

Flask Project

Intelligenza artificiale e tecnologie web
2024

Welcome to the Homepage

This is the main page of the website.

[About](#) | [Contact](#)

About Us

This is the about page.

[Home](#) | [Contact](#)

Contact Us

You can contact us at:

Email: example@email.com

[Home](#) | [About](#)

```
M+ README.md  app.py  about.html  contact.html
1  from flask import Flask, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return render_template('index.html')
8
9  @app.route('/about')
10 def about():
11     return render_template('about.html')
12
13 @app.route('/contact')
14 def contact():
15     return render_template('contact.html')
16
17 if __name__ == '__main__':
18     app.run(debug=True)
```

Welcome to the Homepage

This is the main page of the website.

[About](#) | [Contact](#)

About Us

This is the about page.

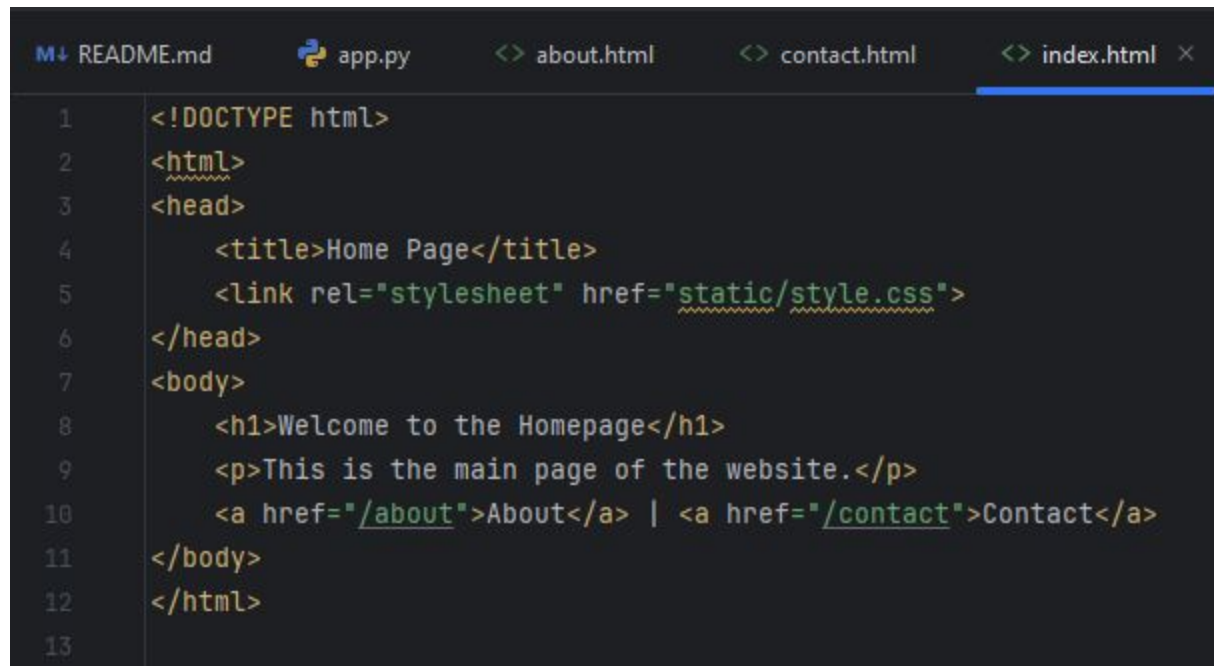
[Home](#) | [Contact](#)

Contact Us

You can contact us at:

Email: example@email.com

[Home](#) | [About](#)

A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'README.md', 'app.py', 'about.html', and 'index.html'. The 'index.html' tab is selected and highlighted with a blue underline. The code in the editor is HTML for a homepage. It starts with a DOCTYPE declaration, followed by the html, head, and body tags. The head section contains a title 'Home Page' and a link to a stylesheet 'static/style.css'. The body section contains a welcome message, a paragraph about the website, and two links: 'About' and 'Contact'. The code is color-coded: HTML tags are in blue, text is in light green, and attribute values are in yellow. Line numbers 1 through 13 are visible on the left side of the editor.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Home Page</title>
5     <link rel="stylesheet" href="static/style.css">
6 </head>
7 <body>
8     <h1>Welcome to the Homepage</h1>
9     <p>This is the main page of the website.</p>
10    <a href="/about">About</a> | <a href="/contact">Contact</a>
11 </body>
12 </html>
13
```

Welcome to the Homepage

This is the main page of the website.

[About](#) | [Contact](#)

About Us

This is the about page.

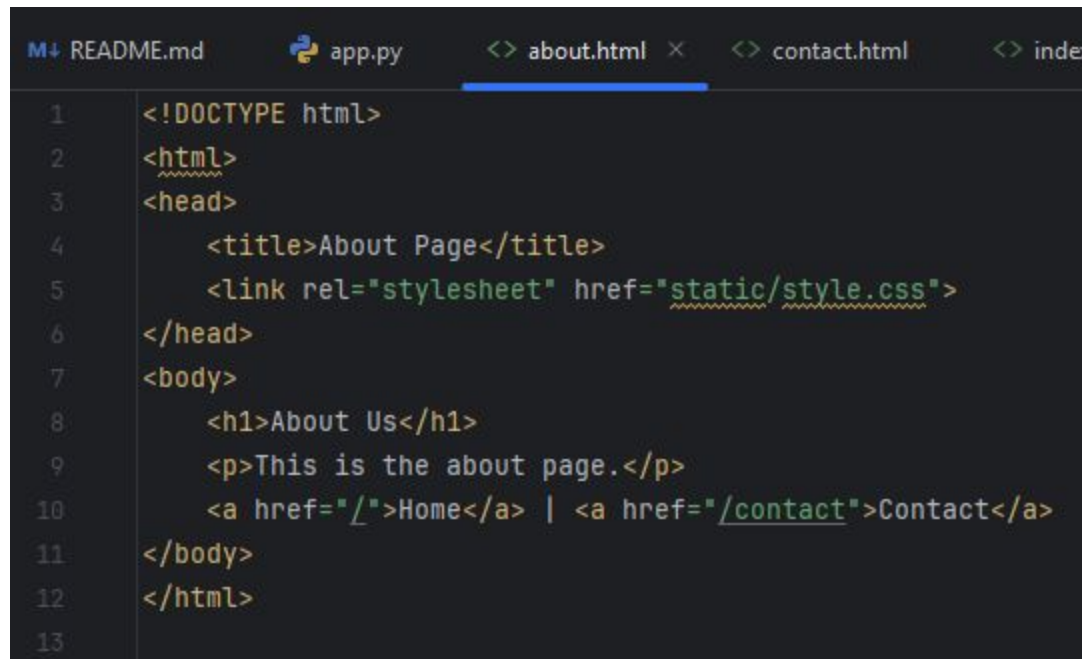
[Home](#) | [Contact](#)

Contact Us

You can contact us at:

Email: example@email.com

[Home](#) | [About](#)



```
M+ README.md  app.py  <> about.html x  <> contact.html  <> index.html

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>About Page</title>
5      <link rel="stylesheet" href="static/style.css">
6  </head>
7  <body>
8      <h1>About Us</h1>
9      <p>This is the about page.</p>
10     <a href="/">Home</a> | <a href="/contact">Contact</a>
11 </body>
12 </html>
13
```

Welcome to the Homepage

This is the main page of the website.

[About](#) | [Contact](#)

About Us

This is the about page.

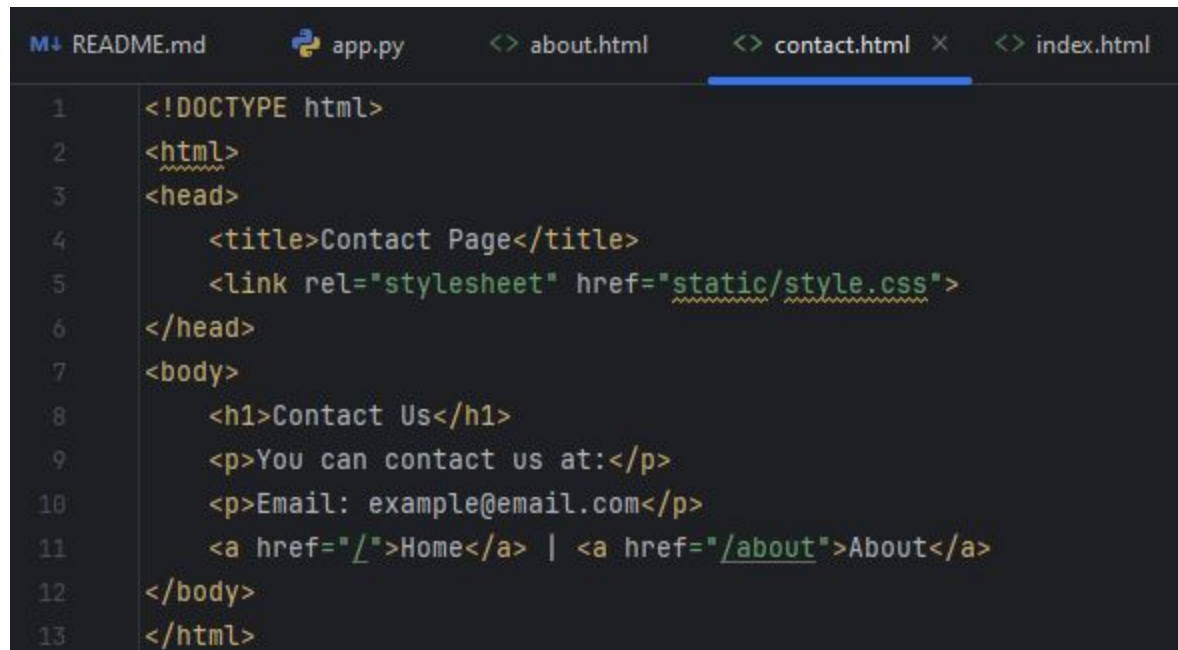
[Home](#) | [Contact](#)

Contact Us

You can contact us at:

Email: example@email.com

[Home](#) | [About](#)

A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'README.md', 'app.py', 'about.html', 'contact.html' (which is the active tab), and 'index.html'. The code in the active tab is HTML for a contact page. It includes a DOCTYPE declaration, an HTML tag, a head section with a title 'Contact Page' and a link to 'static/style.css', and a body section with an h1 'Contact Us', a paragraph 'You can contact us at:', another paragraph 'Email: example@email.com', and a navigation bar with links to '/' (Home) and '/about' (About).

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Contact Page</title>
5     <link rel="stylesheet" href="static/style.css">
6 </head>
7 <body>
8     <h1>Contact Us</h1>
9     <p>You can contact us at:</p>
10    <p>Email: example@email.com</p>
11    <a href="/">Home</a> | <a href="/about">About</a>
12 </body>
13 </html>
```

Welcome to the Homepage

This is the main page of the website.

[About](#) | [Contact](#)

About Us

This is the about page.

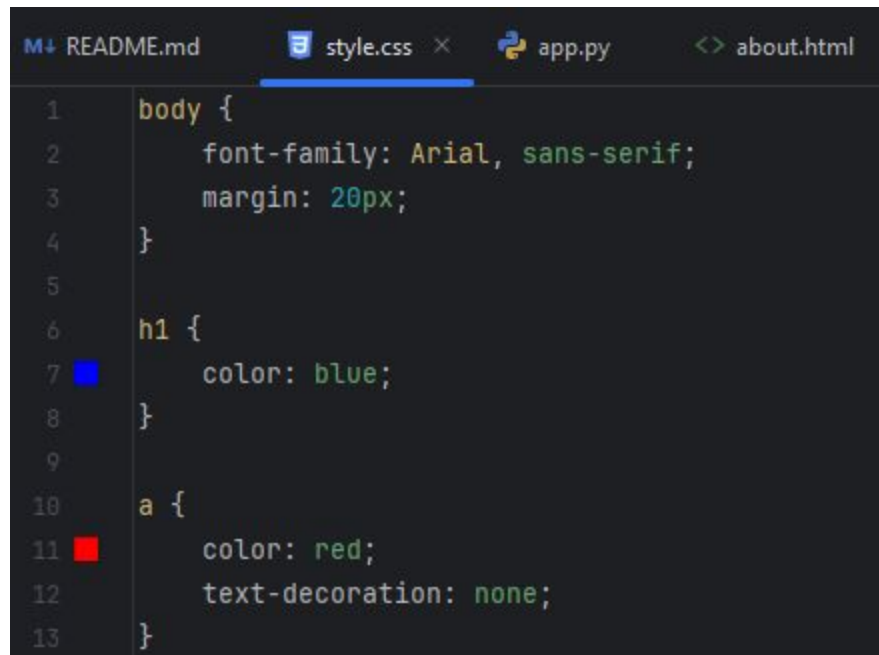
[Home](#) | [Contact](#)

Contact Us

You can contact us at:

Email: example@email.com

[Home](#) | [About](#)


A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'README.md' (with a blue 'M' icon), 'style.css' (with a blue icon and a close button), 'app.py' (with a Python logo icon), and 'about.html' (with a green '<>' icon). The 'style.css' tab is active and highlighted with a blue underline. The code in the editor is CSS, with line numbers 1 through 13 on the left. The code defines styles for the body, h1, and a link (a).

```
1 body {  
2     font-family: Arial, sans-serif;  
3     margin: 20px;  
4 }  
5  
6 h1 {  
7     color: blue;  
8 }  
9  
10 a {  
11     color: red;  
12     text-decoration: none;  
13 }
```


Richieste

Creare una nuova pagina con la presentazione del team


Meet Our Team



John Doe
Founder and CEO
john@company.com



Jane Smith
COO
jane@company.com



Bob Johnson
Lead Engineer
bob@company.com

Home | About | Contact

Richieste

Creare una nuova pagina con la presentazione del team

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Our Team</title>
5   <link rel="stylesheet" href="static/style.css">
6 </head>
7 <body>
8 <h1>Meet Our Team</h1>
9 <div class="team-member">
10   
11   <h2>John Doe</h2>
12   <p>Founder and CEO</p>
13   <p>john@company.com</p>
14 </div>
15
16 <div class="team-member">
17   
18   <h2>Jane Smith</h2>
19   <p>COO</p>
20   <p>jane@company.com</p>
21 </div>
22 |
23 <div class="team-member">
24   
25   <h2>Bob Johnson</h2>
26   <p>Lead Engineer</p>
27   <p>bob@company.com</p>
28 </div>
29
30 <a href="/">Home</a> | <a href="/about">About</a> | <a href="/contact">Contact</a>
31 </body>
32 </html>
```


Richieste

Creare una nuova pagina con la presentazione del team

```
14
15  ✓ .team-member img {
16      width: 200px; /* Set fixed width */
17      height: 200px; /* Set fixed height */
18      object-fit: cover; /* This ensures the image covers the area without distortion */
19      border-radius: 50%; /* This makes the images circular */
20  }
21
22  ✓ .team-member {
23      margin: 20px 0;
24      text-align: center;
25  }
```

Caricare i membri del team dal server

```
</team>

@app.route('/team')
def team():
    team_members = [
        {
            'name': 'John Doe',
            'position': 'Founder and CEO',
            'email': 'john@company.com',
            'image': 'john-doe.jpg'
        },
        {
            'name': 'Jane Smith',
            'position': 'COO',
            'email': 'jane@company.com',
            'image': 'jane-smith.jpg'
        },
        {
            'name': 'Bob Johnson',
            'position': 'Lead Engineer',
            'email': 'bob@company.com',
            'image': 'bob-johnson.jpeg'
        }
    ]

    return render_template(template_name_or_list='team.html', team_members=team_members)
```

Caricare i membri del team dal server

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Our Team</title>
5      <link rel="stylesheet" href="static/style.css">
6  </head>
7  <body>
8      <h1>Meet Our Team</h1>
9
10     {% for member in team_members %}
11     <div class="team-member">
12         
13         <h2>{{ member.name }}</h2>
14         <p>{{ member.position }}</p>
15         <p>{{ member.email }}</p>
16     </div>
17     {% endfor %}
18
19     <a href="/">Home</a> | <a href="/about">About</a> | <a href="/contact">Contact</a>
20 </body>
21 </html>
```

Caricare i membri del team dal server - da file



The image shows a code editor interface with a dark theme. At the top, there is a tab bar with several files: README.md, style.css, app.py, about.html, contact.html, team.html, and team.csv (which is currently selected). The main editing area displays the contents of team.csv, which is a CSV file with four lines of data. The first line is a header row with columns: name, position, email, and image. The subsequent three lines contain data for John Doe, Jane Smith, and Bob Johnson. The text is color-coded: 'name' is orange, 'position' is red, 'email' is purple, and 'image' is green. Line numbers 1 through 4 are visible on the left side of the editor.

```
1 name,position,email,image
2 John Doe,Founder and CEO,john@company.com,john-doe.jpg
3 Jane Smith,COO,jane@company.com,jane-smith.jpg
4 Bob Johnson,Lead Engineer,bob@company.com,bob-johnson.jpg
```

Caricare i membri del team dal server - da file

```
1  from flask import Flask, render_template
2  import csv
3  import os
```

```
23  def load_team_data(): 1 usage
24      team_members = []
25      csv_path = os.path.join(os.path.dirname(__file__), 'static/team.csv')
26
27      try:
28          with open(csv_path, mode='r', encoding='utf-8') as csv_file:
29              csv_reader = csv.DictReader(csv_file)
30              for row in csv_reader:
31                  team_members.append(row)
32          return team_members
33      except FileNotFoundError:
34          print(f"Warning: {csv_path} not found")
35          return []
36      except Exception as e:
37          print(f"Error loading team data: {e}")
38          return []
39
```


Caricare i membri del team dal server - da file

```
37  @app.route('/team')
38  def team():
39      team_members = load_team_data()
40      return render_template(template_name_or_list: 'team.html', team_members=team_members)
```

Utilizzo dei template

```
M+ README.md style.css app.py <> base.html x <> about.html <> contact.html <> team.html team.csv
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>{% block title %}{% endblock %}</title>
5     <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
6 </head>
7 <body>
8     <nav class="navbar">
9         <div class="nav-links">
10             <a href="/" {% if request.path == '/' %}class="active"{% endif %}>Home</a>
11             <a href="/about" {% if request.path == '/about' %}class="active"{% endif %}>About</a>
12             <a href="/team" {% if request.path == '/team' %}class="active"{% endif %}>Team</a>
13             <a href="/contact" {% if request.path == '/contact' %}class="active"{% endif %}>Contact</a>
14         </div>
15     </nav>
16
17     <div class="content">
18         {% block content %}
19         {% endblock %}
20     </div>
21
22     <footer>
23         <p>&copy; 2024 Your Company. All rights reserved.</p>
24     </footer>
25 </body>
26 </html>
27
```

Utilizzo dei template

```
1 {% extends "base.html" %}
2
3 {% block title %}Home{% endblock %}
4
5 {% block content %}
6     <h1>Welcome to the Homepage</h1>
7      <p>This is the main page of the website.</p>
8 {% endblock %}
```


Creare un proprio stile in *style.css*

```
M+ README.md style.css x app.py <> base.html
1 body {
2   font-family: Arial, sans-serif;
3   margin: 0;
4   padding: 0;
5   min-height: 100vh;
6   display: flex;
7   flex-direction: column;
8 }
9
10 .navbar {
11   background-color: #333;
12   padding: 1rem;
13   margin-bottom: 2rem;
14 }
15
16 .nav-links {
17   text-align: center;
18 }
19
20 .nav-links a {
21   color: white;
22   text-decoration: none;
23   padding: 0.5rem 1rem;
24   margin: 0 0.5rem;
25 }
```


```
M+ README.md style.css x app.py <> base.html
26
27 .nav-links a:hover {
28   background-color: #555;
29   border-radius: 4px;
30 }
31
32 .nav-links a.active {
33   background-color: #555;
34   border-radius: 4px;
35 }
36
37 .content {
38   flex: 1;
39   width: 80%;
40   margin: 0 auto;
41   padding: 0 1rem;
42 }
43
44 h1 {
45   color: #333;
46   text-align: center;
47   margin-bottom: 2rem;
48 }
```

```
50 .team-member img {
51   width: 200px;
52   height: 200px;
53   object-fit: cover;
54   border-radius: 50%;
55 }
56
57 .team-member {
58   margin: 20px 0;
59   text-align: center;
60 }
61
62 footer {
63   background-color: #333;
64   color: white;
65   text-align: center;
66   padding: 1rem;
67   margin-top: 2rem;
68 }
```


Creare un proprio stile in *style.css*

[Home](#)[About](#)[Team](#)[Contact](#)


Meet Our Team



John Doe
Founder and CEO
john@company.com



Jane Smith
COO
jane@company.com

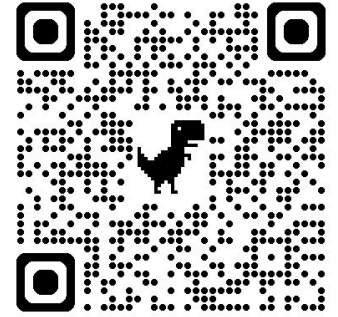


Bob Johnson
Lead Engineer
bob@company.com

© 2024 Your Company. All rights reserved.

API & Form

CSV Libri



```
M+ README.md style.css app.py <> add_book.html <> books.html books.csv x
1 Title,Author,Genre,Height,Publisher
2 Fundamentals of Wavelets,"Goswami, Jaideva",signal_processing,228,Wiley
3 Data Smart,"Foreman, John",data_science,235,Wiley
4 God Created the Integers,"Hawking, Stephen",mathematics,197,Penguin
5 Superfreakonomics,"Dubner, Stephen",economics,179,HarperCollins
6 Orientalism,"Said, Edward",history,197,Penguin
7 "Nature of Statistical Learning Theory, The","Vapnik, Vladimir",data_science,230,Springer
8 Integration of the Indian States,"Menon, V P",history,217,Orient Blackswan
9 "Drunkard's Walk, The","Mlodinow, Leonard",science,197,Penguin
10 Image Processing & Mathematical Morphology,"Shih, Frank",signal_processing,241,CRC
11 How to Think Like Sherlock Holmes,"Konnikova, Maria",psychology,240,Penguin
12 Data Scientists at Work,Sebastian Gutierrez,data_science,230,Apress
```

Importazione

```
def load_books_data(): 4 usages
    books = []
    csv_path = os.path.join(os.path.dirname(__file__), 'static/books.csv')

    try:
        with open(csv_path, mode='r', encoding='utf-8') as csv_file:
            csv_reader = csv.DictReader(csv_file)
            for row in csv_reader:
                books.append(row)
        return books
    except FileNotFoundError:
        print(f"Warning: {csv_path} not found")
        return []
    except Exception as e:
        print(f"Error loading books data: {e}")
        return []
```

Route e visualizzazione HTML

```
@app.route('/books')
def books():
    books = load_books_data()

    return render_template(template_name_or_list: 'books.html', books=books)
```

```
<table class="table table-striped">
  <thead>
    <tr>
      <th>Titolo</th>
      <th>Autore</th>
      <th>Genere</th>
      <th>Altezza</th>
      <th>Editore</th>
      <th>Azioni</th>
    </tr>
  </thead>
  <tbody>
    {% for book in books %}
      <tr>
        <td>{{ book.Title }}</td>
        <td>{{ book.Author }}</td>
        <td>{{ book.Genre }}</td>
        <td>{{ book.Height }}</td>
        <td>{{ book.Publisher }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

API GET

```
from flask import jsonify

@api/books
@app.route(rule: '/api/books', methods=['GET'])
def api_books():
    books = load_books_data()
    return jsonify(books)
```

```
# add api to return a book by Title
@api/books/{title}
@app.route(rule: '/api/books/<title>', methods=['GET'])
def api_book(title):
    books = load_books_data()
    for book in books:
        if book['Title'] == title:
            return jsonify(book)
    return jsonify({'error': 'Book not found'}), 404
```

← → ↺ 127.0.0.1:5000/api/books



» | Tutti i preferiti

Formatta il codice

```
[{"Author": "Goswami, Jaideva", "Genre": "signal_processing", "Height": "228", "Publisher": "Wiley", "Title": "Fundamentals of Wavelets"}, {"Author": "Foreman, John", "Genre": "data_science", "Height": "235", "Publisher": "Wiley", "Title": "Data Smart"}, {"Author": "Hawking, Stephen", "Genre": "mathematics", "Height": "197", "Publisher": "Penguin", "Title": "God Created the Integers"}, {"Author": "Dubner, Stephen", "Genre": "economics", "Height": "179", "Publisher": "HarperCollins", "Title": "Superfreakonomics"}, {"Author": "Said, Edward", "Genre": "history", "Height": "197", "Publisher": "Penguin", "Title": "Orientalism"}, {"Author": "Vapnik, Vladimir", "Genre": "data_science", "Height": "230", "Publisher": "Springer", "Title": "Nature of Statistical Learning Theory, The"}, {"Author": "Menon, V P", "Genre": "history", "Height": "217", "Publisher": "Orient Blackswan", "Title": "Integration of the Indian States"}, {"Author": "Mlodinow, Leonard", "Genre": "science", "Height": "197", "Publisher": "Penguin", "Title": "Drunkard's Walk, The"}, {"Author": "Shih, Frank", "Genre": "signal_processing", "Height": "241", "Publisher": "CRC", "Title": "Image Processing & Mathematical Morphology"}, {"Author": "Konnikova, Maria", "Genre": "psychology", "Height": "240", "Publisher": "Penguin", "Title": "How to Think Like Sherlock Holmes"}, {"Author": "Sebastian Gutierrez", "Genre": "data_science", "Height": "230", "Publisher": "Apress", "Title": "Data Scientists at Work"}, {"Author": "cc", "Genre": "", "Height": "", "Publisher": "", "Title": "aa"}]
```


Form

Aggiungi un libro

Title:

Author:

Titolo	Autore	Genere	Altezza	Editore	Azioni
Fundamentals of Wavelets	Goswami, Jaideva	signal_processing	228	Wiley	
Data Smart	Foreman, John	data_science	235	Wiley	
God Created the Integers	Hawking, Stephen	mathematics	197	Penguin	
Superfreakonomics	Dubner, Stephen	economics	179	HarperCollins	
Orientalism	Said, Edward	history	197	Penguin	
Nature of Statistical Learning Theory, The	Vapnik, Vladimir	data_science	230	Springer	
Integration of the Indian States	Menon, V P	history	217	Orient Blackswan	
Drunkard's Walk, The	Mlodinow, Leonard	science	197	Penguin	
Image Processing & Mathematical Morphology	Shih, Frank	signal_processing	241	CRC	
How to Think Like Sherlock Holmes	Konnikova, Maria	psychology	240	Penguin	
Data Scientists at Work	Sebastian Gutierrez	data_science	230	Apress	

Form con API POST

```
from flask import request

# /api/books
@app.route(rule='/api/books', methods=['POST'])
def add_book_api():
    try:
        new_book = request.get_json() # Get data from the request
        books = load_books_data() # Load existing books
        print(new_book)
        # Validate incoming data
        if not all(key in new_book for key in ('Title', 'Author')):
            return jsonify({'error': 'Missing required fields'}), 400

        books.append(new_book)
        print(books)
        # Append to CSV file
        csv_path = os.path.join(os.path.dirname(__file__), 'static/books.csv')
        with open(csv_path, mode='a', encoding='utf-8', newline='') as csv_file:
            writer = csv.DictWriter(csv_file, fieldnames=['Title', 'Author', 'Genre', 'Height', 'Publisher'])
            writer.writerow(new_book) # Append new book

        return jsonify({'message': 'Book added successfully', 'book': new_book}), 201
    except Exception as e:
        return jsonify({'error': f'An error: {str(e)}'}), 500
```

Form lato HTML

```
<h1>Libri</h1>
<h1>Add a Book</h1>
<form id="bookForm" onsubmit="submitForm(event)">
  <label for="title">Title:</label>
  <input type="text" id="Title" name="Title" required><br><br>

  <label for="author">Author:</label>
  <input type="text" id="Author" name="Author" required><br><br>

  <button type="submit">Add Book</button>
</form>
```

Gestione del form in Javascript

```
{% block content %}  
<script>  
  async function submitForm(event) {  
    event.preventDefault(); // Prevent form from refreshing the page  
  
    const Title = document.getElementById('Title').value;  
    const Author = document.getElementById('Author').value;  
  
    const response = await fetch('/api/books', {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json'  
      },  
      body: JSON.stringify({Title, Author})  
    });  
  
    const result = await response.json();  
    if (response.ok) {  
      alert('Book added successfully!');  
    } else {  
      alert('Error: ${result.error}');  
    }  
  }  
</script>  
  
<h1>Libri</h1>
```

Esempio

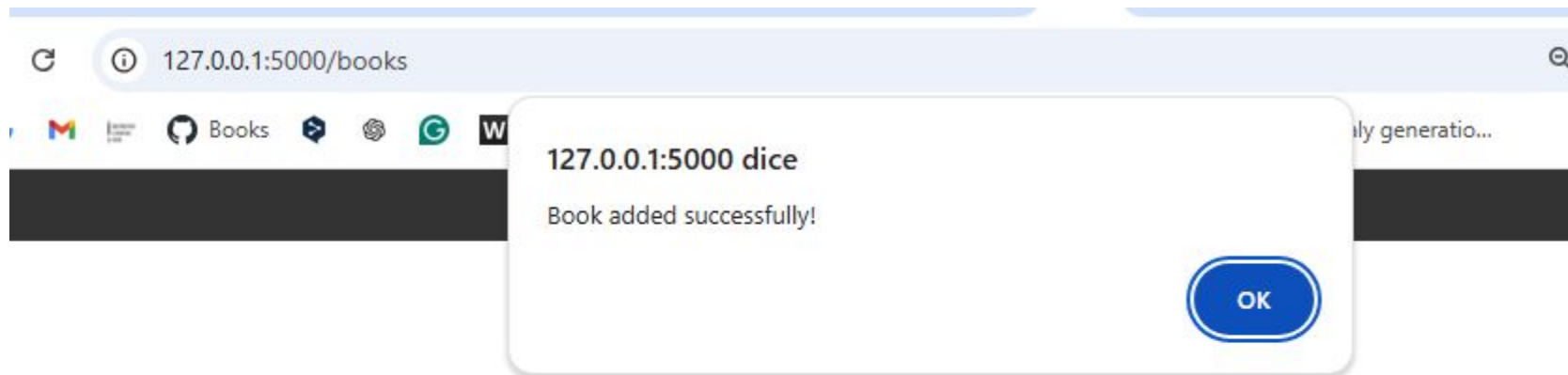
Aggiungi un libro

Title:

Author:

Titolo	Autore	Genere	Altezza	Editore	Azioni
Fundamentals of Wavelets	Goswami, Jaideva	signal_processing	228	Wiley	
Data Smart	Foreman, John	data_science	235	Wiley	
God Created the Integers	Hawking, Stephen	mathematics	197	Penguin	
Superfreakonomics	Dubner, Stephen	economics	179	HarperCollins	
Orientalism	Said, Edward	history	197	Penguin	
Nature of Statistical Learning Theory, The	Vapnik, Vladimir	data_science	230	Springer	
Integration of the Indian States	Menon, V P	history	217	Orient Blackswan	
Drunkard's Walk, The	Mlodinow, Leonard	science	197	Penguin	
Image Processing & Mathematical Morphology	Shih, Frank	signal_processing	241	CRC	
How to Think Like Sherlock Holmes	Konnikova, Maria	psychology	240	Penguin	
Data Scientists at Work	Sebastian Gutierrez	data_science	230	Apress	

Esempio



Aggiungi un libro

Title:

Author:

Titolo	Autore	Genere	Altezza	Editore	Azioni
Fundamentals of Wavelets	Gopinath, Jaidou	signal processing	328	Wiley	

Esempio

Aggiungi un libro

Title:

Author:

Add Book

Titolo	Autore	Genere	Altezza	Editore	Azioni
Fundamentals of Wavelets	Goswami, Jaideva	signal_processing	228	Wiley	
Data Smart	Foreman, John	data_science	235	Wiley	
God Created the Integers	Hawking, Stephen	mathematics	197	Penguin	
Superfreakonomics	Dubner, Stephen	economics	179	HarperCollins	
Orientalism	Said, Edward	history	197	Penguin	
Nature of Statistical Learning Theory, The	Vapnik, Vladimir	data_science	230	Springer	
Integration of the Indian States	Menon, V P	history	217	Orient Blackswan	
Drunkard's Walk, The	Mlodinow, Leonard	science	197	Penguin	
Image Processing & Mathematical Morphology	Shih, Frank	signal_processing	241	CRC	
How to Think Like Sherlock Holmes	Konnikova, Maria	psychology	240	Penguin	
Data Scientists at Work	Sebastian Gutierrez	data_science	230	Apress	
aa	cc				