

Meteo

Matilde Labruzzo 987069

PWM - secondo semestre A.A. 2021/2022

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Analisi dei requisiti . . . . .	2
1.1.1	Destinatari . . . . .	2
1.1.2	Modello di valore . . . . .	2
1.1.3	Flusso dei dati . . . . .	2
1.1.4	Aspetti tecnologici . . . . .	2
<b>2</b>	<b>Interfacce</b>	<b>4</b>
2.1	Index.ejs . . . . .	4
<b>3</b>	<b>Architettura</b>	<b>5</b>
<b>4</b>	<b>Codice</b>	<b>6</b>
<b>5</b>	<b>Sviluppi</b>	<b>7</b>
<b>6</b>	<b>Bibliografia</b>	<b>8</b>

# Capitolo 1

## Introduzione

### 1.1 Analisi dei requisiti

#### 1.1.1 Destinatari

Grazie a una interfaccia semplice ed intuitiva, qualsiasi tipo di utente può accedere e utilizzare facilmente l'applicazione, senza alcun tipo di livello di esperienza richiesto. Infatti, tramite la barra di ricerca, l'utente può cercare e trovare le informazioni metereologiche riguardanti la città desiderata. Inoltre, sempre attraverso la barra di ricerca, nella sezione autenticazione, può accedere a maggiori servizi, come ad esempio personalizzare le città preferite.

Nella versione attuale, non sono posti particolari vincoli di banda grazie alla sola presenza degli elementi utili alla navigazione, riducendo così la latenza il più possibile. Inoltre, le immagini relative alle condizioni metereologiche delle città visualizzate saranno caricate in modalità asincrona, riducendo al minimo il periodo in cui la pagina non risponde agli input dell'utente.

Si consiglia la navigazione via PC per una maggiore semplicità di lettura, ma è comunque possibile utilizzare un telefono grazie alla responsività degli elementi presenti nelle pagine.

Gli utenti che accedono alla piattaforma web sono spinti da motivazioni prettamente personali, in particolare dalla necessità di cercare informazioni metereologiche riguardo a una o più città. Di conseguenza, l'applicazione è progettata in modo tale da fornire le informazioni su richiesta esplicita dell'utente, attraverso la barra di navigazione o l'area personale, previa registrazione e login, in cui viene data la possibilità di salvare le città preferite.

#### 1.1.2 Modello di valore

#### 1.1.3 Flusso dei dati

#### 1.1.4 Aspetti tecnologici

La trasmissione di dati può essere effettuata in chiaro ad eccezione della password usata per accedere alla propria area personale, la quale viene inviata sotto forma di SHA256 per evitare che sia leggibile.

Il database prevede la creazione di una singola collezione in cui ogni documento contiene:

- *\_id*: id univoco assegnato dal database in automatico;
- *user*: username;
- *email*: email dell'utente;
- *pwd*: password dell'utente sotto forma di digest SHA256;

- *pref*: array contenente le città messe tra i preferiti dall'utente; può essere vuoto.

Tecnologie utilizzate:

- *HTML 5 e bootstrap*: realizzazione della struttura delle pagine e gestione degli stili con bootstrap;
- *JavaScript*: realizzazione delle richieste al server e alle API, nonché del toggle della darkmode e l'aggiornamento di frammenti di pagina con dati ricevuti dal server;
- *Node JS*: implementazione del server con tutte le sue funzionalità (caricamento di tutte le pagine, invio di oggetti JSON per aggiornare frammenti di pagina, interrogazione dell'API per ottenere il nome delle città e delle aree geografiche);
- *Express*: framework utilizzato per semplificare il deploy del server;
- *JSON*: formato usato per la trasmissione dei dati;
- *localStorage*: usato per il salvataggio della preferenza utente per la visualizzazione della pagina (dark o light mode)
- *MongoDB*: database utilizzato per salvare le informazioni relative agli utenti registrati sull'app;
- *API*: nel progetto sono state utilizzate 4 API:
  - *openWeather*: API usata per ottenere le condizioni meteo in tempo reale;
  - *pexels*: API utilizzata per reperire le immagini mostrate nelle pagine;
  - *spott*: API utilizzata per ottenere i nomi delle città mondiali e per permettere la ricerca incrementale;
  - *rest countries*: API utilizzata per ottenere i nomi delle capitali delle diverse aree geografiche.

## Capitolo 2

# Interfacce

Per realizzare le pagine sono stati usati *HTML 5* e *bootstrap*.

Gli elementi di personalizzazione, come il nome utente nella navbar una volta autenticati o il meteo in tempo reale delle città salvate tra i preferiti, sono inseriti utilizzando *EJS*.

### 2.1 Index.ejs

Nella parte superiore dell'index come prima cosa troviamo una navbar semplice con elementi completamente intuitivi: a partire da sinistra si ha il nome e il logo dell'applicazione, e il tasto home; grazie al menù a tendina *Aree* è possibile navigare per aree geografiche (i continenti) e cercare informazioni sulle città relative ad esse; cliccando invece sul menù Autenticazione l'utente può accedervi o registrarsi; a seguito del login, questo menù viene sostituito con un altro per poter effettuare il logout e per poter accedere all'area personale, dove sono salvate le città preferite dall'utente e dove quest'ultimo può cambiare le informazioni di accesso. Sempre sulla navbar troviamo un toggle che dà la possibilità all'utente di cambiare la modalità di visualizzazione, *light mode* o *dark mode* a seconda della sua preferenza. Sulla destra troviamo invece una barra di ricerca.

Al di sotto della navbar troviamo

## Capitolo 3

# Architettura

Capitolo 4

Codice

Capitolo 5

Sviluppi



## Capitolo 6

# Bibliografia

- API pexels: <https://www.pexels.com>;
- API openWeather: <https://www.pexels.com>;
- API città: <https://www.spott.dev/>;
- API paesi e capitali: <https://restcountries.com/>;
- Libreria usata per generare digest SHA256: <https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/rollups/sha256.js>;
- Repository github con l'intero progetto e la documentazione con LaTeX sorgente: ;
- Bootstrap: <https://getbootstrap.com>;
- Swiper: <https://swiperjs.com/swiper-api>;