Project Documentation

Movie and TV Show Explorer

Main Objective:

The project aims to develop a comprehensive web application incorporating various functionalities such as Dynamic Single-Page Website, API integration, and Web Socket. The application demonstrates modern web development practices with a focus on both frontend and backend technologies.

Functionalities and Features:

JavaScript Functionality: Handles and manages the main functions of the dynamic content

API Integration: The application will leverage external APIs to enhance its functionality. Specifically, it will utilize:

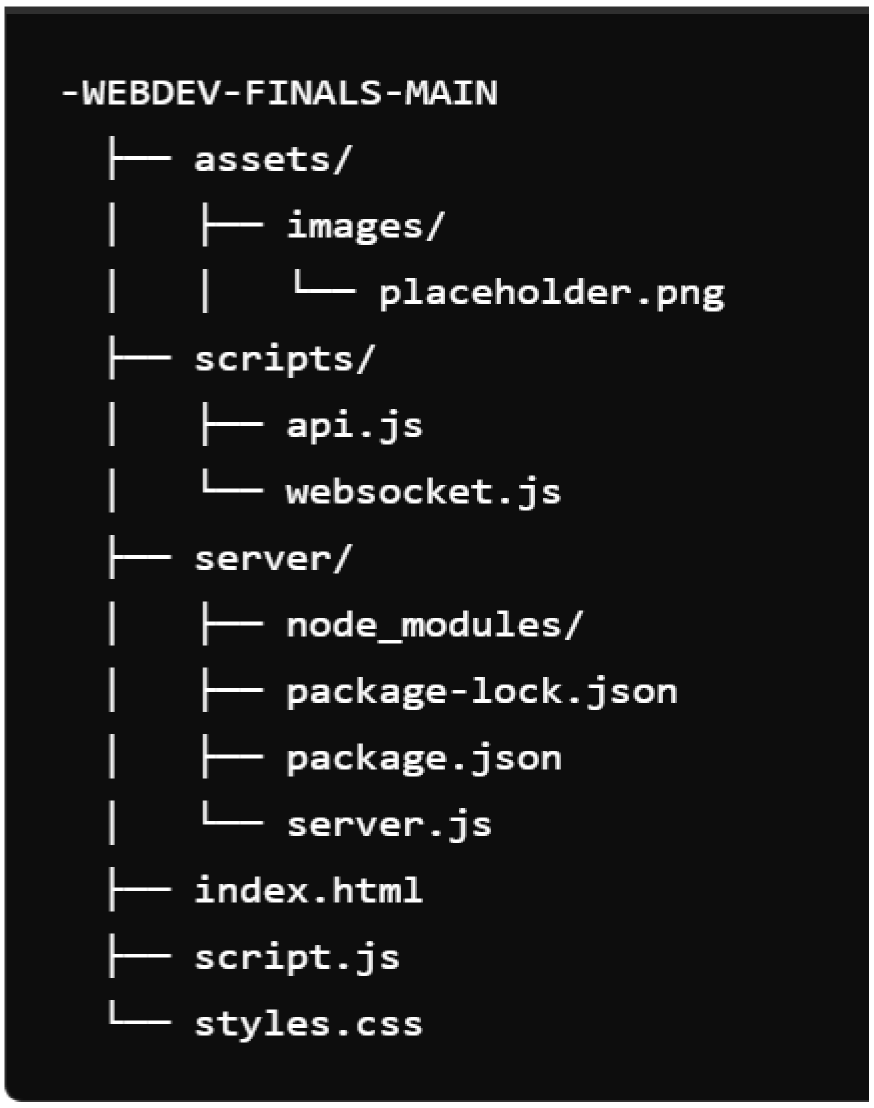
- TMDb API: To retrieve detailed information about movies.
- News API: To fetch and display news articles and updates.

Responsive Design: The application will adapt its layout and functionality to orderly display on various devices.

Web Socket Implementation: The application will employ WebSockets to enable real-time search suggestions. This will provide users with instant feedback as they type their search queries.

Dynamic Content: Website updates based on user interactions. The application will load content dynamically without full page reloads, resulting in a faster and more seamless user experience.

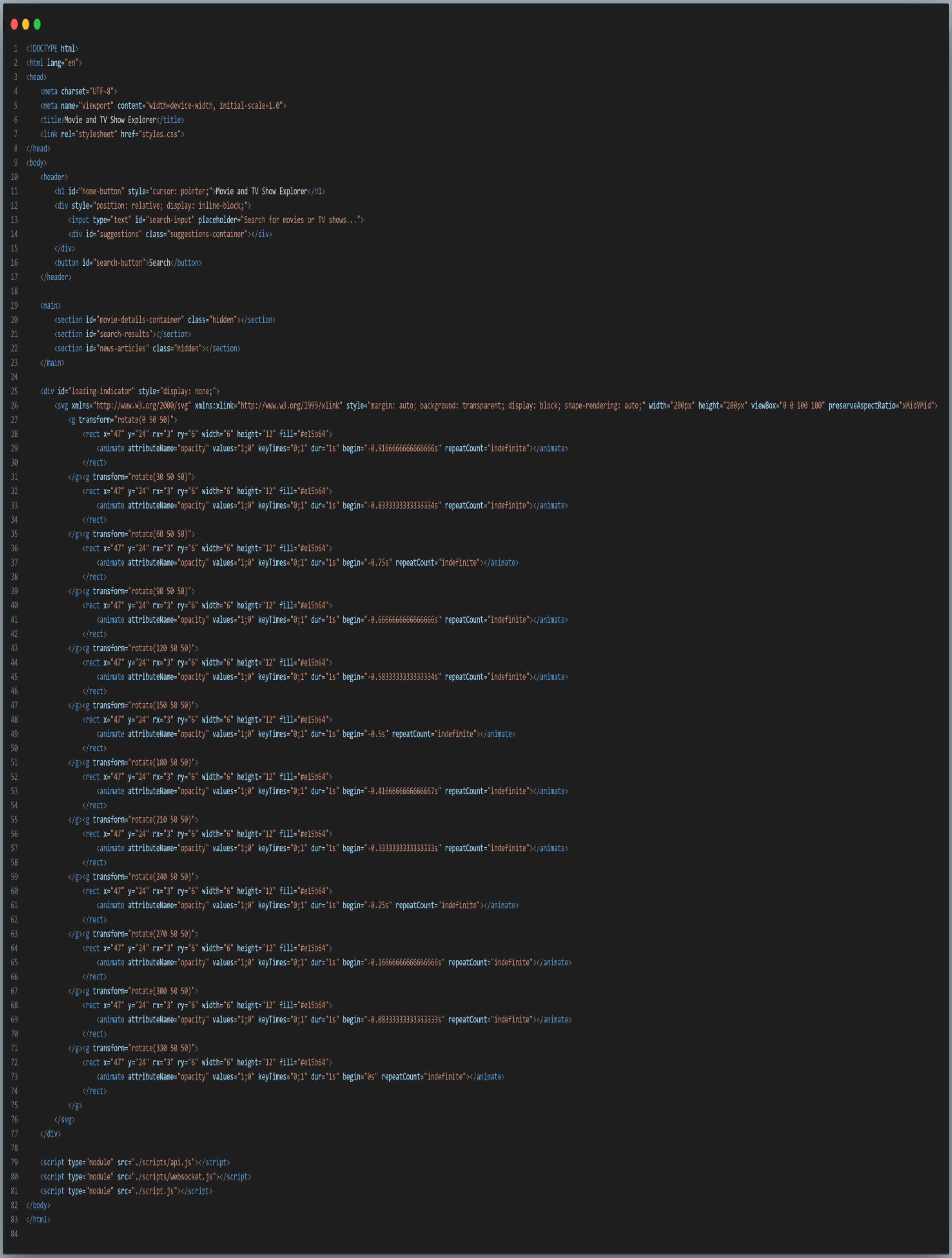
Program Structure:



```
-WEBDEV-FINALS-MAIN
  ├── assets/
  │   ├── images/
  │   │   └── placeholder.png
  ├── scripts/
  │   ├── api.js
  │   └── websocket.js
  ├── server/
  │   ├── node_modules/
  │   ├── package-lock.json
  │   ├── package.json
  │   └── server.js
  ├── index.html
  ├── script.js
  └── styles.css
```

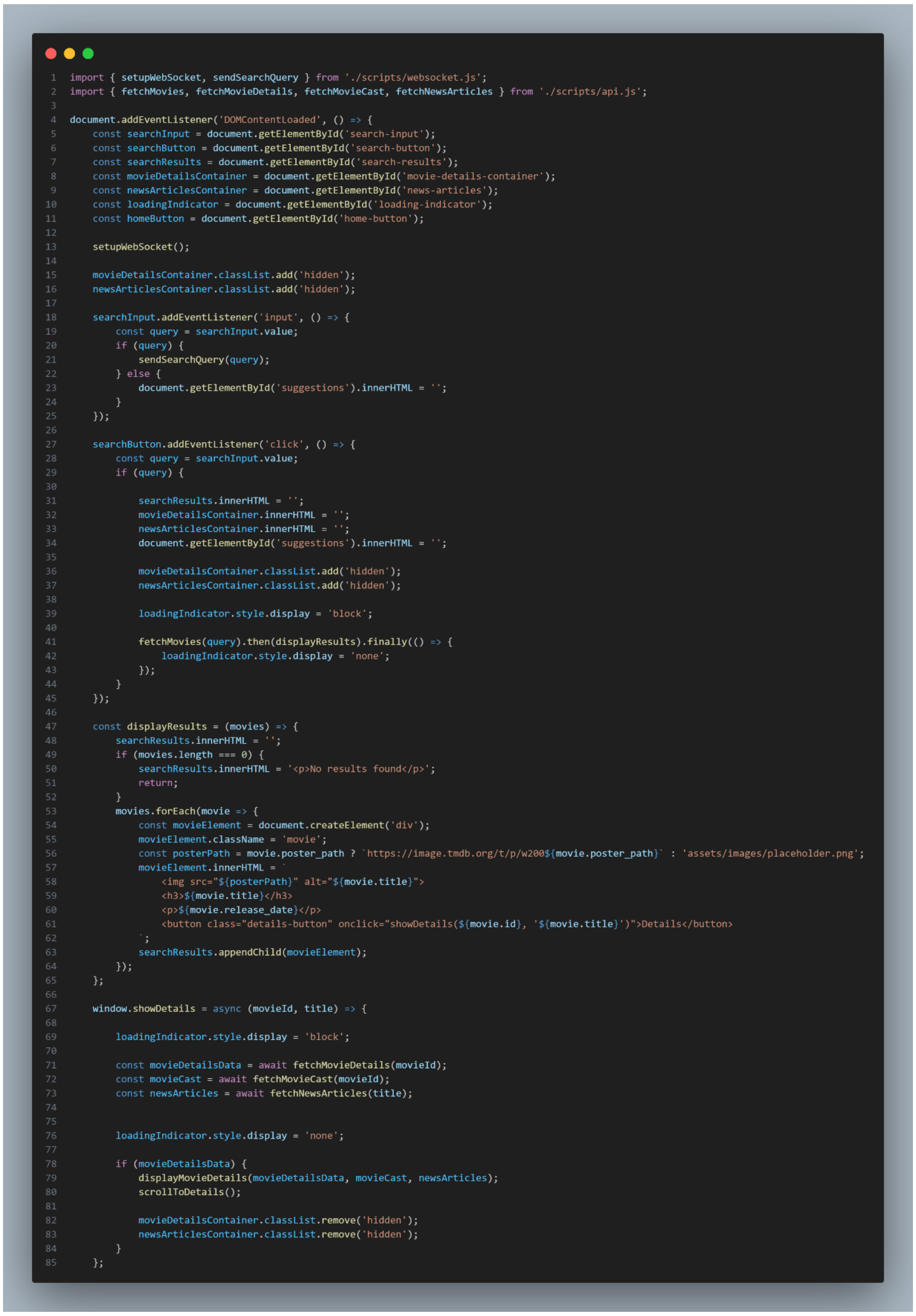
Code Documentation:

index.html

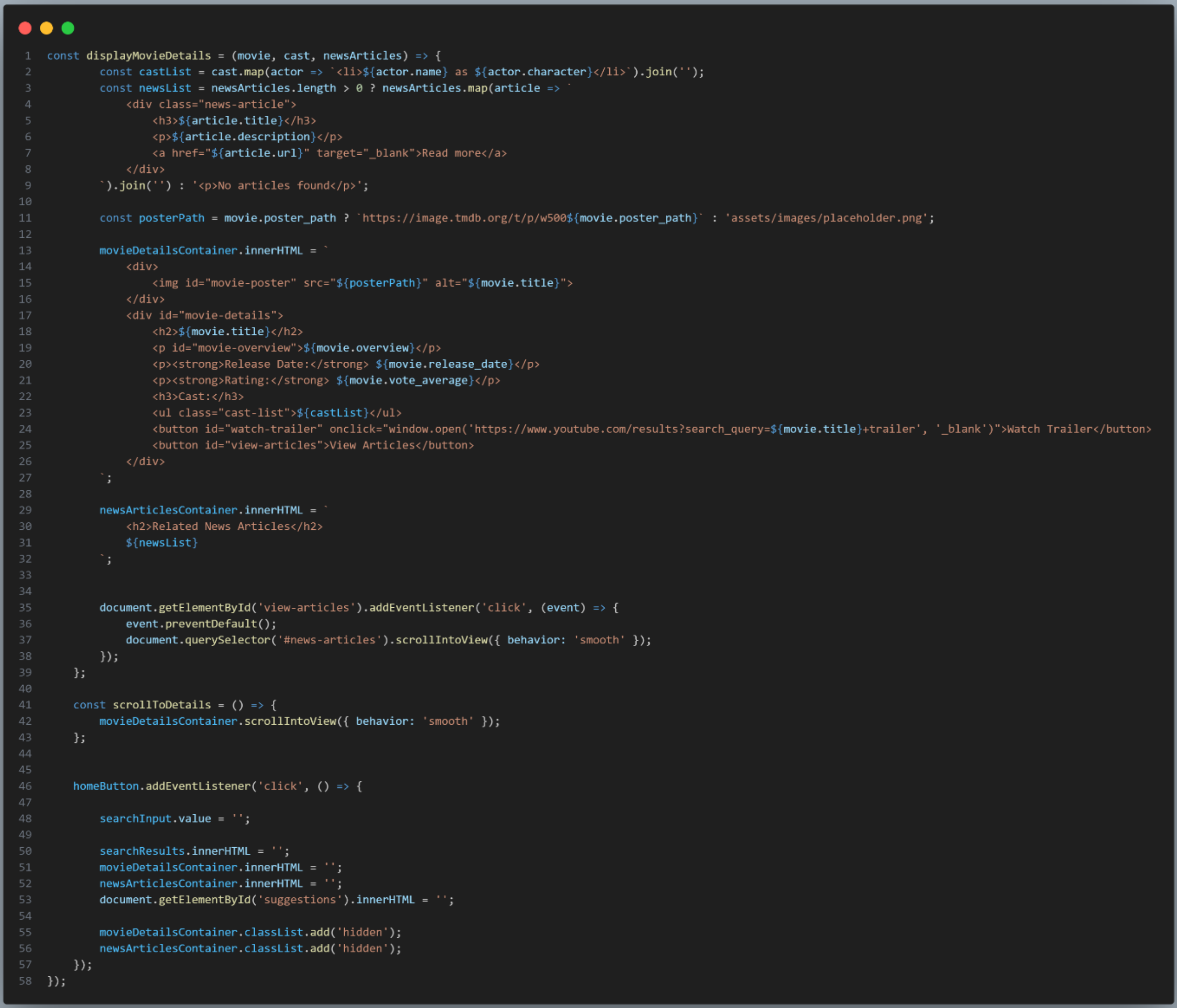


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Movie and TV Show Explorer</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1 id="home-button" style="cursor: pointer;">Movie and TV Show Explorer</h1>
        <div style="position: relative; display: inline-block;">
            <input type="text" id="search-input" placeholder="Search for movies or TV shows...">
            <div id="suggestions" class="suggestions-container"></div>
        </div>
        <button id="search-button">Search</button>
    </header>

    <main>
        <section id="movie-details-container" class="hidden"></section>
        <section id="search-results"></section>
        <section id="news-articles" class="hidden"></section>
    </main>

    <div id="loading-indicator" style="display: none;">
        <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style="margin: auto; background: transparent; display: block; shape-rendering: auto;" width="200px" height="200px" viewBox="0 0 100 100" preserveAspectRatio="xMidYMid">
            <g transform="rotate(0 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.9166666666666666s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(30 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.8333333333333334s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(60 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.75s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(90 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.6666666666666666s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(120 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.5833333333333334s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(150 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.5s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(180 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.4166666666666667s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(210 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.3333333333333333s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(240 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.25s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(270 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.16666666666666666s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(300 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="-0.08333333333333333s" repeatCount="indefinite"></animate>
                </rect>
            </g><g transform="rotate(330 50 50)">
                <rect x="47" y="24" rx="3" ry="6" width="6" height="12" fill="#e15b64">
                    <animate attributeName="opacity" values="1;0" keyTimes="0;1" dur="1s" begin="0s" repeatCount="indefinite"></animate>
                </rect>
            </g>
        </svg>
    </div>

    <script type="module" src="./scripts/api.js"></script>
    <script type="module" src="./scripts/websocket.js"></script>
    <script type="module" src="./script.js"></script>
</body>
</html>
```

script.js

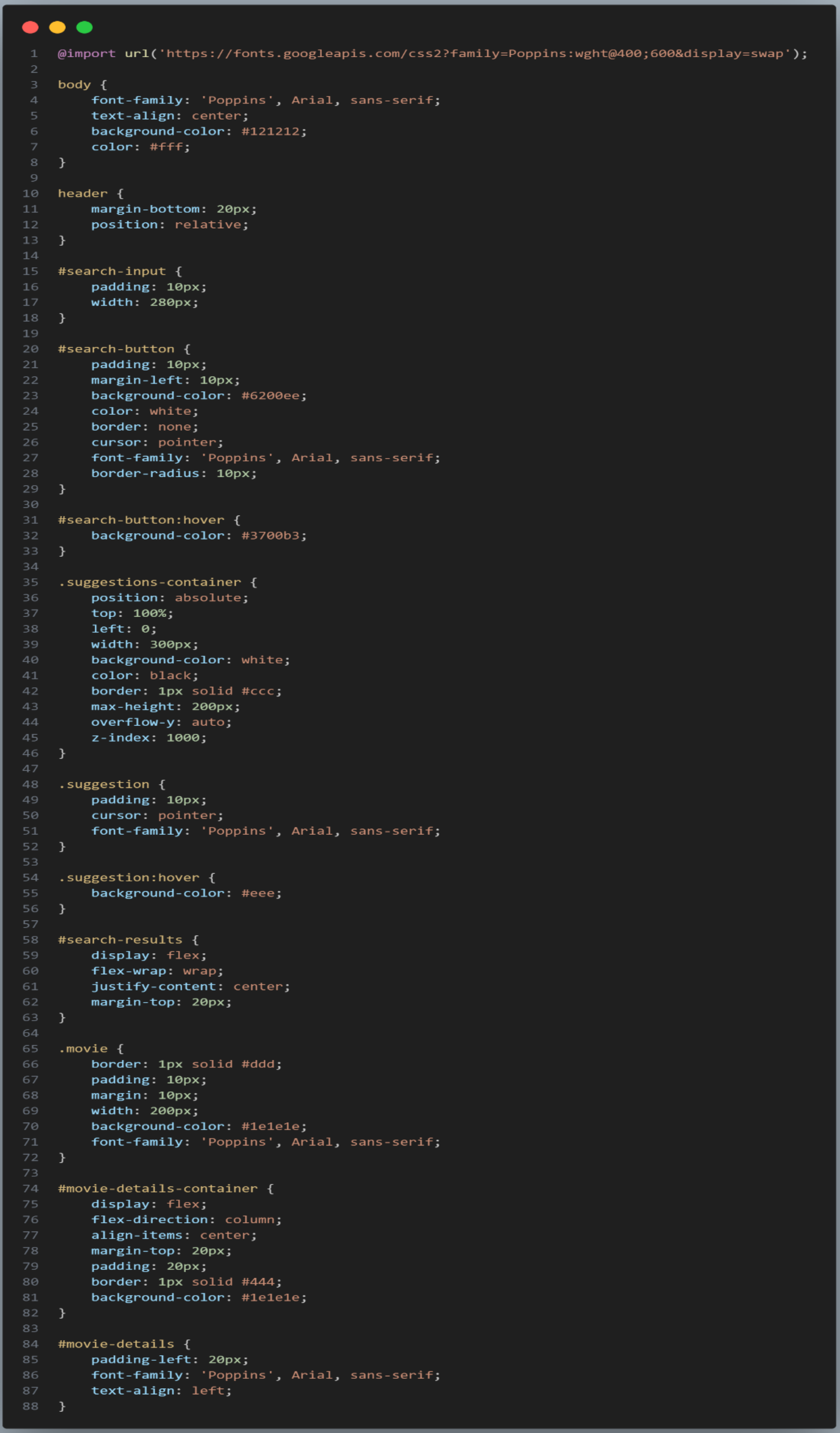


```javascript
import { setupWebSocket, sendSearchQuery } from './scripts/websocket.js';
import { fetchMovies, fetchMovieDetails, fetchMovieCast, fetchNewsArticles } from './scripts/api.js';

document.addEventListener('DOMContentLoaded', () => {
    const searchInput = document.getElementById('search-input');
    const searchButton = document.getElementById('search-button');
    const searchResults = document.getElementById('search-results');
    const movieDetailsContainer = document.getElementById('movie-details-container');
    const newsArticlesContainer = document.getElementById('news-articles');
    const loadingIndicator = document.getElementById('loading-indicator');
    const homeButton = document.getElementById('home-button');

    setupWebSocket();

    movieDetailsContainer.classList.add('hidden');
    newsArticlesContainer.classList.add('hidden');

    searchInput.addEventListener('input', () => {
        const query = searchInput.value;
        if (query) {
            sendSearchQuery(query);
        } else {
            document.getElementById('suggestions').innerHTML = '';
        }
    });

    searchButton.addEventListener('click', () => {
        const query = searchInput.value;
        if (query) {

            searchResults.innerHTML = '';
            movieDetailsContainer.innerHTML = '';
            newsArticlesContainer.innerHTML = '';
            document.getElementById('suggestions').innerHTML = '';

            movieDetailsContainer.classList.add('hidden');
            newsArticlesContainer.classList.add('hidden');

            loadingIndicator.style.display = 'block';

            fetchMovies(query).then(displayResults).finally(() => {
                loadingIndicator.style.display = 'none';
            });
        }
    });

    const displayResults = (movies) => {
        searchResults.innerHTML = '';
        if (movies.length === 0) {
            searchResults.innerHTML = '<p>No results found</p>';
            return;
        }
        movies.forEach(movie => {
            const movieElement = document.createElement('div');
            movieElement.className = 'movie';
            const posterPath = movie.poster_path ? `https://image.tmdb.org/t/p/w200${movie.poster_path}` : 'assets/images/placeholder.png';
            movieElement.innerHTML = `
                <img src="${posterPath}" alt="${movie.title}">
                <h3>${movie.title}</h3>
                <p>${movie.release_date}</p>
                <button class="details-button" onclick="showDetails(${movie.id}, '${movie.title}')">Details</button>
            `;
            searchResults.appendChild(movieElement);
        });
    };

    window.showDetails = async (movieId, title) => {

        loadingIndicator.style.display = 'block';

        const movieDetailsData = await fetchMovieDetails(movieId);
        const movieCast = await fetchMovieCast(movieId);
        const newsArticles = await fetchNewsArticles(title);


        loadingIndicator.style.display = 'none';

        if (movieDetailsData) {
            displayMovieDetails(movieDetailsData, movieCast, newsArticles);
            scrollToDetails();

            movieDetailsContainer.classList.remove('hidden');
            newsArticlesContainer.classList.remove('hidden');
        }
    };
```

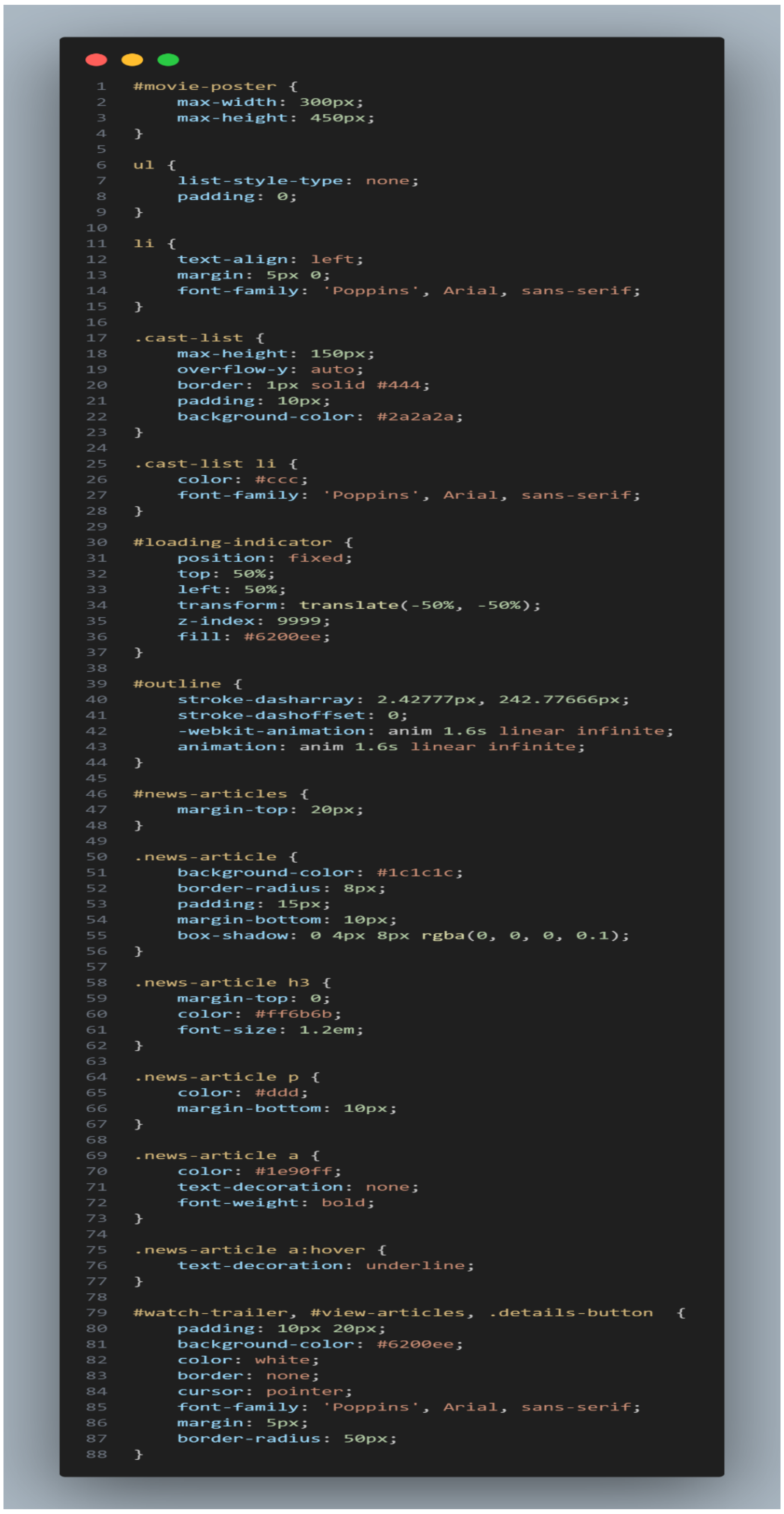


```javascript
const displayMovieDetails = (movie, cast, newsArticles) => {
        const castList = cast.map(actor => `<li>${actor.name} as ${actor.character}</li>`).join('');
        const newsList = newsArticles.length > 0 ? newsArticles.map(article => `
            <div class="news-article">
                <h3>${article.title}</h3>
                <p>${article.description}</p>
                <a href="${article.url}" target="_blank">Read more</a>
            </div>
        `).join('') : '<p>No articles found</p>';

        const posterPath = movie.poster_path ? `https://image.tmdb.org/t/p/w500${movie.poster_path}` : 'assets/images/placeholder.png';

        movieDetailsContainer.innerHTML = `
            <div>
                <img id="movie-poster" src="${posterPath}" alt="${movie.title}">
            </div>
            <div id="movie-details">
                <h2>${movie.title}</h2>
                <p id="movie-overview">${movie.overview}</p>
                <p><strong>Release Date:</strong> ${movie.release_date}</p>
                <p><strong>Rating:</strong> ${movie.vote_average}</p>
                <h3>Cast:</h3>
                <ul class="cast-list">${castList}</ul>
                <button id="watch-trailer" onclick="window.open('https://www.youtube.com/results?search_query=${movie.title}+trailer', '_blank')">Watch Trailer</button>
                <button id="view-articles">View Articles</button>
            </div>
        `;

        newsArticlesContainer.innerHTML = `
            <h2>Related News Articles</h2>
            ${newsList}
        `;


        document.getElementById('view-articles').addEventListener('click', (event) => {
            event.preventDefault();
            document.querySelector('#news-articles').scrollIntoView({ behavior: 'smooth' });
        });
    };

    const scrollToDetails = () => {
        movieDetailsContainer.scrollIntoView({ behavior: 'smooth' });
    };


    homeButton.addEventListener('click', () => {

        searchInput.value = '';

        searchResults.innerHTML = '';
        movieDetailsContainer.innerHTML = '';
        newsArticlesContainer.innerHTML = '';
        document.getElementById('suggestions').innerHTML = '';

        movieDetailsContainer.classList.add('hidden');
        newsArticlesContainer.classList.add('hidden');
    });
});
```

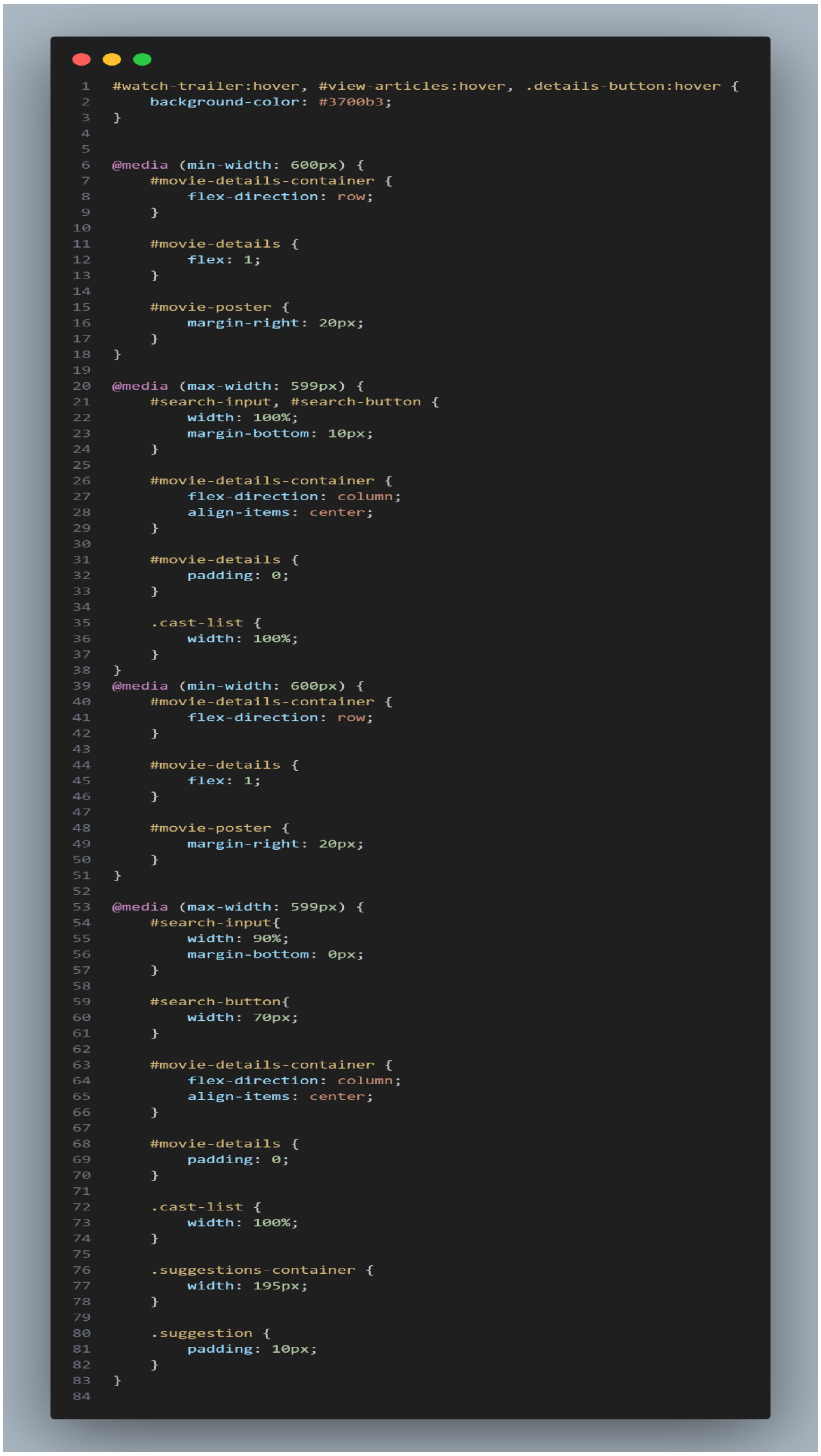
styles.css



```css
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap');

body {
    font-family: 'Poppins', Arial, sans-serif;
    text-align: center;
    background-color: #121212;
    color: #fff;
}

header {
    margin-bottom: 20px;
    position: relative;
}

#search-input {
    padding: 10px;
    width: 280px;
}

#search-button {
    padding: 10px;
    margin-left: 10px;
    background-color: #6200ee;
    color: white;
    border: none;
    cursor: pointer;
    font-family: 'Poppins', Arial, sans-serif;
    border-radius: 10px;
}

#search-button:hover {
    background-color: #3700b3;
}

.suggestions-container {
    position: absolute;
    top: 100%;
    left: 0;
    width: 300px;
    background-color: white;
    color: black;
    border: 1px solid #ccc;
    max-height: 200px;
    overflow-y: auto;
    z-index: 1000;
}

.suggestion {
    padding: 10px;
    cursor: pointer;
    font-family: 'Poppins', Arial, sans-serif;
}

.suggestion:hover {
    background-color: #eee;
}

#search-results {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    margin-top: 20px;
}

.movie {
    border: 1px solid #ddd;
    padding: 10px;
    margin: 10px;
    width: 200px;
    background-color: #1e1e1e;
    font-family: 'Poppins', Arial, sans-serif;
}

#movie-details-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 20px;
    padding: 20px;
    border: 1px solid #444;
    background-color: #1e1e1e;
}

#movie-details {
    padding-left: 20px;
    font-family: 'Poppins', Arial, sans-serif;
    text-align: left;
}
```

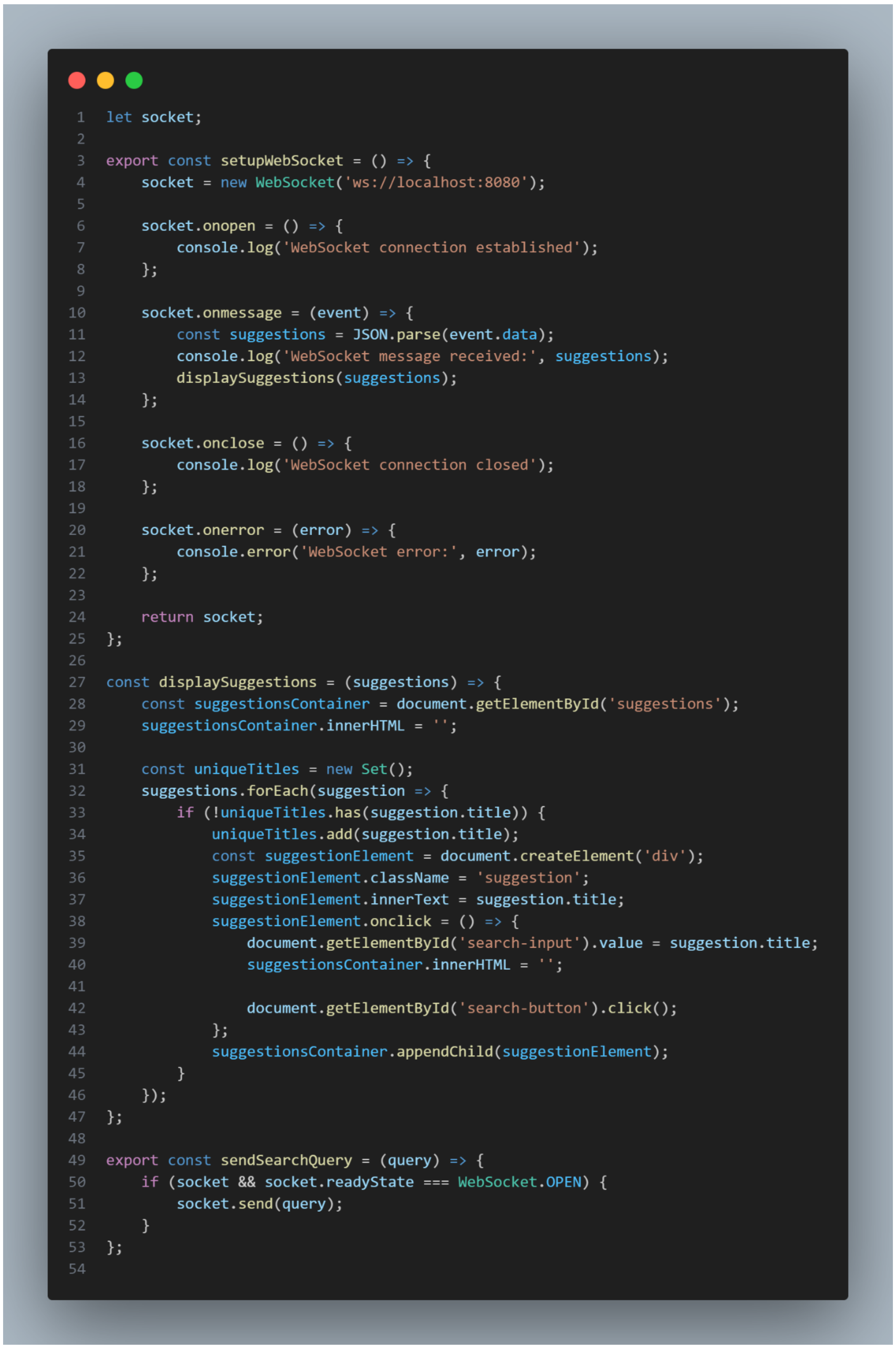


```css
#movie-poster {
    max-width: 300px;
    max-height: 450px;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    text-align: left;
    margin: 5px 0;
    font-family: 'Poppins', Arial, sans-serif;
}

.cast-list {
    max-height: 150px;
    overflow-y: auto;
    border: 1px solid #444;
    padding: 10px;
    background-color: #2a2a2a;
}

.cast-list li {
    color: #ccc;
    font-family: 'Poppins', Arial, sans-serif;
}

#loading-indicator {
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    z-index: 9999;
    fill: #6200ee;
}

#outline {
    stroke-dasharray: 2.42777px, 242.77666px;
    stroke-dashoffset: 0;
    -webkit-animation: anim 1.6s linear infinite;
    animation: anim 1.6s linear infinite;
}

#news-articles {
    margin-top: 20px;
}

.news-article {
    background-color: #1c1c1c;
    border-radius: 8px;
    padding: 15px;
    margin-bottom: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

.news-article h3 {
    margin-top: 0;
    color: #ff6b6b;
    font-size: 1.2em;
}

.news-article p {
    color: #ddd;
    margin-bottom: 10px;
}

.news-article a {
    color: #1e90ff;
    text-decoration: none;
    font-weight: bold;
}

.news-article a:hover {
    text-decoration: underline;
}

#watch-trailer, #view-articles, .details-button  {
    padding: 10px 20px;
    background-color: #6200ee;
    color: white;
    border: none;
    cursor: pointer;
    font-family: 'Poppins', Arial, sans-serif;
    margin: 5px;
    border-radius: 50px;
}
```



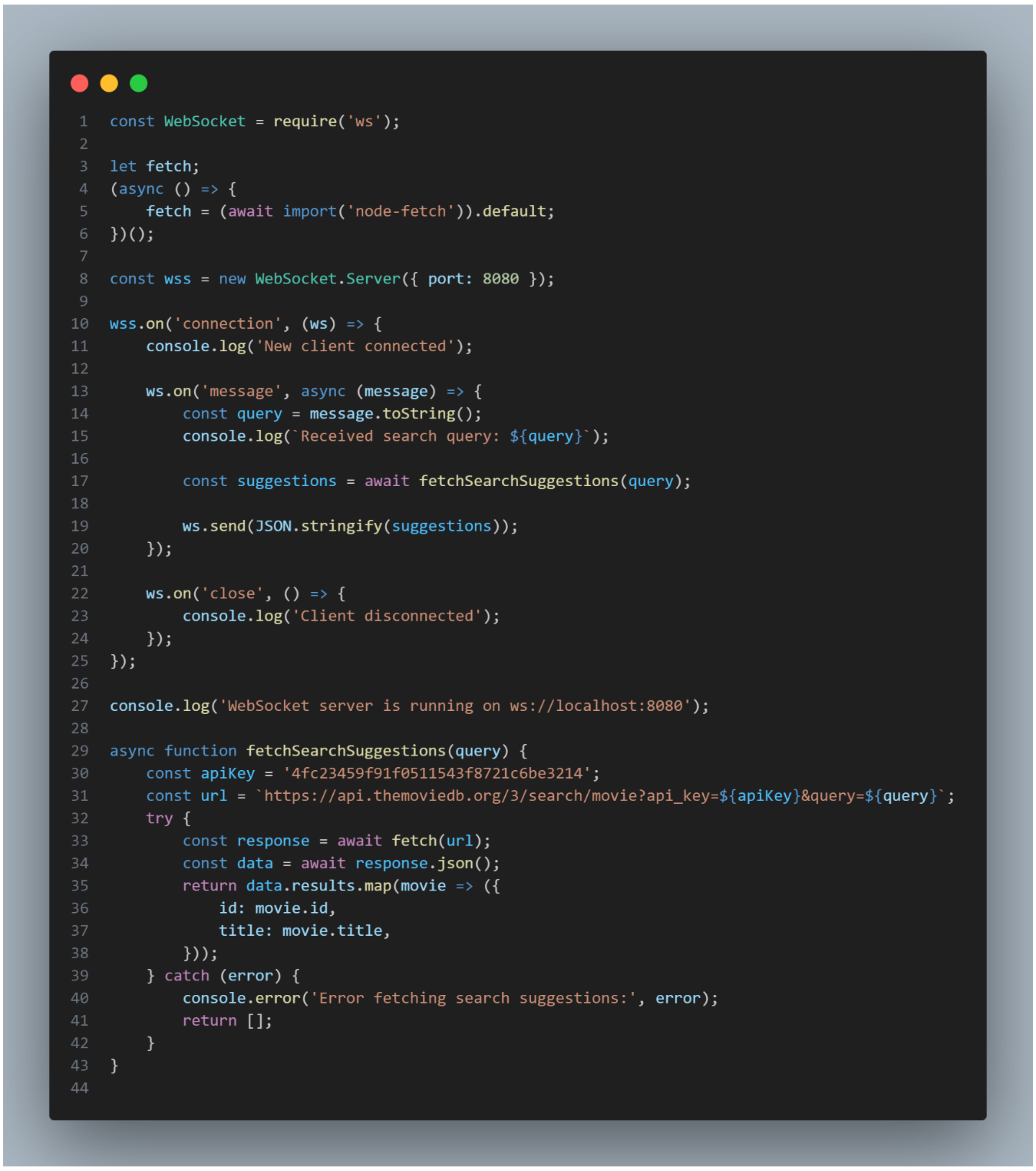
```css
#watch-trailer:hover, #view-articles:hover, .details-button:hover {
    background-color: #3700b3;
}


@media (min-width: 600px) {
    #movie-details-container {
        flex-direction: row;
    }

    #movie-details {
        flex: 1;
    }

    #movie-poster {
        margin-right: 20px;
    }
}

@media (max-width: 599px) {
    #search-input, #search-button {
        width: 100%;
        margin-bottom: 10px;
    }

    #movie-details-container {
        flex-direction: column;
        align-items: center;
    }

    #movie-details {
        padding: 0;
    }

    .cast-list {
        width: 100%;
    }
}
@media (min-width: 600px) {
    #movie-details-container {
        flex-direction: row;
    }

    #movie-details {
        flex: 1;
    }

    #movie-poster {
        margin-right: 20px;
    }
}

@media (max-width: 599px) {
    #search-input{
        width: 90%;
        margin-bottom: 0px;
    }

    #search-button{
        width: 70px;
    }

    #movie-details-container {
        flex-direction: column;
        align-items: center;
    }

    #movie-details {
        padding: 0;
    }

    .cast-list {
        width: 100%;
    }

    .suggestions-container {
        width: 195px;
    }

    .suggestion {
        padding: 10px;
    }
}

```

websocket.js



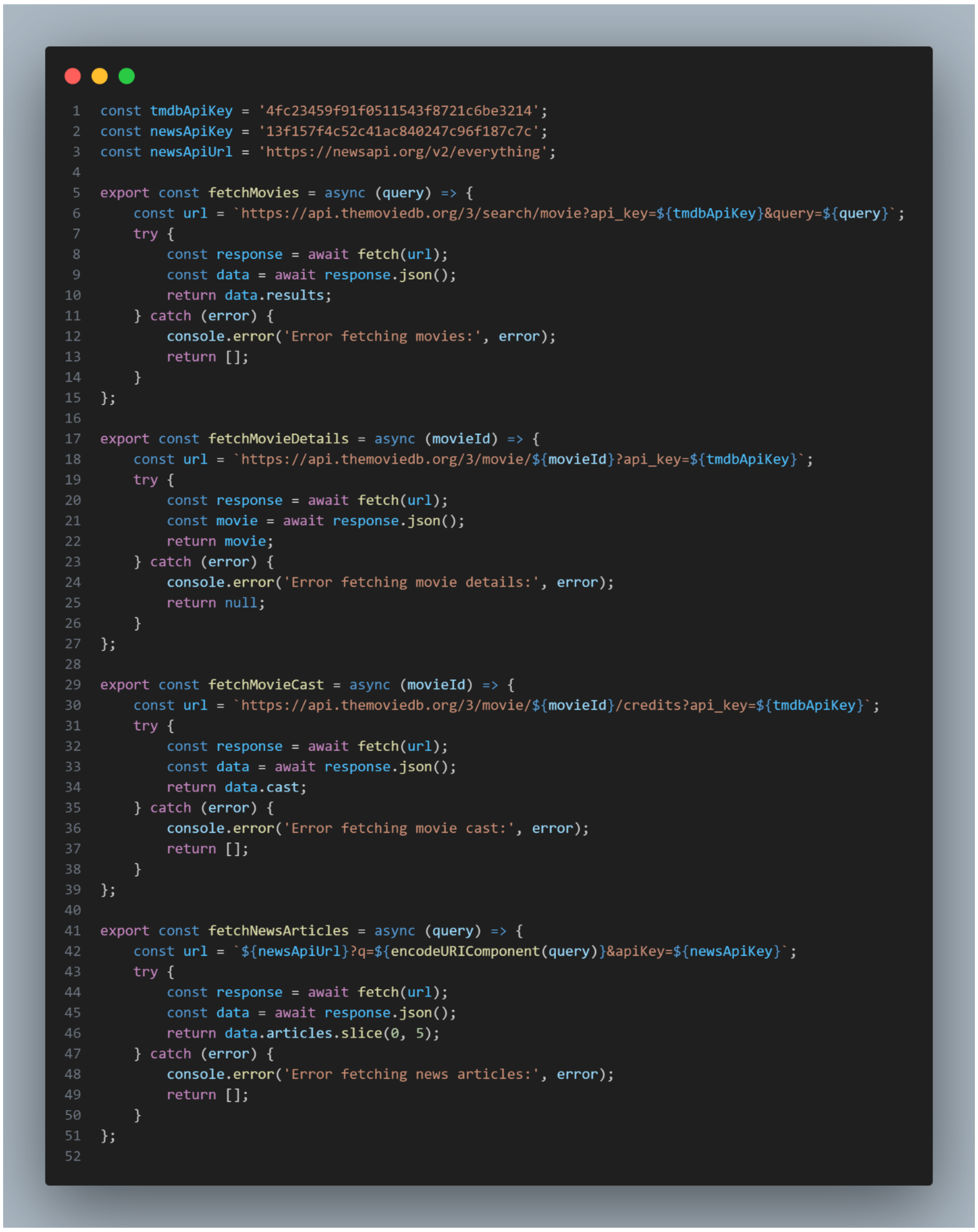
```javascript
let socket;

export const setupWebSocket = () => {
    socket = new WebSocket('ws://localhost:8080');

    socket.onopen = () => {
        console.log('WebSocket connection established');
    };

    socket.onmessage = (event) => {
        const suggestions = JSON.parse(event.data);
        console.log('WebSocket message received:', suggestions);
        displaySuggestions(suggestions);
    };

    socket.onclose = () => {
        console.log('WebSocket connection closed');
    };

    socket.onerror = (error) => {
        console.error('WebSocket error:', error);
    };

    return socket;
};

const displaySuggestions = (suggestions) => {
    const suggestionsContainer = document.getElementById('suggestions');
    suggestionsContainer.innerHTML = '';

    const uniqueTitles = new Set();
    suggestions.forEach(suggestion => {
        if (!uniqueTitles.has(suggestion.title)) {
            uniqueTitles.add(suggestion.title);
            const suggestionElement = document.createElement('div');
            suggestionElement.className = 'suggestion';
            suggestionElement.innerText = suggestion.title;
            suggestionElement.onclick = () => {
                document.getElementById('search-input').value = suggestion.title;
                suggestionsContainer.innerHTML = '';

                document.getElementById('search-button').click();
            };
            suggestionsContainer.appendChild(suggestionElement);
        }
    });
};

export const sendSearchQuery = (query) => {
    if (socket && socket.readyState === WebSocket.OPEN) {
        socket.send(query);
    }
};

```

server.js



```javascript
const WebSocket = require('ws');

let fetch;
(async () => {
    fetch = (await import('node-fetch')).default;
})();

const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', (ws) => {
    console.log('New client connected');

    ws.on('message', async (message) => {
        const query = message.toString();
        console.log(`Received search query: ${query}`);

        const suggestions = await fetchSearchSuggestions(query);

        ws.send(JSON.stringify(suggestions));
    });

    ws.on('close', () => {
        console.log('Client disconnected');
    });
});

console.log('WebSocket server is running on ws://localhost:8080');

async function fetchSearchSuggestions(query) {
    const apiKey = '4fc23459f91f0511543f8721c6be3214';
    const url = `https://api.themoviedb.org/3/search/movie?api_key=${apiKey}&query=${query}`;
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data.results.map(movie => ({
            id: movie.id,
            title: movie.title,
        }));
    } catch (error) {
        console.error('Error fetching search suggestions:', error);
        return [];
    }
}

```

api.js



```javascript
const tmdbApiKey = '4fc23459f91f0511543f8721c6be3214';
const newsApiKey = '13f157f4c52c41ac840247c96f187c7c';
const newsApiUrl = 'https://newsapi.org/v2/everything';

export const fetchMovies = async (query) => {
    const url = `https://api.themoviedb.org/3/search/movie?api_key=${tmdbApiKey}&query=${query}`;
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data.results;
    } catch (error) {
        console.error('Error fetching movies:', error);
        return [];
    }
};

export const fetchMovieDetails = async (movieId) => {
    const url = `https://api.themoviedb.org/3/movie/${movieId}?api_key=${tmdbApiKey}`;
    try {
        const response = await fetch(url);
        const movie = await response.json();
        return movie;
    } catch (error) {
        console.error('Error fetching movie details:', error);
        return null;
    }
};

export const fetchMovieCast = async (movieId) => {
    const url = `https://api.themoviedb.org/3/movie/${movieId}/credits?api_key=${tmdbApiKey}`;
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data.cast;
    } catch (error) {
        console.error('Error fetching movie cast:', error);
        return [];
    }
};

export const fetchNewsArticles = async (query) => {
    const url = `${newsApiUrl}?q=${encodeURIComponent(query)}&apiKey=${newsApiKey}`;
    try {
        const response = await fetch(url);
        const data = await response.json();
        return data.articles.slice(0, 5);
    } catch (error) {
        console.error('Error fetching news articles:', error);
        return [];
    }
};

```