



WEB-APP BRESCIA CITY

INTRODUZIONE:

La web-app che abbiamo sviluppato permette ai turisti che vogliono venire a visitare Brescia di avere una breve introduzione con le principali informazioni riguardanti Brescia, inoltre mettiamo a disposizione 3 mappe:

Una mappa per le ZTL presenti a Brescia, questa ci mostrerà le aree delle ZTL di Brescia evidenziate e con dei marker ci mostrerà i varchi di esse.

Una mappa per gli Hot-Spot Wi-Fi presenti a Brescia e questa ci mostrerà le aree di Brescia dove è presente la connessione Wi-Fi utilizzabile gratuitamente.

Tutti gli Hot-Spot sono segnalati da appositi cartelli informativi che indicano le modalità di accesso al servizio Wi-Fi, mentre per ottenere la password di accesso basta inviare un SMS al numero +39 342 4112665. Dopo aver effettuato la prima registrazione il dispositivo si collega automaticamente alla rete "Brescia Wi-Fi".

Una mappa complessiva che permette di vedere dei posti da visitare, dei posti dove mangiare, dei posti dove dormire e dei posti dove parcheggiare tramite dei marker, e utilizzando un selettore si può selezionare cosa far vedere sulla mappa.

ANALISI/SPECIFICA DEI REQUISITI:

Per l'analisi/specifica dei requisiti ci siamo affidati a Trello, un software gestionale in stile Kanban basato sul web.

Ecco il link per vedere il modello sviluppato:

<https://trello.com/invite/b/5uFayLXx/ATTI7a48fc968dab9a335db01f64c5501d4243E71323/mappa-interattiva-utilita>

DESIGN DELL'INTERFACCIA UTENTE:

Per il design dell'interfaccia utente abbiamo utilizzato Figma, un editor di grafica vettoriale e uno strumento di prototipazione principalmente basato sul Web.

Ecco il link per vedere un'anteprima dell'interfaccia utente:

<https://www.figma.com/file/c2AfAPLjzS0RXpeL3X1mq/BRESCIA-CITY?type=design&node-id=0%3A1&mode=design&t=Vum8ybxEpoXiXIDR-1>

ARCHITETTURA DELLA WEB-APP:

La web-app è strutturata su 4 pagine principali:

- Homepage: contenente la descrizione di Brescia e i link per le varie mappe.
- Mappa complessiva: contenente la mappa con i marker dei punti d'interesse, il menu per la selezione dei filtri della mappa e un collegamento per tornare alla homepage.
- Mappa ZTL: contenente la mappa con le aree delle ZTL, i marker dei varchi e il link per tornare alla homepage.
- Mappa Hot-Spot: contenente la mappa con i marker dove è presente l'Hot-Spot Wi-Fi e il link per tornare alla homepage.

Nelle varie pagine è stata sviluppata una sidebar che permette la navigazione tra le varie pagine.

IMPLEMENTAZIONE DELLA WEB-APP:

Per lo sviluppo della web-app abbiamo utilizzato vari linguaggi di programmazione come: HTML, CSS e Javascript. Per l'implementazione delle mappe, delle aree e dei marker abbiamo utilizzato [Leaflet](#), una libreria JavaScript per sviluppare mappe geografiche interattive.

Anche per fare il menu di selezione dei marker della mappa complessiva abbiamo utilizzato Leaflet con la sua funzione dei layers.

Per la mappa delle ZTL abbiamo utilizzato 2 dataset diversi, uno per i varchi e uno per le aree, mentre per la mappa dell'Hot-Spot Wi-Fi abbiamo utilizzato un altro dataset.

DATASET MAPPA ZTL:

<https://www.dati.lombardia.it/resource/ufkd-whtz.json> (AREE)

<https://www.dati.lombardia.it/resource/ufkd-whtz.json> (VARCHI)

DATASET MAPPA HOT-SPOT WI-FI:

<https://www.dati.lombardia.it/resource/nkxw-x4xd.json>

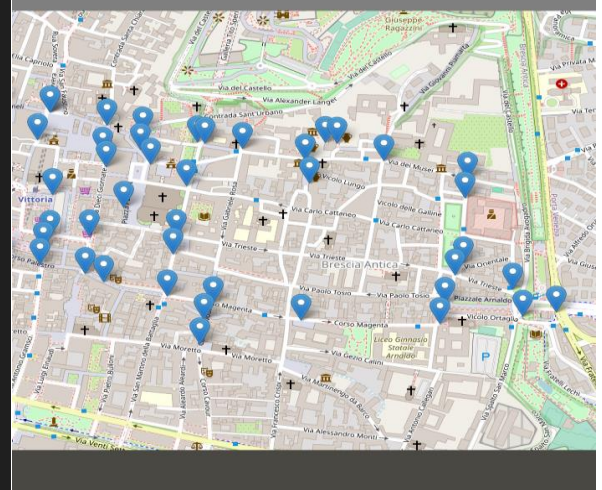
PROGRAMMAZIONE DELLA WEB-APP:

Per la programmazione della mappa degli Hot-Spot abbiamo utilizzato le funzioni fetch e await per prendere i dati dal dataset.

```
Mappa HotSpot > JS scriptHotSpot.js > fetchData
1  var map = L.map('map').setView([45.53971139337194, 10.221328339942707], 18);
2
3  L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
4    maxZoom: 19,
5    attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
6  }).addTo(map);
7
8  fetchData();
9
10 async function fetchData() { FETCH
11   try {
12     // using await outside an async function is only allowed in a module
13     const response = await fetch("https://www.dati.lombardia.it/resource/nkxw-x4xd.json");
14
15     if (!response.ok) {
16       throw new Error('HTTP error: ${response.status}');
17     }
18
19     const data = await response.json(); AWAIT
20
21     let latitudine = 0;
22     let longitudine = 0;
23     let popup = "";
24     let attivo = "Si";
25
26     for (let index = 0; index < data.length; index++) {
27       latitudine = data[index].latitudine;
28       longitudine = data[index].longitudine; Scorrimeto del dataset e
29
30       if (data[index].attivo == 'si') ricavo delle coordinate
31         attivo = "Si";
32       else
33         attivo = "No";
34
35       popup =
36         "ID: " + data[index].id_accesso + "<br>" +
37         "Dove: " + data[index].sito + "<br>" + Creazione dei marker/popup
38         "Attivo: " + attivo;
39
40       L.marker([latitudine, longitudine]).addTo(map); Aggiunta dei marker/popup
41       .bindPopup(popup) sulla mappa
42       .openPopup();
43
44       popup = "";
45     }
46   } catch (error) {
47     console.error("Could not get products: ${error}");
48   }
49 }
```

RISULTATO

HOTSPOT WIFI



Quando sulla mappa puntiamo il mouse su uno dei marker questo ci fornisce le principali informazioni di quel marker.

Stesso procedimento per la mappa delle ZTL con la differenza che invece di avere solo una funzione con la fetch e await, abbiamo 2 funzioni con 2 fetch e await per i 2 diversi dataset.

Essendo che il dataset conteneva dei testi con alcuni caratteri speciali che non venivano visualizzati abbiamo fatto una funzione che sostituiva il testo non leggibile in un testo leggibile.

Mappa ZTL > JS scriptZTL.js > fetchDatiZTL

```
1  var map = L.map('map').setView([45.53971139337194, 10.221328339942707], 18);
2
3  L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
4      maxZoom: 19,
5      attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
6  }).addTo(map);
7
8  //Fetch Dati, Aggiunta Poligoni ZTL e Marker Varchi
9  fetchDatiVarchi();
10 fetchDatiZTL();
11
12 async function fetchDatiZTL() {
13     try {
14         //Fetch Dati
15         const response = await fetch(
16             "https://www.dati.lombardia.it/resource/ufkd-whtz.json",
17         );
18         if (!response.ok) {
19             throw new Error(`HTTP error: ${response.status}`);
20         }
21
22         //Data contiene i dati del DataSet
23         const data = await response.json();
24
25         let coordinate = "";
26         for(let i = 0; i < data.length; i++) {
27
28             //Recupero Coordinate
29             for (let j=0; j < data[i].the_geom.coordinates[0][0].length; j++){
30
31                 if (j == (data[i].the_geom.coordinates[0][0].length-1)){
32                     coordinate += "[" + data[i].the_geom.coordinates[0][0][j][1] + ", " +
33                     data[i].the_geom.coordinates[0][0][j][0] + "]"
34                 } else {
35                     coordinate += "[" + data[i].the_geom.coordinates[0][0][j][1] + ", " +
36                     data[i].the_geom.coordinates[0][0][j][0] + "], "
37                 }
38             }
39
40             //Aggiunta Poligono alla Mappa
41             coordinate = "var polygon = L.polygon([" + coordinate + "], {fillOpacity: 0.1,}).addTo(map);"
42             eval(coordinate);
43
44             polygon.bindPopup(
45                 "Periodo: " + data[i].periodo + "<br>" +
46                 "Nome ZTL: " + data[i].nome_ztl + "<br><br>" +
47                 "Dettagli: " + data[i].dettagli
48             );
49
50             coordinate = "";
51         }
52     } catch (error) {
53         console.error(`Could not get data: ${error}`);
54     }
55 }
56
```

Codice contenente la prima fetch con la creazione delle aree sulla mappa ↑

```

57 async function fetchDatiVarchi() {
58     try {
59         const response = await fetch("https://www.dati.lombardia.it/resource/jffv-3yqx.json");
60
61         if (!response.ok) {
62             throw new Error(`HTTP error: ${response.status}`);
63         }
64
65         const data = await response.json();
66
67         var greenIcon = new L.Icon({
68             iconUrl: 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-icon-2x-green.png',
69             shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
70             iconSize: [25, 41],
71             iconAnchor: [12, 41],
72             popupAnchor: [1, -34],
73             shadowSize: [41, 41]
74         });
75
76         for (let index = 0; index < data.length; index++) {
77             const latitudine = parseFloat(data[index].latitudine);
78             const longitudine = parseFloat(data[index].longitudine);
79
80             let deroghe = data[index].deroghe;
81             deroghe = caratteriSpecialiDeroghe(data, index, deroghe);
82             deroghe.replace(/(?:\r\n|\r|\n)/g, '<br>');
83             deroghe.replace(";", "<br>");
84
85             const popup =
86                 "Via: " + data[index].via + "<br>" +
87                 "Tipo: " + data[index].tipo + "<br>" +
88                 "Deroghe: " + deroghe;
89
90             L.marker([latitudine, longitudine], {icon: greenIcon}).addTo(map)
91                 .bindPopup(popup)
92                 .openPopup();
93
94         }
95     } catch (error) {
96         console.error(`Could not get data: ${error}`);
97     }
98 }
99

```

Codice contenente la seconda fetch con la creazione dei marker sulla mappa↑

```

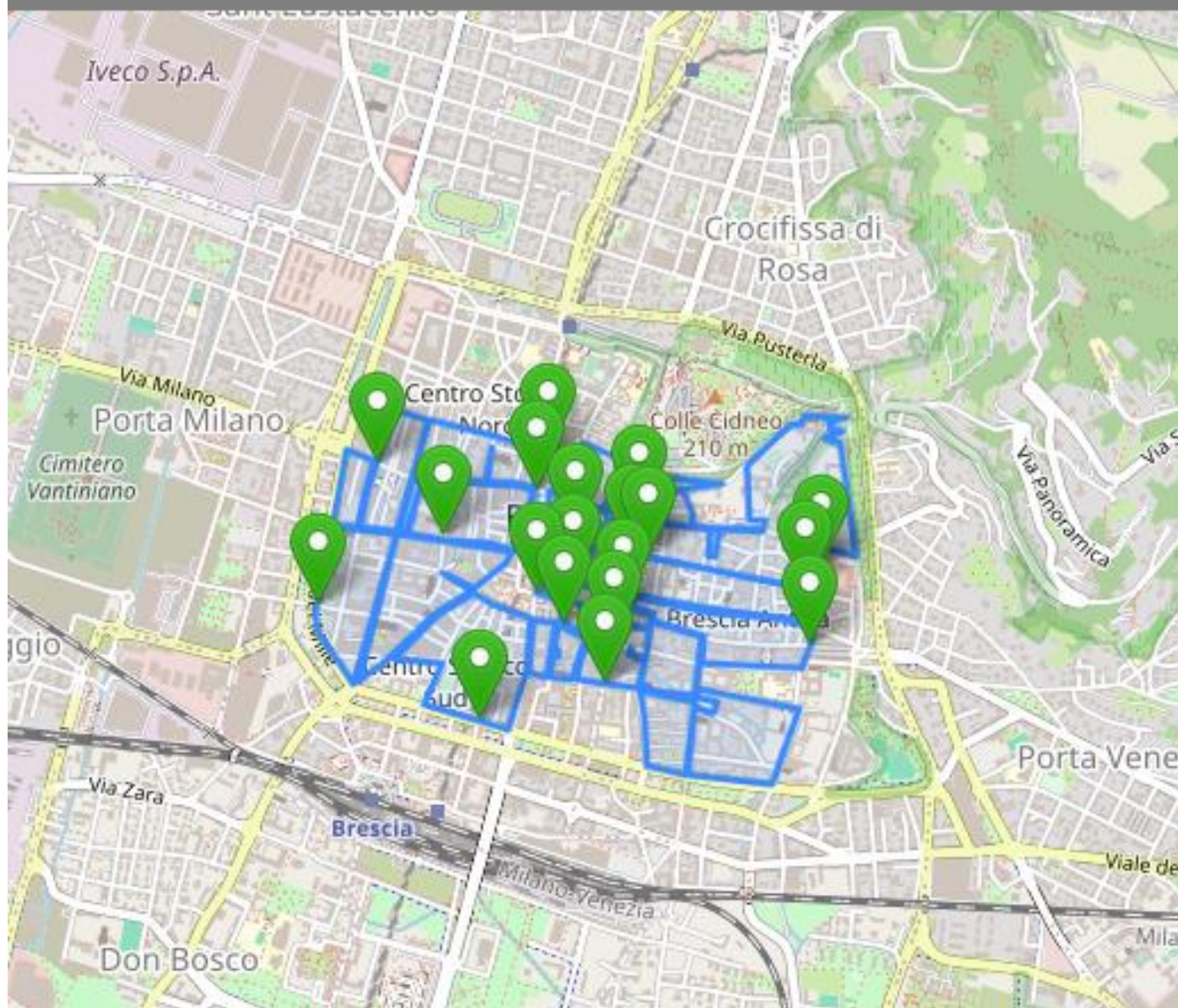
// Funzione per togliere manualmente i caratteri speciali dalle Deroghe dei Varchi
function caratteriSpecialiDeroghe(data, index, deroghe) {
    switch (data[index].via) {
        case "Via Musei":
            deroghe = "Possibile attivare il titolo Gratta e Sosta;Orari di carico
            break;
        case "Via Trieste":
            deroghe = "Area a Pedonalità privilegiata;Utilizzabile ticket Gratta e
            break;
        case "Via X Giornate/Loggia":
            deroghe = "Area Pedonale;Utilizzabile ticket Gratta e Sosta;L'accesso è
            break;
        case "Corso Zanardelli (Est)":
            deroghe = "Area Pedonale non utilizzabile ticket Gratta e Sosta;L'acces
            break;
        case "Via Querini":
            deroghe = "Area a Pedonalità privilegiata;Utilizzabile ticket Gratta e
            break;
        case "Corso Zanardelli (Ovest)":
            deroghe = "Area Pedonale;Utilizzabile ticket Gratta e Sosta;L'accesso è
            break;
        default:
            deroghe = data[index].deroghe;
            break;
    }
    return deroghe;
}

```

Funzione per la correzione dei testi non leggibili↑

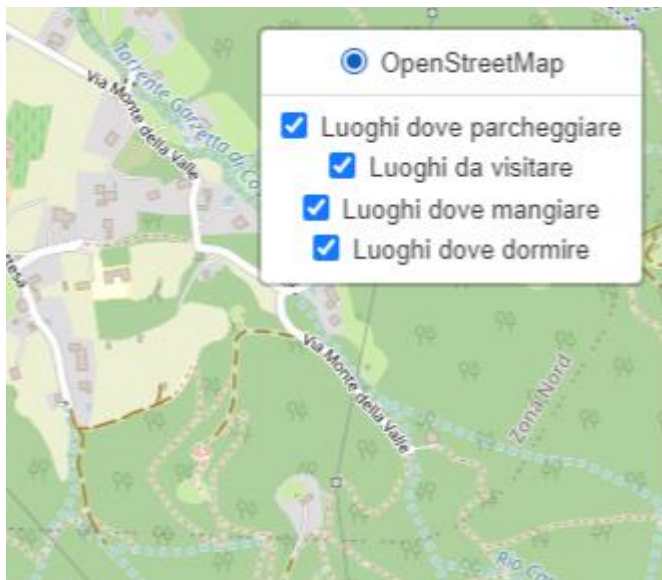
RISULTATO

ZTL



Anche in questa mappa quando sulla mappa puntiamo il mouse su uno dei marker questo ci fornisce le principali informazioni di quel marker.

A detailed map of the Brescia area, showing the city center and surrounding hills. The map includes street names, landmarks like the Basilica of San Faustino, and various icons for points of interest. A callout box highlights Piazza Arnaldo, describing it as a popular spot for aperitifs and socializing, and includes a small photo of the square.



← Menu di selezione dei marker



HOME PAGE

MAPPA COMPLESSIVA

MAPPA ZTL

MAPPA HOSTSPOT



Brescia è una città della Lombardia, situata nel nord Italia. È la seconda città più grande della regione, dopo Milano, e conta circa 200.000 abitanti.

Brescia è una città ricca di storia e cultura, con un centro storico ben conservato. Tra i monumenti più importanti si possono menzionare:

- Il Duomo, una cattedrale romanica del XII secolo;
- Il Museo di Santa Giulia, che ospita reperti archeologici e artistici dal Paleolitico all'età moderna;
- La Torre dell'Orologio, una torre medievale con un quadrante astronomico;
- Il Teatro Grande, un teatro dell'opera del XIX secolo.

Brescia è anche una città industriale e commerciale.

L'economia di Brescia si basa anche sul turismo. La città è una meta popolare per i turisti italiani e stranieri, che vengono attratti dalla sua storia, cultura e natura.

I turisti che visitano Brescia possono scegliere tra diverse attività, tra cui:

- Visitare i monumenti e i musei della città;
 - Passeggiare per il centro storico;
 - Fare shopping;
 - Visitare le località turistiche nei dintorni della città;
 - Partecipare agli eventi culturali e sportivi che si svolgono in città durante tutto l'anno.
- Brescia è una città che offre qualcosa per tutti i gusti. È una destinazione ideale per un weekend o una vacanza più lunga.

Ecco alcuni consigli per i turisti che visitano Brescia:

- La città è facilmente raggiungibile in auto, treno o aereo.
- Il centro storico è pedonale, quindi è consigliabile indossare scarpe comode.
- Brescia è una città sicura, ma è sempre bene fare attenzione ai propri effetti personali.
- Il periodo migliore per visitare Brescia è la primavera o l'autunno, quando il clima è mite.

Buona visita a Brescia!

WBS:

Progetto WebApp Brescia

-Inizializzazione del progetto

- Definizione degli Obiettivi del Progetto

- Analisi dei Requisiti

- Identificazione delle Parti Interessate

-Pianificazione del progetto

- Creazione del Pianificazione del Progetto

- Ricerca delle Risorse

- Definizione del Calendario del Progetto

-Progettazione

- Progettazione dell'Interfaccia Utente (UI)

 - Creazione di Mockup

 - Creazione di Prototipi

- Progettazione dell'Architettura del Frontend

 - Scelta del Framework

 - Definizione delle Componenti Principali

-Sviluppo

- Configurazione dell'Ambiente di Sviluppo

- Implementazione delle Componenti Frontend

 - Header

 - Footer

 - Pagina Principale

 - Altre Pagine

- Implementazione delle Funzionalità Interattive

 - Caricamento Dati Asincrono

 - Animazioni

-Test

- Testing Unitario delle Componenti

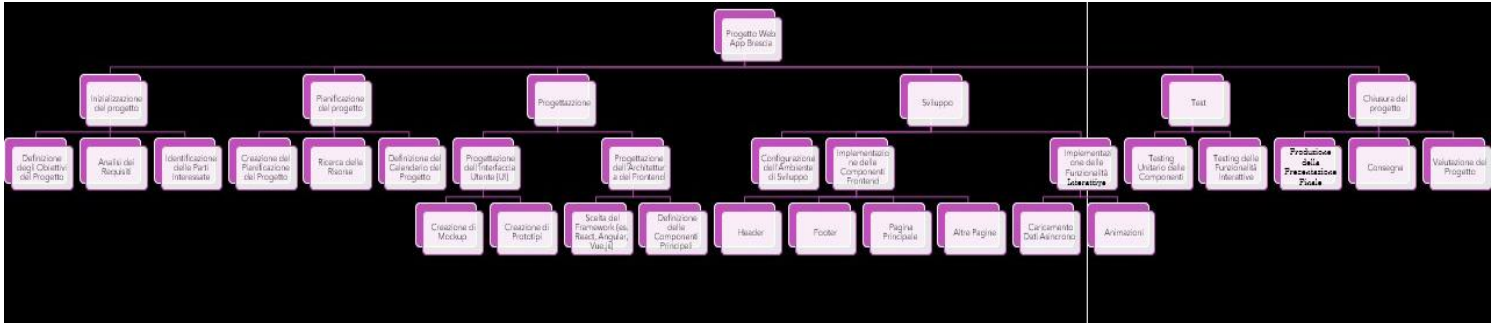
- Testing delle Funzionalità Interattive

-Chiusura del progetto

- Produzione della Presentazione Finale

- Consegna

- Valutazione del Progetto



GANTT:

