scoredwords.push(word + ":" + score);
        }
    }
    averageScore = total_score / scoredwords.length;
}

```

## MOOD CALCULATION

The way we calculate sentiment is by mapping values from -5 to 5 to a 0-1 scale. We then tally up the total score for all the emotion words in a sentence and divide by the number of emotion words to derive an average score. We divide this average score into 5 intervals: less than 20% good, 20-40% good, 40-60% good, 60-80% good, and more than 80% good. Depending on the range in which the sentiment score falls, we recommend one of 15 corresponding mood drinks.

# PHASE 3: CODING STAGE

## BULLET CHAT DESIGN - TRIGGER

We have updated the UI for our dan mu (bullet comments) design compared to the A1 version, providing users with a more immersive experience. Regrettably, we didn't have time to customize preset danmu for users, we only managed to design the relevant UI.

Danmu connects to the server via MQTT. The communication is one-way, as the mobile end does not receive any responses from the server. In terms of the danmu font, we will choose more attractive fonts, colors, and quantity in the future. However, due to time constraints, we haven't managed to implement more aesthetic designs for the dan mu.

Danmu can trigger three events. When the sentence sent by the user contains 'love', it will trigger a heart event; when the sentence contains 'nice', it will trigger a wine glass collision event; when the sentence sent by the user contains 'happy', fireworks effects will appear. When writing the code, we chose a strategy that combines images and native code, writing as much code as possible when time allows, and using images when pressed for time. Here, we implemented the fireworks using code.

```
function page_2_furnish(){
  image(page_2_img, width/2, height/2);
  textSize(60);
  fill(255,70);
  textAlign(CENTER);
  text('Fly your text on the screen',width/2,0.08*height);
  fill(255,90);
  text("Hello",width/2, 0.35*height);
  text("你好", 0.18*width, 0.21*height);
  text("Hola",0.85*width, 0.27*height);
  textSize(25);
  text("안녕하세요", 0.05*width, 0.45*height);
  text("Bonjoun", 0.3*width, 0.5*height);
  text("こんにちは",0.83*width,0.47*height);
  image(circle, width/2, 0.35*height,0.4*width,0.4*width);
  image(circle,0.18*width,0.21*height,0.24*width,0.24*width);
  image(circle,0.05*width,0.45*height,0.25*width,0.25*width);
  image(circle,0.3*width,0.5*height,0.15*width,0.15*width);
  image(circle, 0.85*width, 0.27*height,0.25*width,0.25*width);
  image(circle,0.83*width,0.47*height,0.25*width,0.25*width);
  bullet_input.show();
  page_2_button.show();
}
```

```
//Code Referencing: (Luke,2022), some parts of code are from week 4 example
function drawO () {
  clear();
  //此处为文字显示及其属性/
  push();
  for (var i = fruits.length - 1; i > 0; i--) {
    var msg = fruits[i];
    msg.x -= 2; // 控制屏幕移动速度 每帧 -2
    //此处为圆圈完全重叠屏幕，从数组中移除///
    if (msg.x + textWidth(msg.textMessage) < 0) {
      fruits.splice(i, 1);
    } else {
      textSize(40);
      fill(random(255), random(255), random(255));
      text(msg.textMessage, msg.x, msg.y);
    }
  }
  pop();
}

if (showLoveImage) {
  image(lovingImg,loveX,loveY>windowWidth,lovingImg.height * (windowWidth / lovingImg.width));
  loveY -= 2;
  if (loveY < -lovingImg.height) showLoveImage = false;
}

updateFireworks();
```

```
if (cupX>windowWidth / 2 - 20 || cupX>windowWidth / 2 + 20) {
  cupX += windowWidth / 2 / (3 * 60); // Move to center in 3 seconds
  cup2X -= windowWidth / 2 / (3 * 60); // Move to center in 3 seconds
} else {
  if (imgFade > 0) {
    imgFade -= 255 / (1 * 60); // Fade out in 3 seconds
  }
  if (ellipseSize < 2 * windowWidth) {
    ellipseSize += windowWidth / (1 * 60); // Grow to windowWidth in 3 seconds
    ellipseFade += 255 / (1 * 60); // Fade in in 3 seconds
  } else {
    if (ellipseFade > 0) {
      ellipseFade -= 255 / (1 * 60); // Fade out in 3 seconds
    }
  }
}
```

```
//Code Referencing: (Luke,2022), The addNewMessage() is from the example of IDE9101 WS.
//关键词触发功能////
function addNewMessage(newMessage) {
  console.log("ADDING NEW MSG");
  fruits.push(newMessage);
}

if (newMessage.textMessage.toLowerCase().includes("happy")) {
  console.log("Happy detected in message");
  //////////////////////////////////////////////////////////////////
  createFireworks(windowWidth / 2, windowHeight / 2, 1000); // 创建烟花特效

  for (let Fire = 0; Fire < 8; Fire++) {
    fill(random(255), random(255), random(255));
    createFireworks(
      random(windowWidth),
      random(windowHeight),
      random(10, 500)
    ); // 创建烟花特效
  }
} else if (newMessage.textMessage.toLowerCase().includes("love")) {
  //////////////////////////////////////////////////////////////////
  loveY = windowHeight;
  showLoveImage = true;
} else if (newMessage.textMessage.toLowerCase().includes("nice")) {
  //////////////////////////////////////////////////////////////////
  showGlasses = true;
  cup1X = 0;
  cup2X = windowWidth;
  cup1Y = cup2Y = windowHeight / 2;
  imgFade = 255;
  ellipseSize = 0;
  ellipseFade = 0;
}

//维护 allMessages 数组的长度，确保其长度不会超过 MAX_NUM_MESSAGES 这个设定的最大消息数量////
if (allMessages.length > MAX_NUM_MESSAGES) {
  allMessages.shift();
}
```

# PHASE 3: CODING STAGE

## TECHNICAL DECISION - SERVER AUDIO VISUALIZATION ANIMATION TRIGGERS

Regarding the server design, we opted for a visual display and animation effects based on HTML and CSS because we considered that P5.JS, being a library intended for beginners, has many inconveniences.

We previously considered creating visually appealing 3D effects with spline and Three.js, and even thought about creating generative art backgrounds to continuously stimulate the audience's attention, but due to time constraints, learning requirements, and the web app's demand for lightweight storage, we temporarily abandoned these plans.

In the future, we plan to introduce lyrics to accommodate the audience's need to view song lyrics.

As for the music visualization part, due to time constraints, we've only created stars that respond to the rhythm of the music, in order to match our chosen theme background and to pursue aesthetic effects as much as possible. We hope to create better music visualization designs in the future.

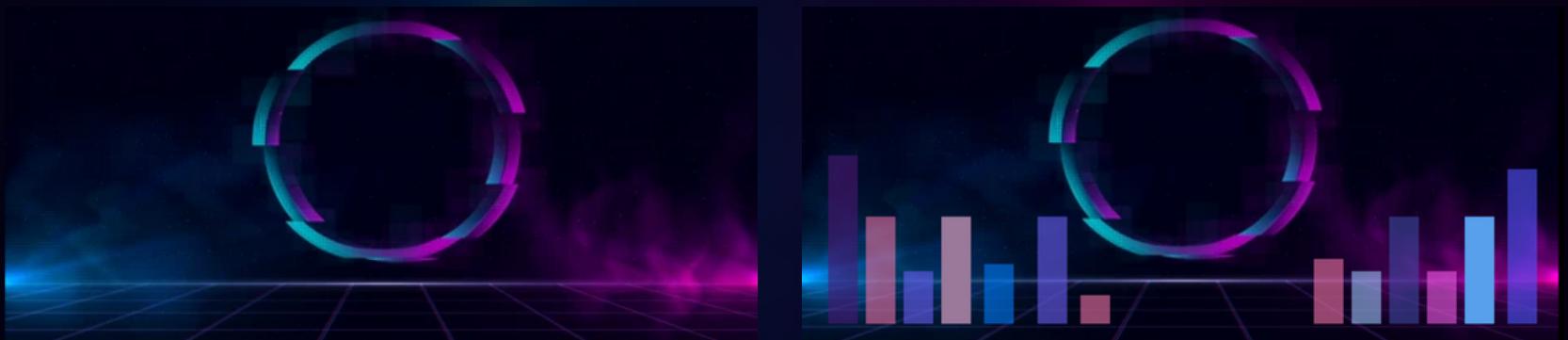
However, the overall effect currently is acceptable. It's worth noting that on our main page, there is a boy holding a star and a girl, which were generated by Midjourney. Their prompts are as shown in the picture.

```
// 计算频谱的平均幅度
let avgAmp = 0;
for (let i = 0; i < spectrum.length; i++) {
  avgAmp += spectrum[i];
}
avgAmp /= spectrum.length;

// 基于平均幅度绘制星星
for (let i = 0; i < stars.length; i++) {
  let star_size = map(avgAmp, 0, 255, 0, 30); // 增大星星的大小变化范围
  let starObj = stars[i];
  star(starObj.x, starObj.y, star_size, star_size, starObj.r, starObj.g, starObj.b);
}
```

# WHAT'S NEXT

1. Studying the rules of different browsers: It's important to understand why the same piece of code might work in one browser but not in another. This can often be due to differences in how browsers interpret or handle certain HTML, CSS, or JavaScript features. As developers, we should always be testing our code in different environments to ensure broad compatibility and the best possible user experience.
2. Leveraging more advanced NLP or Sentiment Analysis APIs: Instead of reiterating what others have already accomplished, we should be integrating more sophisticated tools into our work. This could mean using pre-built NLP or sentiment analysis APIs to extract valuable insights from user feedback or other text data. If the cost is prohibitive, another strategy could be to train a customer-specific dictionary dataset based on the GPT model, which would allow us to tailor our NLP capabilities to our specific needs.
3. Conducting more field testing: It's crucial to understand our users' needs and wants, particularly when it comes to AI technologies. By conducting more on-site tests and user interviews, we can pinpoint users' pain points more precisely. This approach will allow us to iterate on our product design more effectively and deliver a product that truly meets user needs.
4. Developing more mature app products: Rather than relying solely on basic HTML, CSS, and p5.js for app development, we should strive for more complex and mature product development. This might involve using more robust frameworks or languages, and implementing features that take full advantage of the capabilities of modern smartphones and tablets.
5. Improving server interface design: The server interface is a crucial part of any web or app product, and good design here can greatly enhance the user experience. By focusing on making our server interface not only more functional but also more visually appealing, we can further satisfy our users' needs and expectations. This could involve, for example, improving the layout, typography, or color scheme of the interface, or adding helpful features like tooltips or real-time status updates.



# APPENDIX

