WEB & CLOUD COMPUTING

MIXMASTER

ANGELINA SUCHKOVA
GUILHERME RAIMUNDO
Mª LEONOR HERMENEGILDO
TIAGO BARROS

CONTENT

01 API

02 IMPORTS

03 FUNCTIONALITIES

04 DATABASE

05 CODE HIGHLIGHTS

API

TheCocktailDB

A comprehensive collection of cocktail recipes, including details on ingredients, measurements, and instructions, designed to cater to a variety of tastes and preferences.

API that enables searching for cocktails by name, ingredient, or ID, listing cocktails by the first letter, and providing filters for categories, glasses, and alcoholic content.

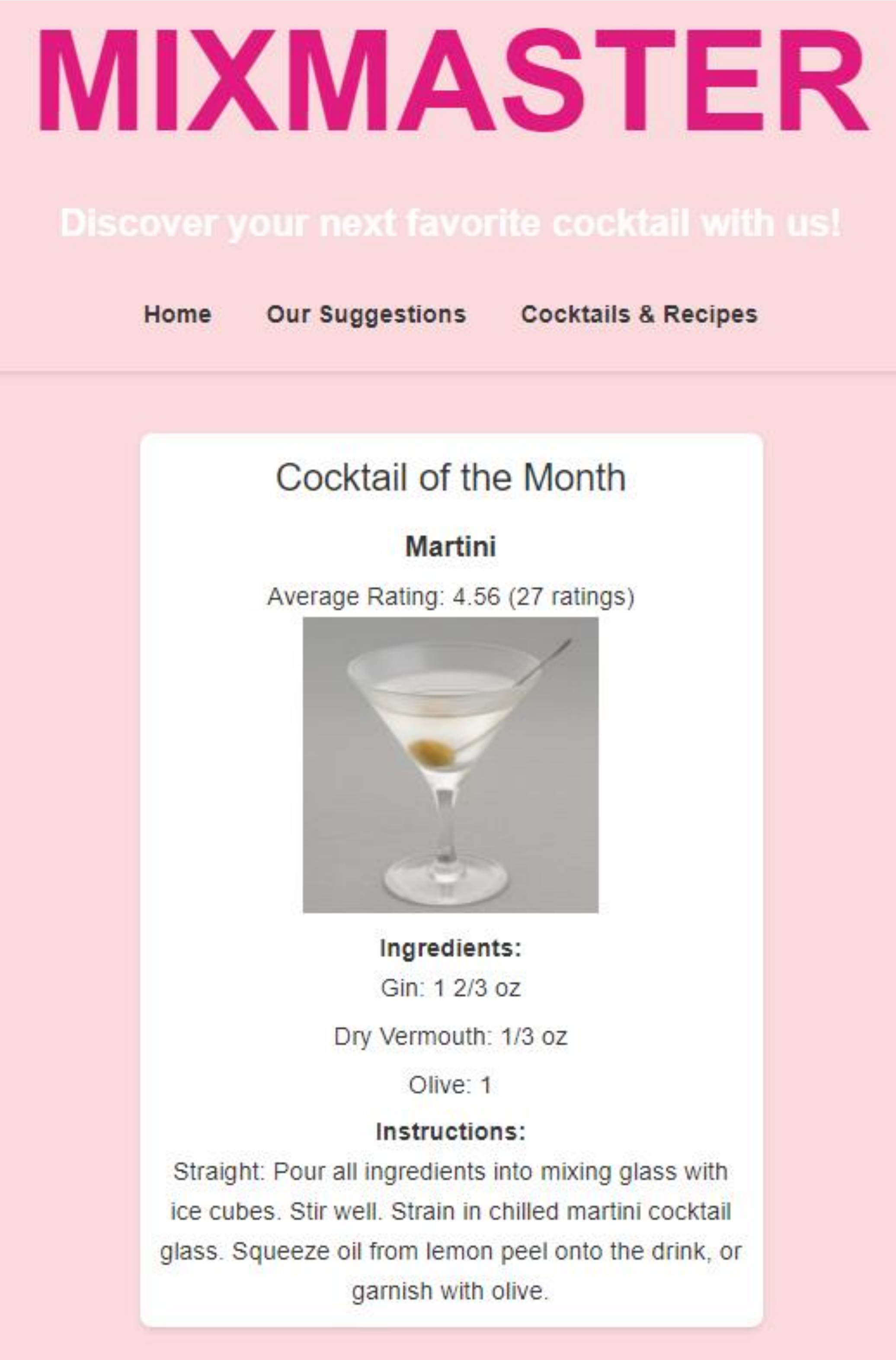
IMPORTS

Node.js is a runtime environment that allows us to run JavaScript on the server side, enabling the development of scalable network applications.

Express is a lightweight framework for Node.js that simplifies the process of building server-side applications and APIs. It provides a robust set of features to manage routes, requests, and responses, making it easier to develop web applications and RESTful services.

SQLite is a compact, reliable SQL database engine widely used worldwide. It's perfect for websites and applications, especially for development, testing, and smaller projects, offering full database functionality without needing a full-scale database system.

FUNCTIONALITIES



MIXMASTER

Discover your next favorite cocktail with us!

Cocktail of the Month

Martini

Average Rating: 4.56 (27 ratings)

Ingredients:

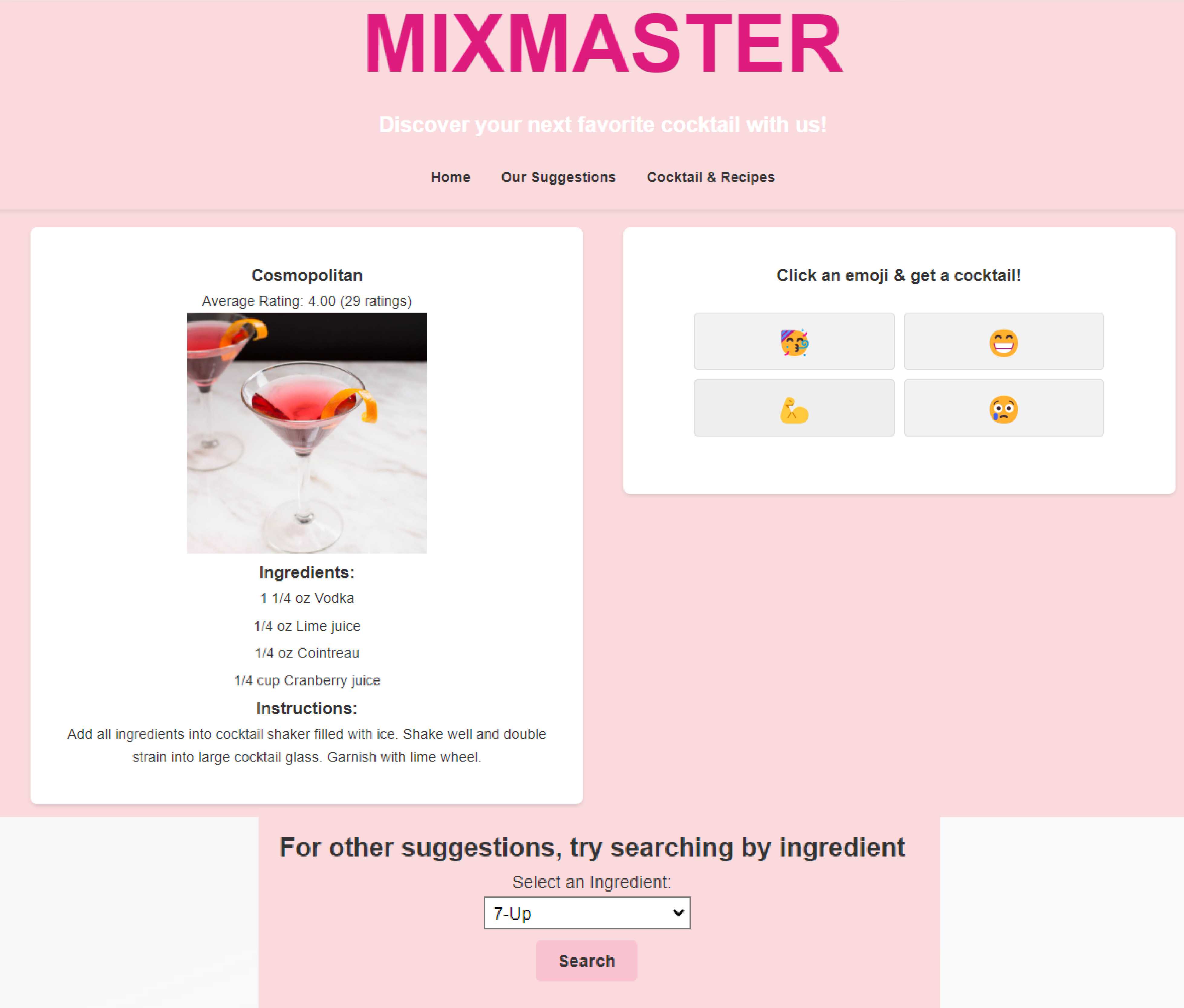
Gin: 1 2/3 oz

Dry Vermouth: 1/3 oz

Olive: 1

Instructions:

Straight: Pour all ingredients into mixing glass with ice cubes. Stir well. Strain in chilled martini cocktail glass. Squeeze oil from lemon peel onto the drink, or garnish with olive.



MIXMASTER

Discover your next favorite cocktail with us!

Cosmopolitan

Average Rating: 4.00 (29 ratings)

Ingredients:

1 1/4 oz Vodka

1/4 oz Lime juice

1/4 oz Cointreau

1/4 cup Cranberry juice

Instructions:

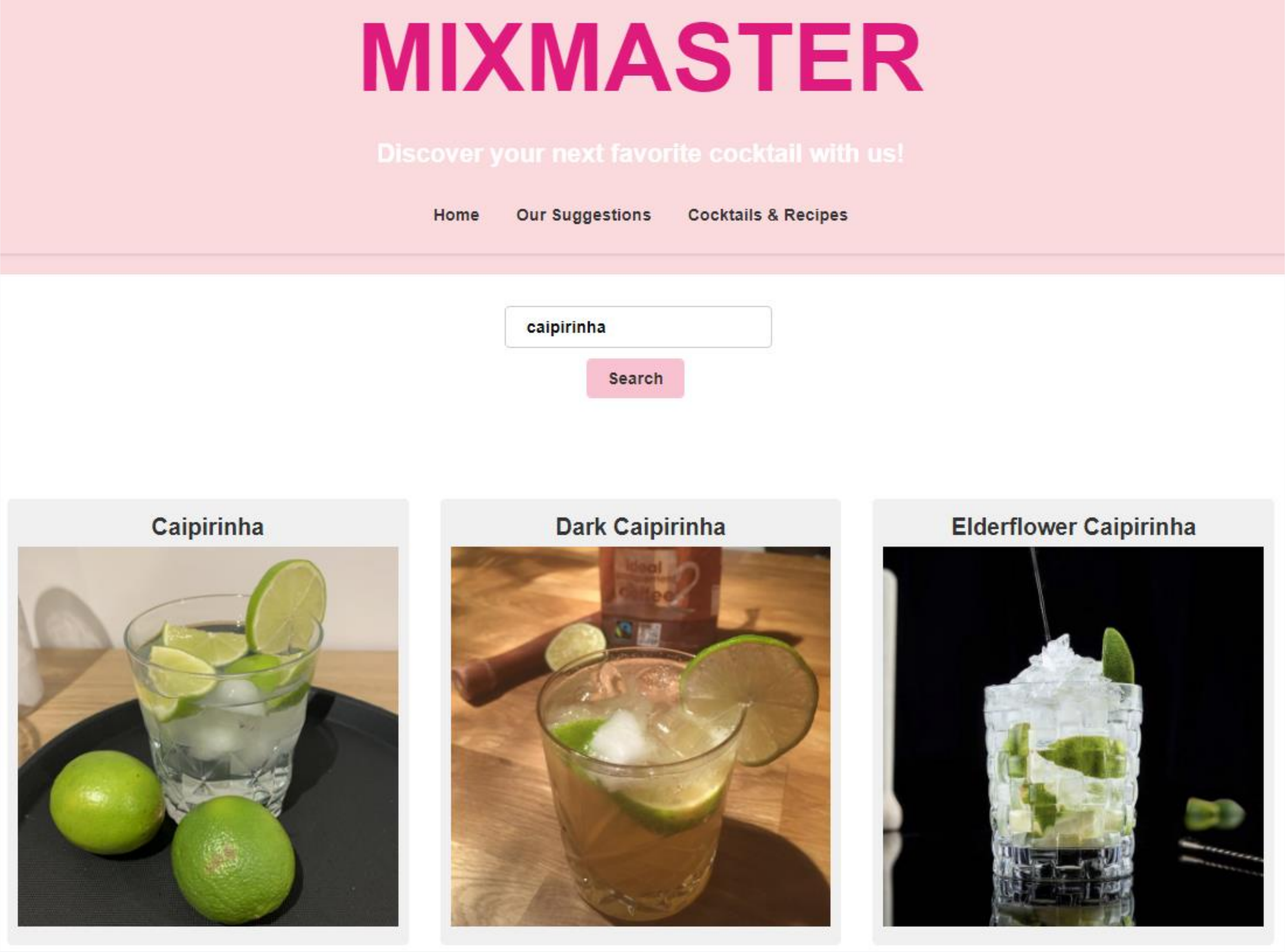
Add all ingredients into cocktail shaker filled with ice. Shake well and double strain into large cocktail glass. Garnish with lime wheel.

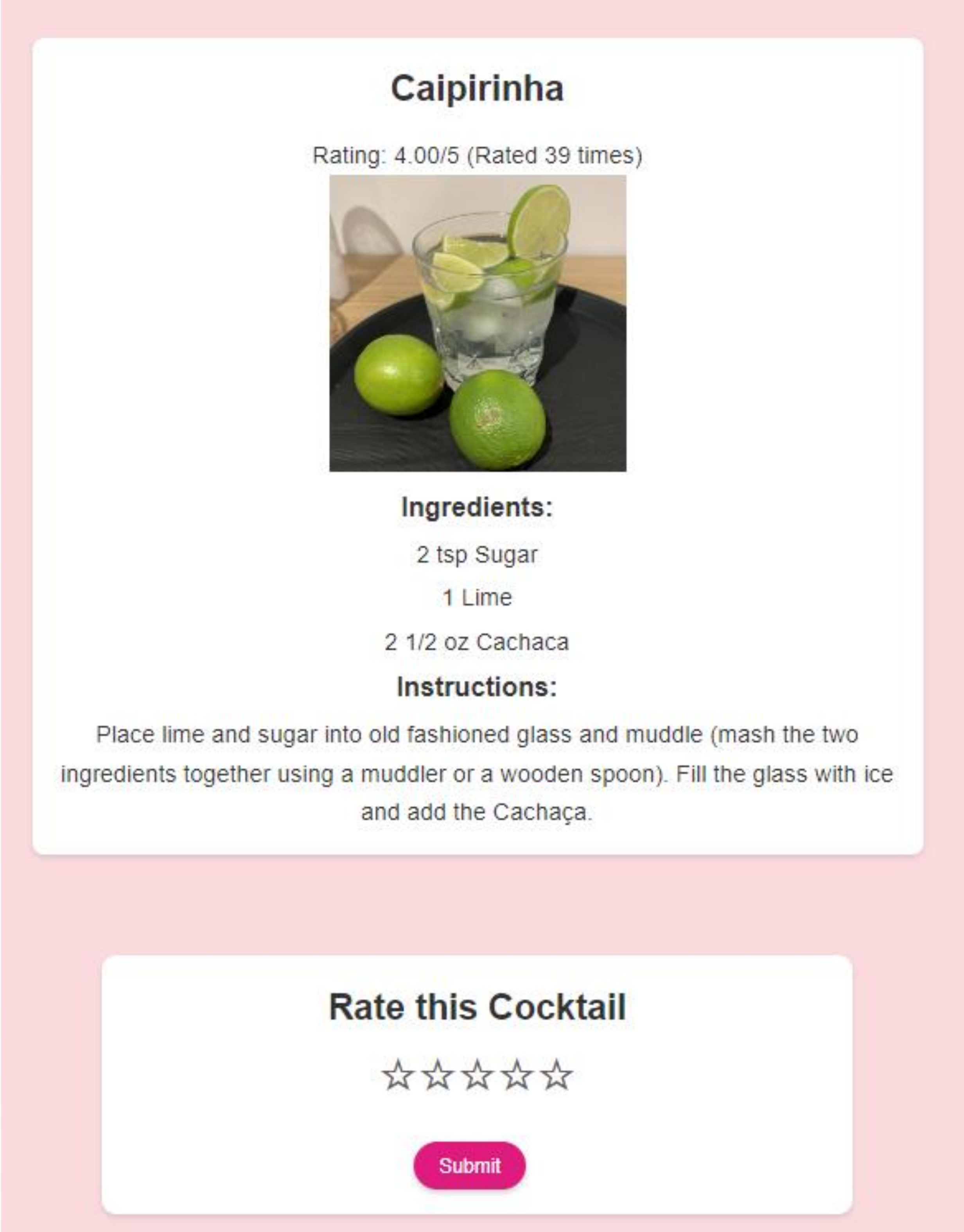
Click an emoji & get a cocktail!

For other suggestions, try searching by ingredient

Select an Ingredient: 7-Up

FUNCTIONALITIES





Caipirinha

Rating: 4.00/5 (Rated 39 times)

Ingredients:

2 tsp Sugar

1 Lime

2 1/2 oz Cachaca

Instructions:

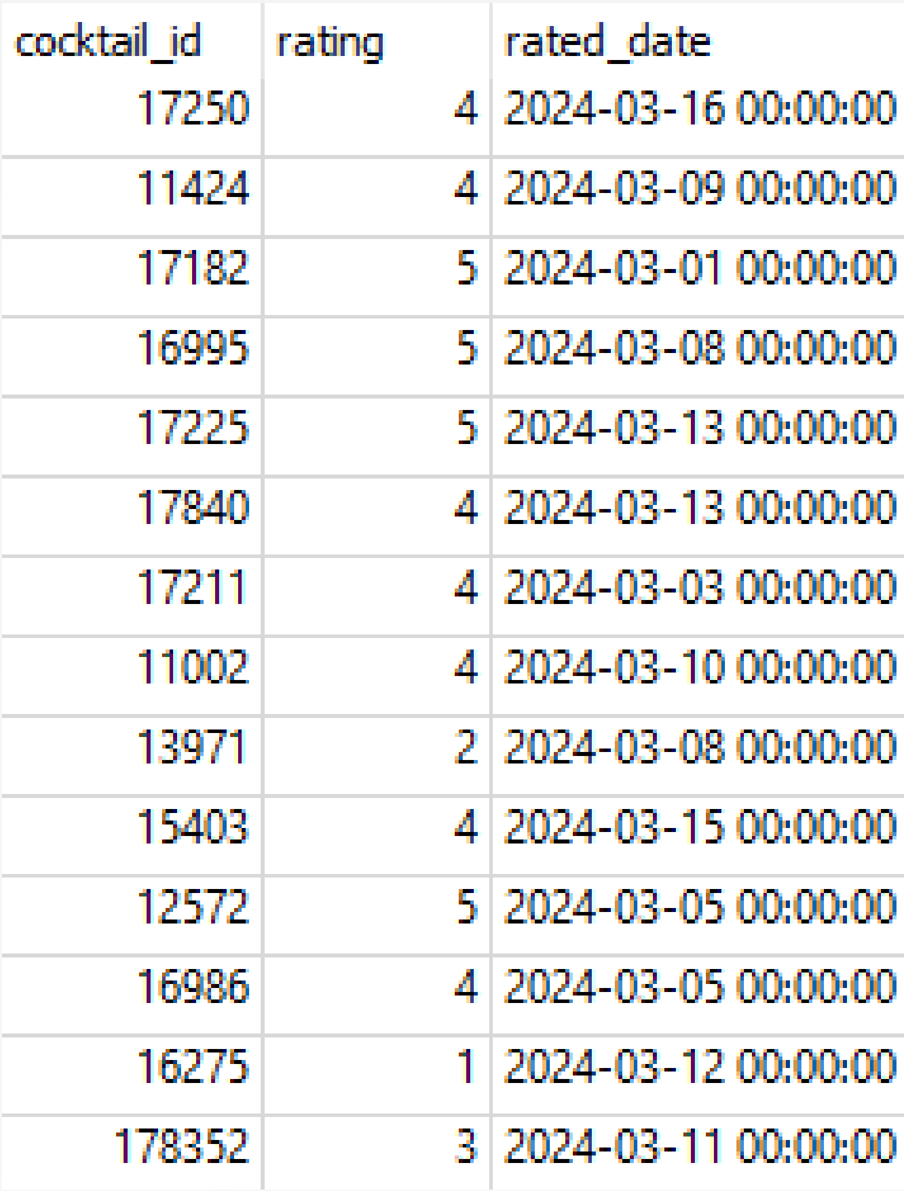
Place lime and sugar into old fashioned glass and muddle (mash the two ingredients together using a muddler or a wooden spoon). Fill the glass with ice and add the Cachaça.

Rate this Cocktail

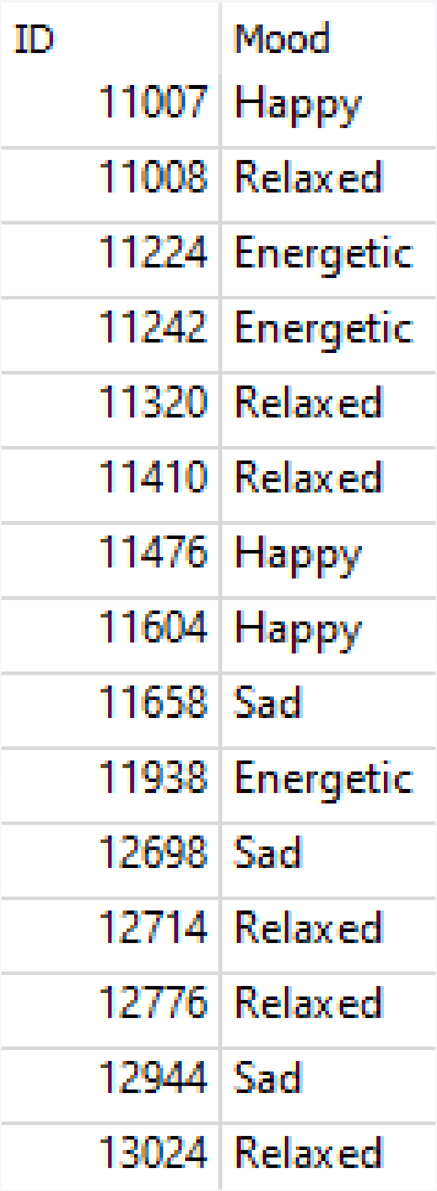
☆☆☆☆☆

DATABASE

SQLite



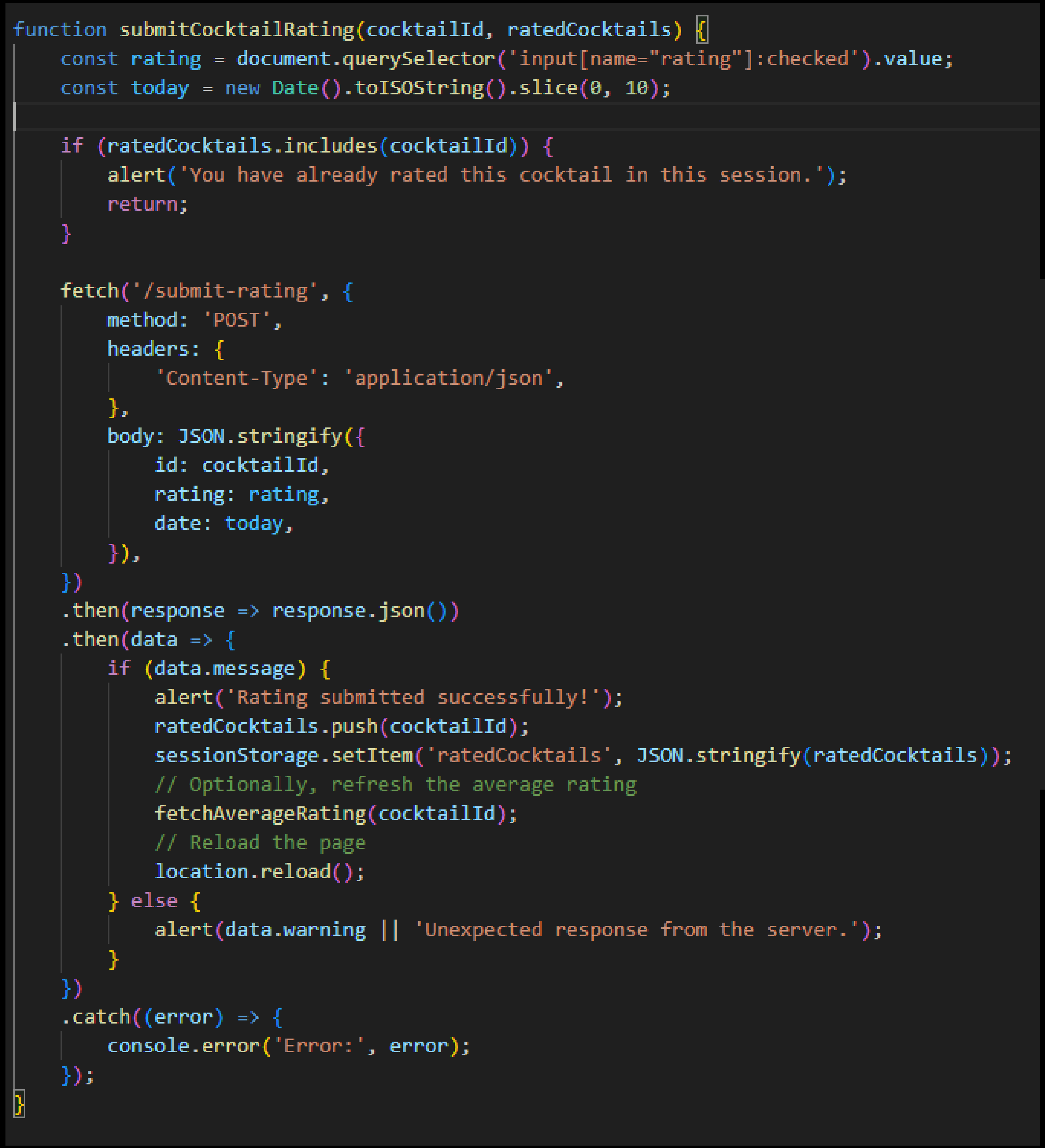
cocktail_id	rating	rated_date
17250	4	2024-03-16 00:00:00
11424	4	2024-03-09 00:00:00
17182	5	2024-03-01 00:00:00
16995	5	2024-03-08 00:00:00
17225	5	2024-03-13 00:00:00
17840	4	2024-03-13 00:00:00
17211	4	2024-03-03 00:00:00
11002	4	2024-03-10 00:00:00
13971	2	2024-03-08 00:00:00
15403	4	2024-03-15 00:00:00
12572	5	2024-03-05 00:00:00
16986	4	2024-03-05 00:00:00
16275	1	2024-03-12 00:00:00
178352	3	2024-03-11 00:00:00



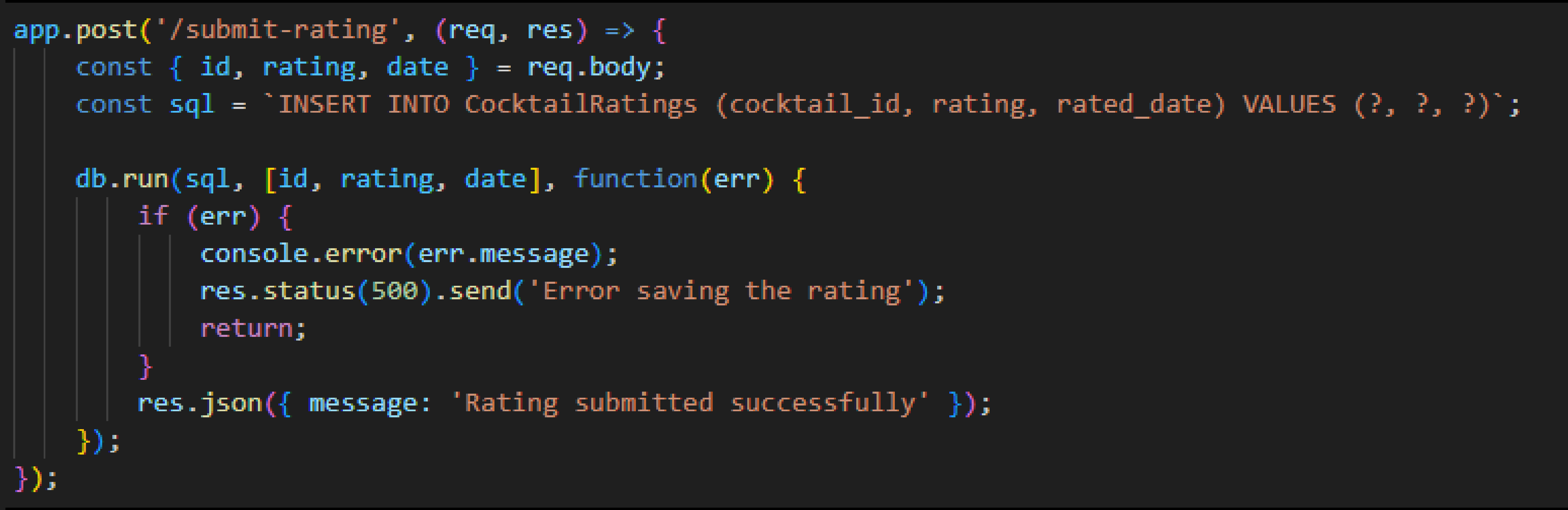
ID	Mood
11007	Happy
11008	Relaxed
11224	Energetic
11242	Energetic
11320	Relaxed
11410	Relaxed
11476	Happy
11604	Happy
11658	Sad
11938	Energetic
12698	Sad
12714	Relaxed
12776	Relaxed
12944	Sad
13024	Relaxed

CODE HIGHLIGHTS

RATING SUBMISSIONS



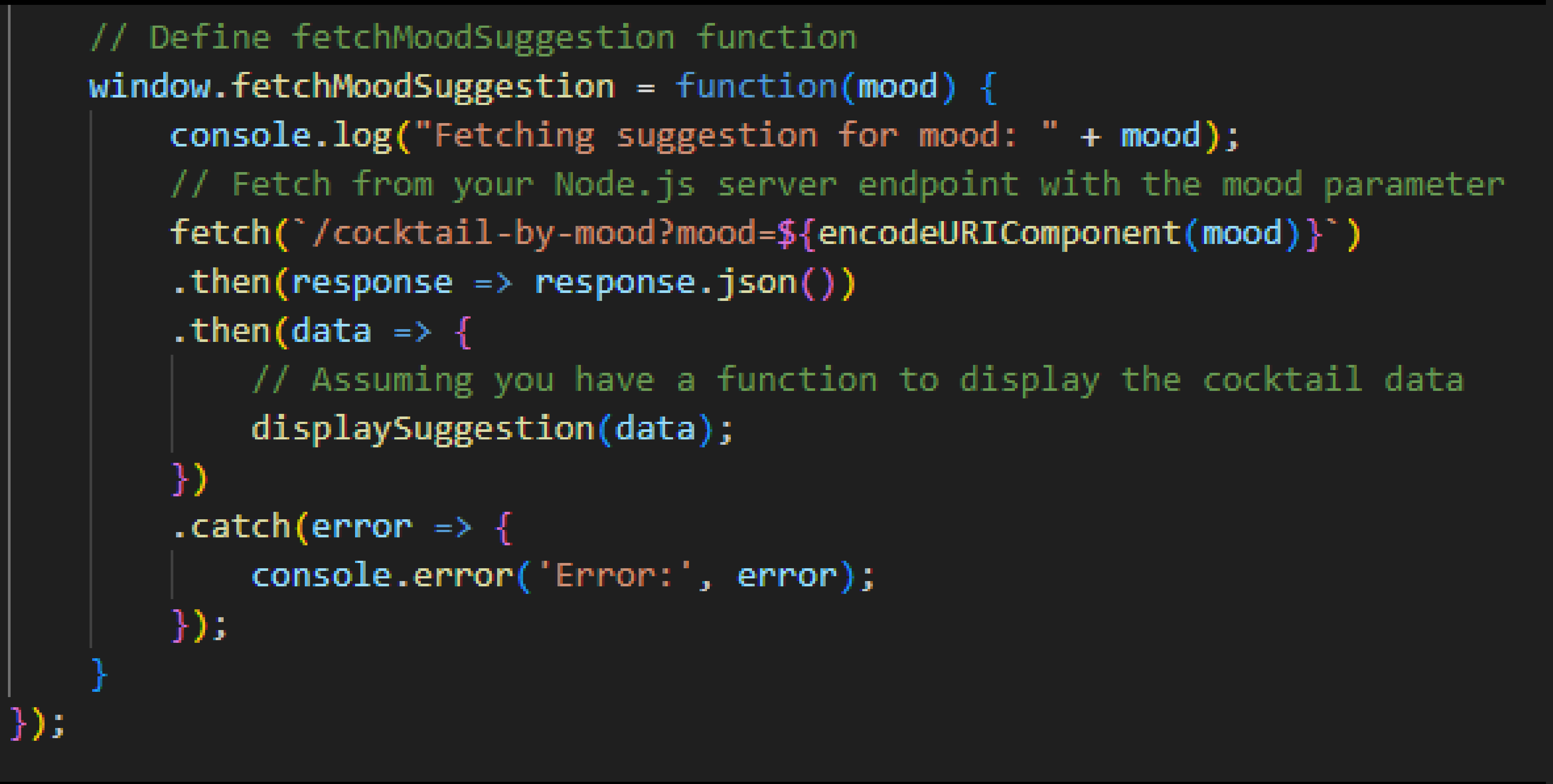
```
function submitCocktailRating(cocktailId, ratedCocktails) {
    const rating = document.querySelector('input[name="rating"]:checked').value;
    const today = new Date().toISOString().slice(0, 10);

    if (ratedCocktails.includes(cocktailId)) {
        alert('You have already rated this cocktail in this session.');
        return;
    }

    fetch('/submit-rating', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            id: cocktailId,
            rating: rating,
            date: today,
        }),
    })
    .then(response => response.json())
    .then(data => {
        if (data.message) {
            alert('Rating submitted successfully!');
            ratedCocktails.push(cocktailId);
            sessionStorage.setItem('ratedCocktails', JSON.stringify(ratedCocktails));
            // Optionally, refresh the average rating
            fetchAverageRating(cocktailId);
            // Reload the page
            location.reload();
        } else {
            alert(data.warning || 'Unexpected response from the server.');
        }
    })
    .catch((error) => {
        console.error('Error:', error);
    });
}
```



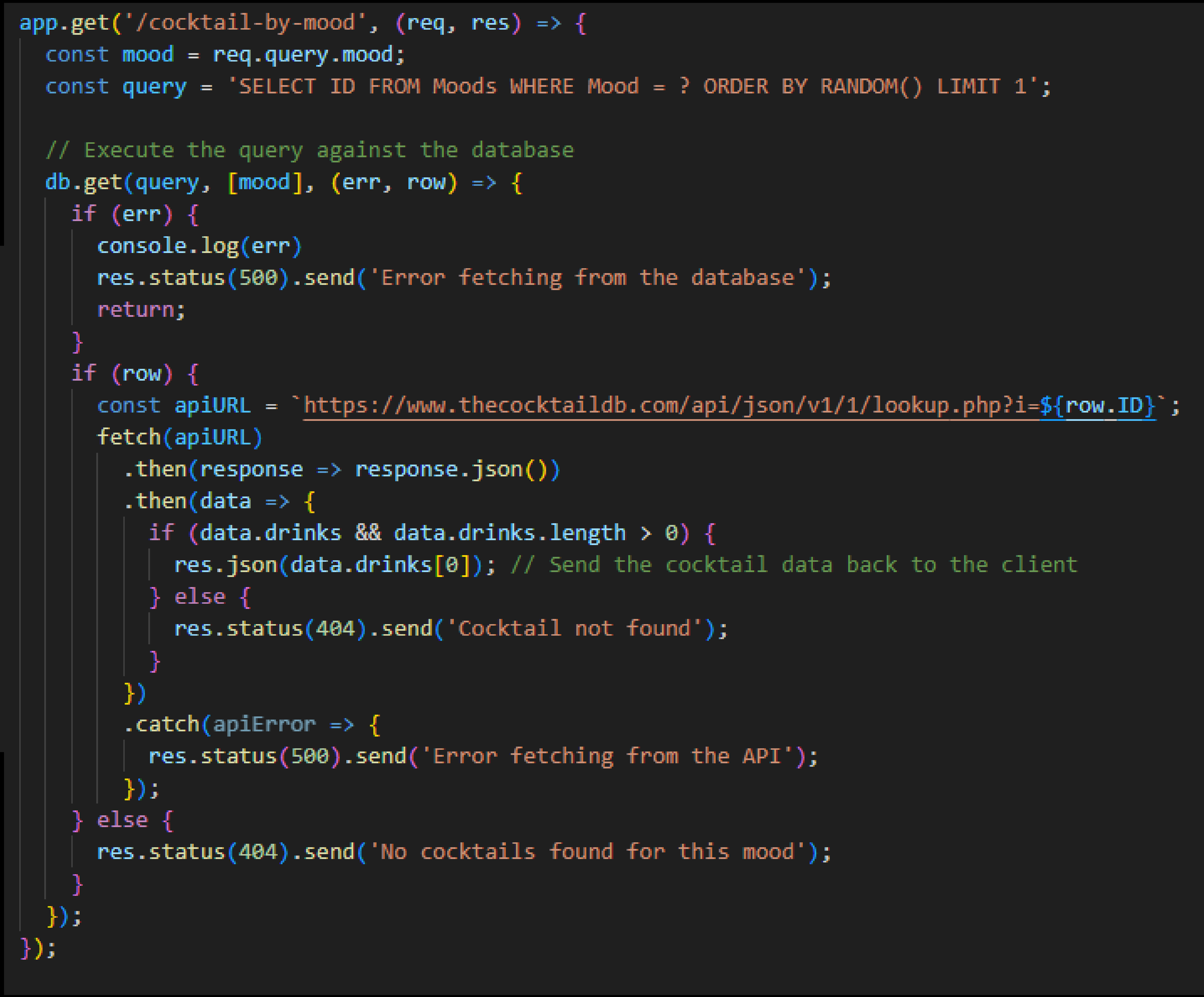
```
app.post('/submit-rating', (req, res) => {
    const { id, rating, date } = req.body;
    const sql = `INSERT INTO CocktailRatings (cocktail_id, rating, rated_date) VALUES (?, ?, ?)`;

    db.run(sql, [id, rating, date], function(err) {
        if (err) {
            console.error(err.message);
            res.status(500).send('Error saving the rating');
            return;
        }
        res.json({ message: 'Rating submitted successfully' });
    });
});
```

CODE HIGHLIGHTS

MOOD SUGGESTIONS



```
    // Define fetchMoodSuggestion function
    window.fetchMoodSuggestion = function(mood) {
        console.log("Fetching suggestion for mood: " + mood);
        // Fetch from your Node.js server endpoint with the mood parameter
        fetch(`/cocktail-by-mood?mood=${encodeURIComponent(mood)}`)
        .then(response => response.json())
        .then(data => {
            // Assuming you have a function to display the cocktail data
            displaySuggestion(data);
        })
        .catch(error => {
            console.error('Error:', error);
        });
    }
});
```



```
app.get('/cocktail-by-mood', (req, res) => {
  const mood = req.query.mood;
  const query = 'SELECT ID FROM Moods WHERE Mood = ? ORDER BY RANDOM() LIMIT 1';

  // Execute the query against the database
  db.get(query, [mood], (err, row) => {
    if (err) {
      console.log(err)
      res.status(500).send('Error fetching from the database');
      return;
    }
    if (row) {
      const apiURL = `https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=${row.ID}`;
      fetch(apiURL)
        .then(response => response.json())
        .then(data => {
          if (data.drinks && data.drinks.length > 0) {
            res.json(data.drinks[0]); // Send the cocktail data back to the client
          } else {
            res.status(404).send('Cocktail not found');
          }
        })
        .catch(apiError => {
          res.status(500).send('Error fetching from the API');
        });
    } else {
      res.status(404).send('No cocktails found for this mood');
    }
  });
});
```

THANK'S FOR WATCHING

We are ready to answer your questions!

mixmaster.com