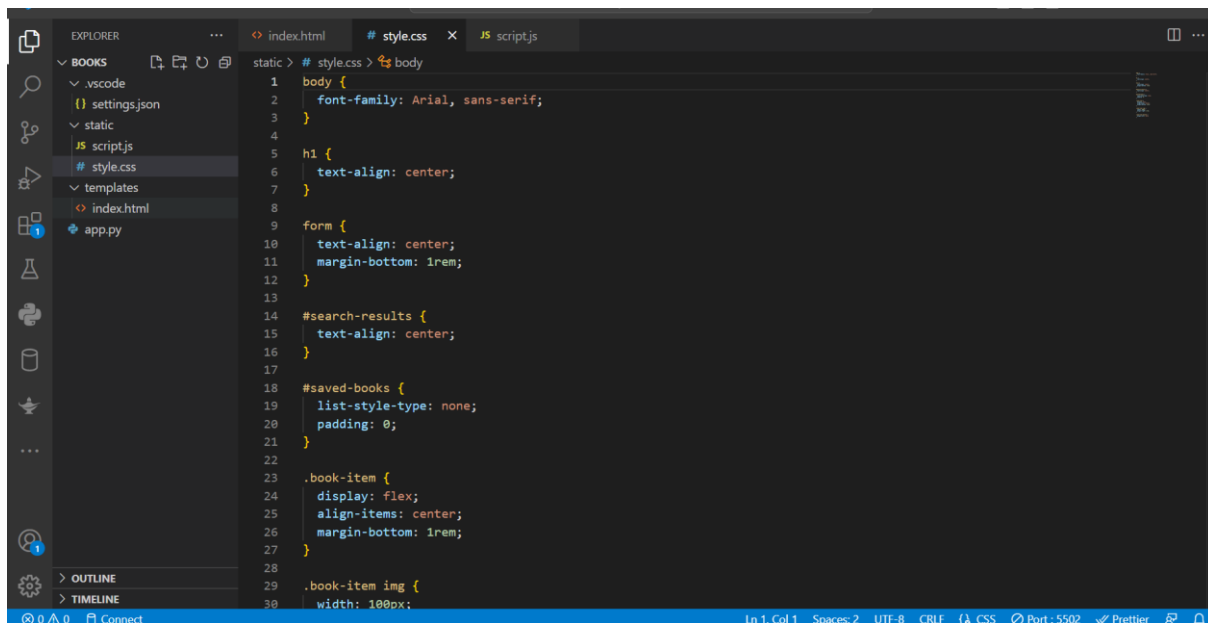


The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like .vscode, static, and templates. The static folder is expanded, showing files like script.js, style.css, index.html, and app.py. The script.js file is selected and its content is displayed in the main editor area. The code is a JavaScript script that handles a search function. It uses document.getElementById to get the search button and input, and fetch to send a request to a search API. It then processes the response and updates the search results div with HTML for each book found. The status bar at the bottom shows 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'CRLF', 'JavaScript', 'Port: 5502', 'Prettier', and 'D'.

```
static > JS script.js
1 document.getElementById('search-button').addEventListener('click', async () => {
2   const query = document.getElementById('search-input').value;
3   if (!query) {
4     return;
5   }
6
7   const response = Loading... ch('/api/search?q=${query}');
8   const data = await response.json();
9
10  const searchResultsDiv = document.getElementById('search-results');
11  searchResultsDiv.innerHTML = '';
12
13  if (data.error) {
14    searchResultsDiv.innerHTML = `<p>${data.error}</p>`;
15  } else {
16    data.items.forEach(book => {
17      const bookDiv = document.createElement('div');
18      bookDiv.className = 'book';
19      bookDiv.innerHTML = `
20        
21        <h3>${book.title}</h3>
22        <p>Author(s): ${book.authors.join(', ')}</p>
23        <p>Description: ${book.description}</p>
24        <button class="add-to-library" data-book-id="${book.id}">Add to Library</button>
25      `;
26      searchResultsDiv.appendChild(bookDiv);
27    });
28
29    // Add event listener to "Add to Library" buttons
30    const addToLibraryButtons = document.getElementsByClassName('add-to-library');
```

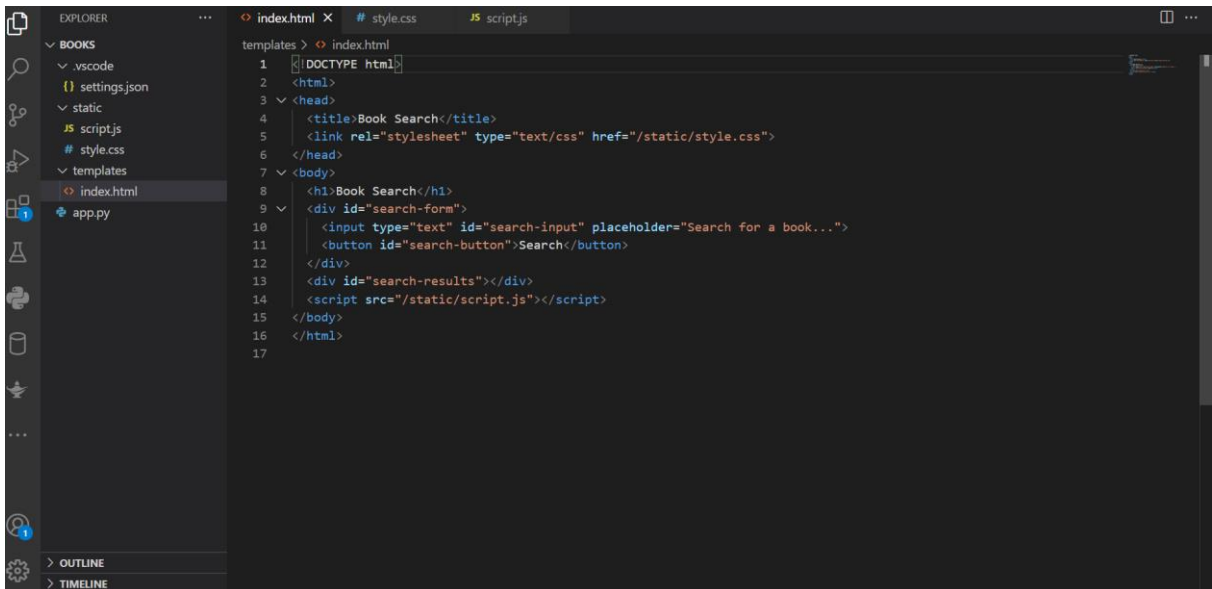
Figure1.1: Shows the JavaScript code



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like .vscode, static, and templates. The static folder is expanded, showing files like script.js, style.css, index.html, and app.py. The style.css file is selected and its content is displayed in the main editor area. The code is a CSS file that defines styles for the search results. It sets the font-family to Arial, sans-serif, and the text-align to center for the h1, form, and search-results. It also defines a list-style-type of none and padding of 0 for the #saved-books, and a display of flex, align-items of center, and margin-bottom of 1rem for the .book-item. The status bar at the bottom shows 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'CRLF', 'CSS', 'Port: 5502', 'Prettier', and 'D'.

```
static > # style.css > body
1 body {
2   font-family: Arial, sans-serif;
3 }
4
5 h1 {
6   text-align: center;
7 }
8
9 form {
10  text-align: center;
11  margin-bottom: 1rem;
12 }
13
14 #search-results {
15   text-align: center;
16 }
17
18 #saved-books {
19   list-style-type: none;
20   padding: 0;
21 }
22
23 .book-item {
24   display: flex;
25   align-items: center;
26   margin-bottom: 1rem;
27 }
28
29 .book-item img {
30   width: 100px;
```

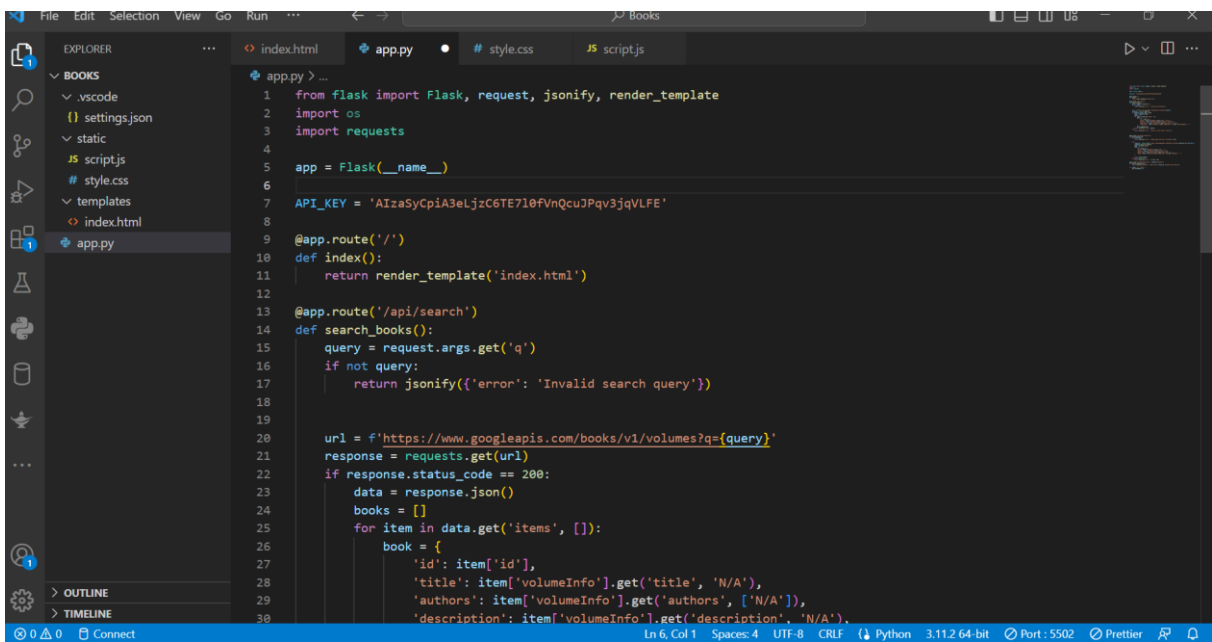
Figure1.2:Shows the Css code



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders 'BOOKS', '.vscode', and 'static'. Inside 'BOOKS', there are files 'script.js', 'style.css', and 'templates'. The 'templates' folder is expanded, showing 'index.html' and 'app.py'. The main editor area displays the content of 'index.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Book Search</title>
5   <link rel="stylesheet" type="text/css" href="/static/style.css">
6 </head>
7 <body>
8   <h1>Book Search</h1>
9   <div id="search-form">
10     <input type="text" id="search-input" placeholder="Search for a book...">
11     <button id="search-button">Search</button>
12   </div>
13   <div id="search-results"></div>
14   <script src="/static/script.js"></script>
15 </body>
16 </html>
17
```

Figure1.3shows html code



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders 'BOOKS', '.vscode', and 'static'. Inside 'BOOKS', there are files 'script.js', 'style.css', and 'templates'. The 'templates' folder is expanded, showing 'index.html' and 'app.py'. The main editor area displays the content of 'app.py'. The code is as follows:

```
1 from flask import Flask, request, jsonify, render_template
2 import os
3 import requests
4
5 app = Flask(__name__)
6
7 API_KEY = 'AIzaSyCpiA3elJzC6TE7l0fVnQcu3Pqv3jqVLFE'
8
9 @app.route('/')
10 def index():
11     return render_template('index.html')
12
13 @app.route('/api/search')
14 def search_books():
15     query = request.args.get('q')
16     if not query:
17         return jsonify({'error': 'Invalid search query'})
18
19     url = f'https://www.googleapis.com/books/v1/volumes?q={query}'
20     response = requests.get(url)
21     if response.status_code == 200:
22         data = response.json()
23         books = []
24         for item in data.get('items', []):
25             book = {
26                 'id': item['id'],
27                 'title': item['volumeInfo'].get('title', 'N/A'),
28                 'authors': item['volumeInfo'].get('authors', ['N/A']),
29                 'description': item['volumeInfo'].get('description', 'N/A')
30             }
```

figure1.4: Shows the python

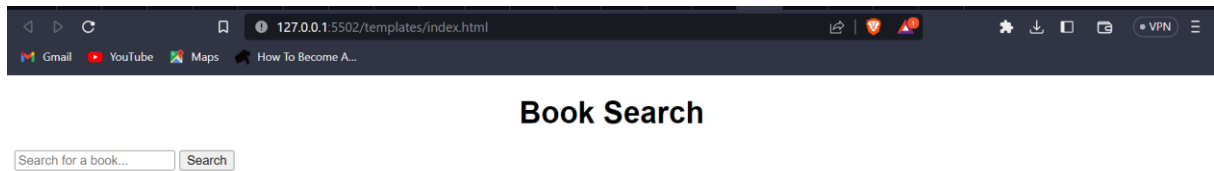


Figure1.5: Shows the output of the code