

The background is a dark navy blue. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the bottom-left corner, there is a circular inset showing a detailed, high-contrast image of a circuit board. In the top-right corner, there is a faint, grey, 3D-rendered pattern of interlocking cubes or a circuit layout. The word "Booksharing" is centered in the middle-right area in a white, sans-serif font.

Booksharing

Introduzione

Sviluppo di una webapi per il booksharing nell'atrio dell'istituto con lo scopo di fornire agli studenti un servizio di consultazione e valutazione dei libri messi a disposizione



Flask

```
server.py > ricerca
1  from flask import Flask, render_template, redirect, url_for, request, jsonify
2  from flask_bootstrap import Bootstrap
3  import sqlite3
4  import webbrowser
5
6  app = Flask(__name__)
7  Bootstrap(app)
8  db = 'data.db'
```

```
if __name__ == "__main__":
    app.run()
```

Modifica la definizione della tabella

Tabella

Libri

Avanzate

Campi

Aggiungi campo

Rimuovi campo

Muovi campo in sù

Muovi campo in giù

Nome	Tipo	NN	CP	AI	U	Default
idL	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
titolo	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
autore	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
anno	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
genere	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```

1 CREATE TABLE "Libri" (
2     "idL" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
3     "titolo" TEXT NOT NULL,
4     "autore" TEXT,
5     "anno" INTEGER,
6     "genere" TEXT
7 );
    
```

Annulla

OK

```
andrea@inf-carpinato:~/Scrivania/tesinal.1$ /usr/bin/python3 /home/andrea/Scrivania/tesinal.1/server.py
```

```
* Serving Flask app "server" (lazy loading)
```

```
* Environment: production
```

```
WARNING: This is a development server. Do not use it in a production deployment.
```

```
Use a production WSGI server instead.
```

```
* Debug mode: off
```

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
127.0.0.1 - - [04/Jun/2020 23:49:04] "GET / HTTP/1.1" 200 -
```

```
127.0.0.1 - - [04/Jun/2020 23:49:05] "GET /static/books2.jpg HTTP/1.1" 304 -
```

```
[(1, 'Il ritratto di Dorian Grey', 'Oscar Wilde', 1898, None), (2, 'Il codice Da Vinci', 'Dan Brown', 2003, 'romanzo'), (3, 'dracula', 'Stoker', 1897, 'romanzo gotico'), (9, 'Pensieri', 'Blaise Pascal', 1894, 'filosofia')]
```

```
127.0.0.1 - - [04/Jun/2020 23:49:07] "GET /allbooks HTTP/1.1" 200 -
```

Il risultato



JSON	Dati non elaborati	Header
Salva	Copia	Comprimi tutto Espandi tutto Filtra JSON
▼ 0:		
0:	1	
1:	"Il ritratto di Dorian Grey"	
2:	"Oscar Wilde"	
3:	1898	
4:	null	
▼ 1:		
0:	2	
1:	"Il codice Da Vinci"	
2:	"Dan Brown"	
3:	2003	
4:	"romanzo"	
▼ 2:		
0:	3	
1:	"dracula"	
2:	"Stoker"	
3:	1897	
4:	"romanzo gotico"	
▼ 3:		
0:	9	
1:	"Pensieri"	
2:	"Blaise Pascal"	
3:	1894	
4:	"filosofia"	



Home & Visualizza i libri

```
@app.route("/",methods=['GET', 'POST'])
def home():
    return render_template("index.html")

@app.route("/allbooks",methods=['GET', 'POST'])
def visualizzaTutto():
    conn=sqlite3.connect(db)
    c=conn.cursor()
    c.execute("SELECT * FROM Libri")
    rows = c.fetchall()
    c.close()
    conn.close()
    # name=#read from db sqlite
    print(rows)
    return jsonify(rows)
```

render_template serve per collegare la pagina html con la funzione flask
Nel marcatore bisogna sempre mettere i metodi che si usano altrimenti non può svolgere le richieste

```

@app.route('/addbook',methods=['GET','POST'])
def aggiungi():
    error=None
    if request.method == 'POST':
        titolo=request.form['titolo']
        autore=request.form['autore']
        anno=request.form['anno']
        genere=request.form['genere']
        conn=sqlite3.connect(db)
        c=conn.cursor()
        #print(f'INSERT INTO Libri("title","author","year_published") VALUES ("{title}", "{author}", "{year}")')
        c.execute(f'INSERT INTO Libri("titolo","autore","anno","genere") VALUES ("{titolo}", "{autore}", "{anno}", "{genere}")')
        c.execute('commit') #salva modifiche db
        c.close()
        conn.close()
        return render_template('index.html')
    else :
        return render_template("agglibro.html")

```

Per la lettura dei valori dal file html si usa request.form['identificativo']
 Bisogna sempre chiudere il database altrimenti è impossibile lavorarci sopra

```

<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
    <!-- jQuery library -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <!-- Popper JS -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <!-- Latest compiled JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
  </head>
  <body>
    <!-- A grey horizontal navbar that becomes vertical on small screens -->
    <nav class="navbar navbar-expand-sm bg-light">

      </nav>
      <div class="container-fluid" align='center'>
        <h1>Booksharing</h1>
        
          <a href="/allbooks"><button type="button" class="btn btn-primary">Visualizza tutti i libri</button></a>
          <!--<a href="/research"><button type="button" class="btn btn-primary">Ricerca</button></a>-->
          <a href="/addbook"><button type="button" class="btn btn-primary">Inserisci libro</button></a>
        </div>
      </div>
    </body>

```

Per l'indirizzo dell'immagine usiamo la funzione `url_for` di Flask appositamente creata per i file media del sito presenti nella cartella *Static*. I file html vanno invece nella cartella *Templates*.



Query

```
c.execute("SELECT * FROM Libri")
```

```
c.execute(f'INSERT INTO Libri("titolo","autore","anno","genere") VALUES ("{titolo}", "{autore}", "{anno}", "{genere}")')  
c.execute('commit') #salva modifiche db
```

```
c.execute(f"SELECT * FROM Libri WHERE Libri.titolo={ricerca}")
```



Sviluppi futuri

- Creazione di nuove funzioni(es recensioni,acquisto,vendita)
- Possibilità di condividere anche i film
- Possibilità di fare recensioni su opere d'arte o musei
- sviluppo di un'applicazione per dispositivi mobili
- connessione con altri social network