

# ***your favorite category***

Autore:Andrea Altomare



***link***

***•[Git]***

***👉 [Netlify(prova l'app)]***

***•[readme]***

# *Presentazione della home*

## Search Your Book Category

Inserisci il testo

Near

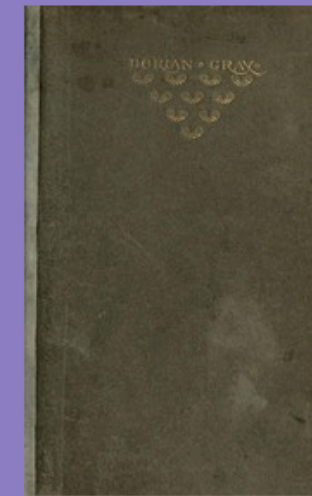
## Search Your Book Category

horror

Near

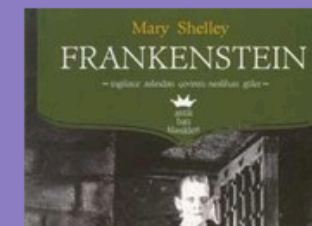
### The Picture of Dorian Gray

Autore: Oscar Wilde, Oscar Wilde, Oscar Wilde



### Frankenstein or The Modern Prometheus

Autore: Mary Shelley



# codice html

```
<!DOCTYPE html>
<html lang="it">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="your favorite category" />
    <meta name="keywords" content="HTML, CSS, JavaScript,webpack" />
    <meta name="author" content="Andrea Altomare" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta property="og:title" content="Junior Front End Developer Cosenza" />
    <meta
      property="og:description"
      content="puoi cercare la tua categoria di libri preferiti"
    />
    <meta
      property="og:url"
      content="https://677fd87f6011d97262e68329—cercalatuacategoriapreferita.netlify.app"
    />
    <meta property="og:type" content="website" />
    <title>your favorite category</title>
    <link rel="stylesheet" href="/src/assets/css/stile.css" />
  </head>
  <body>
    <main>
      <header>
        <h1>Search Your Book Category</h1>
        <form id="searchForm">
          <input type="text" id="inputField" placeholder="Inserisci il testo" />
          <button type="submit" id="button">Near</button>
        </form>
      </header>
      <div id="resultDiv"></div>
      <div id="apilistDiv"></div>
      <div id="bookDetailsDiv"></div>
    </main>
    <script src="/dist/bundle.js"></script>
    <script src="/assets/js/index.js"></script>
  </body>
</html>
```

# codice webpack

```
const path = require("path");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const MiniCssExtractPlugin = require("mini-css-extract-plugin"); // 207.8k (gzipped: 53.4k)

module.exports = {
  mode: "development", // Usa 'development' o 'production' a seconda del tuo ambiente
  entry: "./src/assets/js/index.js", // Percorso del file di ingresso
  output: {
    filename: "bundle.js",
    path: path.resolve(__dirname, "dist"),
    publicPath: "/", // Necessario per il dev server
  },
  module: {
    rules: [
      {
        test: /\.js$/, // Regola per file JS
        exclude: /node_modules/,
        use: {
          loader: "babel-loader",
          options: {
            presets: ["@babel/preset-env"],
          },
        },
      },
      {
        test: /\.css$/, // Regola per file CSS
        use: [
          MiniCssExtractPlugin.loader, // Estrae CSS in file separati
          "css-loader",
        ],
      },
    ],
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: "./src/index.html", // Percorso del template HTML
      favicon: false, // Disabilita la favicon
    }),
    new MiniCssExtractPlugin({
      filename: "stiles.css", // Nome del file CSS
    }),
  ],
  devtool: "source-map", // Per il debugging
  devServer: {
    static: path.join(__dirname, "dist"),
    compress: true,
  },
}
```

# codice js

```
import "/src/assets/css/stile.css";

alert(
  "AVVISO IMPORTANTE:se clicchi sul libro desiderato, avrai le info del libro"
);

Complexity is 35 Bloody hell...
document.addEventListener("DOMContentLoaded", () => {
  const button = document.getElementById("button");
  const resultDiv = document.getElementById("resultDiv");
  const inputElement = document.getElementById("inputField");

  Complexity is 15 You must be kidding
  const performSearch = async () => {
    const inputText = inputElement.value.trim();
    resultDiv.textContent = "Caricamento...";

    if (!inputText) {
      resultDiv.textContent = "Per favore inserisci una categoria.";
      return;
    }

    try {
      const response = await fetch(
        `https://openlibrary.org/subjects/${inputText}.json`
      );
      if (!response.ok) {
        throw new Error(`Errore nella fetch! Status: ${response.status}`);
      }

      const data = await response.json();

      if (!data.works || data.works.length === 0) {
        resultDiv.textContent = "Nessun libro trovato in questa categoria.";
        return;
      }

      resultDiv.innerHTML = "";
      Complexity is 5 Everything is cool!
      data.works.forEach((book) => {
        const bookItem = document.createElement("div");
        bookItem.classList.add("book-item");
        bookItem.innerHTML = `
          <h3>${book.title}</h3>
          <p>Autore: ${
            book.authors
              ? book.authors
                .map((author) => author.name || "Autore sconosciuto")
          }
        `;
      });
    } catch (error) {
      console.error("Errore durante la fetch:", error);
    }
  };

  button.addEventListener("click", performSearch);
});
```

# **PRESENTAZIONE**

**vi presento il mio progetto di js advanced!**

**Applicazione per la ricerca di libri nel catalogo "Open Library",  
sviluppata come progetto finale del corso Javascript Advanced in  
Start2Impact.**

- istruzioni dell'app: Selezionate la vostra categoria preferita e visualizzerete un elenco di libri. Sceglietene uno, cliccateci sopra e verranno mostrati il titolo e la descrizione del libro.**

## • SPIEGAZIONE DELL'APP

**L'applicazione funziona tramite una barra di ricerca, simile a quella di Google, che permette agli utenti di cercare libri selezionando una categoria specifica (ad esempio, "Fantasia"). Una volta inserita la categoria, l'utente avvia la ricerca cliccando sul pulsante "near". Al termine della ricerca, l'applicazione visualizza nella sezione "Risultati" un elenco di libri con dei titoli e dei rispettivi autori. Cliccando su un libro, l'utente può consultare la descrizione del libro .**

**FINE**