

Progetto di Applicazioni Web e Cloud – Applicazione Web “Il Volo”

Alessandro Pettinà Matricola #892623 alessandro.pettina1@studenti.unimi.it

1. Analisi dei requisiti

Il progetto si pone l'obiettivo di sviluppare l'applicazione web “Il Volo” per la gestione di un aeroporto.

2. Identificazione delle funzionalità da sviluppare

“Il Volo” gestisce un numero di tratte aeree non conosciuto a priori ed è composto da due macro-scenari principali:

- **Gestione delle tratte**

Questo primo macro-scenario consiste nella gestione delle tratte che possono essere prenotate dalle compagnie aeree in finestre temporali di una settimana e contengono l'associazione tra giorno, ora e prenotazione. Per ogni tratta si vogliono gestire informazioni quali tipologia (nazionale o internazionale), aeroporto di partenza / arrivo, slot orari disponibili e numero di prenotazioni massimo per ogni slot, aeromobili abilitati alla tratta, ecc...

Ogni aeromobile contiene informazioni quali numero di posti, velocità massima, apertura alare, raggio, tipo di motore. In aggiunta a quanto detto si deve predisporre una sezione per la ricerca delle tratte in base a un mix delle informazioni sopra riportate (ad esempio tipologia, aeroporto di partenza / arrivo, aeromobile).

Questi dati devono poter essere visualizzabili anche da utenti non registrati.

- **Gestione delle prenotazioni delle tratte**

Fornisce un meccanismo di gestione delle prenotazioni delle tratte.

Le compagnie devono potersi registrare, modificare i propri dati e cancellarsi.

Per ogni compagnia si devono memorizzare informazioni specifiche (ad esempio nazione, valore azionario e partner), informazioni sulla flotta aerea (numero e tipo di aeromobili) e informazioni relative all'account generato.

Una volta registrate, le compagnie aeree possono effettuare login al sito e prenotare una o più tratte a patto di possedere aeromobili abilitati alla tratta e secondo la disponibilità dei relativi slot.

Utenti non registrati possono solo ricercare tratte e accedere a informazioni pubbliche.

3. Progettazione della struttura e della presentazione delle pagine web

Per la realizzazione di tale applicazione web è stato utilizzato il framework [Bootstrap](#) (v3.3.7). Si tratta di una raccolta di strumenti per la creazione di siti e applicazioni per il web. Essa contiene modelli di progettazione basati su *HTML* e *CSS*.

Di seguito troviamo allegata la struttura principale del progetto:

```
IlVolo/
+-----
|      aereo.html
|      account.html
|      book.html
|      change.html
|      index.html
|      login.html
|      register.html
|      tratte.html
|
+---css/
|      bootstrap.css
|      DidactGothic.css
|      account.css
|      book.css
|      index.css
|      login.css
|      register.css
|      tratte.css
|      change.css
|
+---font/
|      Roboto-Regular.ttf
|
+---fonts/
|      glyphs-halflings-regular.eot
|      glyphs-halflings-regular.svg
|      glyphs-halflings-regular.ttf
|      glyphs-halflings-regular.woff
|      glyphs-halflings-regular.woff2
|
+---js/
|      bootstrap.js
|      jQuery.js
|      list-group.js
|      passiveEvents.js
|      popper.js
|      tratte.js
|
\---json/
      tratte.json
```

Entriamo nel dettaglio delle pagine *HTML*:

- `index.html` è la pagina iniziale nella quale è presente un riassunto dei servizi svolti dalla nostra applicazione web;
- `register.html` contiene il form per registrare una compagnia alla nostra applicazione web;
- `login.html` contiene il form per accedere alla nostra applicazione web;
- `account.html` contiene un riassunto delle principali informazioni della compagnia e lo storico delle prenotazioni effettuate;
- `change.html` contiene il form per modificare i dati della compagnia che ha eseguito l'accesso;
- `tratte.html` contiene l'elenco delle tratte disponibili su "Il Volo" e permette ad una compagnia registrata di poter procedere alla prenotazione della tratta;
- `book.html` contiene il form per prenotare la tratta scelta;
- `aereo.html` contiene le informazioni relative agli aerei presenti nel local storage.

Per quanto riguarda le pagine *CSS*:

- `index.css`, `register.css`, `login.css`, `account.css`, `change.css`, `tratte.css` e `book.css` sono le pagine *CSS* corrispondenti ad ogni pagina *HTML*, nelle quali sono definite le parti grafiche di esse;
- `bootstrap.css` contiene la parte grafica relativa al framework Bootstrap;
- `DidactGothic.css` contiene i riferimenti al font importato.

Il file `tratte.json` contiene le informazioni delle tratte e degli aerei in formato JSON.

Dentro la cartella `js/` sono contenuti i file JavaScript in particolare relativi a Bootstrap e jQuery.

All'interno di essa sono presenti inoltre `tratte.js` e `list-group.js` da me implementati che consentono, il primo di trasformare il testo del pulsante "Visualizza dettagli" in "Nascondi dettagli" e viceversa nella pagina `tratte.html` e il secondo di espandere i dettagli relativi agli slot all'interno della pagina `book.html`.

4. Progettazione della sorgente di informazioni statica o dinamica

Le strutture dati presenti all'interno del progetto sono tre:

- **Compagnie aeree**

La prima struttura dati presente nell'applicazione web è quella necessaria per la gestione delle compagnie aeree registrate. Essa non viene creata a partire da un file *JSON*, ma viene creata e inizializzata direttamente al primo avvio del sito.

La struttura presente nel local Storage è così composta:

- `login`: che indica se una compagnia ha effettuato l'accesso o no;
- `companyName`: che contiene il nome della compagnia aerea;
- `psw`: che contiene la password inserita al momento della registrazione;
- `country`: che indica la nazione nella quale ha sede la compagnia aerea;
- `valAz`: che indica il valore azionario della compagnia aerea;
- `partner`: che indica il partner commerciale della compagnia aerea;
- il vettore `plane[]` che contiene gli aerei posseduti dalla compagnia aerea;
- il vettore `prenotazione[]` che contiene le prenotazioni effettuate dalla compagnia.

A loro volta, vediamo l'organizzazione dei due vettori presenti all'interno della struttura principale:

1. `plane[]` permette la memorizzazione degli aerei registrati all'atto della registrazione.

È composto da alcuni attributi quali:

- `id`: progressivo numerico e univoco per ogni aereo;
- `aereo`: nome di ogni singolo aereo registrato;
- `numero`: quantità relativa ad ogni singolo aereo registrato;
- `tipologia`: se un aereo è adatto a tratte nazionali o internazionali.

2. `prenotazione[]` permette la memorizzazione delle prenotazioni effettuate dalla compagnia.

È composto da alcuni attributi quali:

- `idPrenotazione`: progressivo numerico della prenotazione;
- `idTratta`: id relativo alla tratta prenotata;
- `modelloAereo`: id relativo all'aereo utilizzato nella prenotazione;
- `slot`: id relativo allo slot prenotato.

Di seguito allego un esempio di oggetto del vettore `register`, nel quale salvo i dati relativi alle compagnie aeree, presente sul localStorage:

```
[
  login: 1,
  companyName: "Alitalia",
  psw: "prova",
  country: "Italia",
  valAz: "0.2",
  partner: "Barilla",
  plane:
  [
    0: {id: 2, aereo: "Boeing 737-800", numero: 2, tipologia: "Nazionale"},
    1: {id: 3, aereo: "Boeing 777", numero: 1, tipologia: "Internazionale"}
  ],
  prenotazione:
  [
    0: {idPrenotazione: 1, idTratta: 1, modelloAereo: "4", slot: "1"}
  ]
]
```

- **Tratte**

La seconda struttura dati utilizzata nello sviluppo del progetto è quella relativa alle tratte. Le informazioni sono contenute in un file JSON, la cui struttura è molto semplice da comprendere.

Il vettore `tratte` memorizza al suo interno le informazioni di base di ogni tratta e due sotto-vettori, il primo dei quali contiene al suo interno l'id degli aerei compatibili tramite i quali posso risalire al vettore degli aerei; il secondo sotto-vettore, invece, permette di conoscere la disponibilità e le informazioni relative agli slot di ogni singola tratta.

Questi dati vengono salvati nel `localStorage` alla prima visualizzazione della pagina `index.html`, e verranno modificati dinamicamente in base alle prenotazioni effettuate dalle compagnie.

Di seguito allego un esempio di oggetto del vettore `tratte` presente sul `localStorage`:

```
[
  {
    "id": 1,
    "name": "Tratta 1",
    "type": "Internazionale",
    "from": "Bangkok",
    "to": "Milano",
    "accepted_plane": [2,4,5],
    "slots": [
      {
        "id": 1,
        "slot": "AK-101",
        "slotDisp": 3,
        "time": {
          "day": "Lunedì",
          "from": "15:00",
          "to": "16:00"
        }
      },
      {
        "id": 2,
        "slot": "AK-102",
        "slotDisp": 1,
        "time": {
          "day": "Giovedì",
          "from": "22:00",
          "to": "23:00"
        }
      }
    ]
  }, ...
]
```

- **Aerei**

La terza e ultima struttura dati utilizzata nel progetto è quella relativa agli aerei. Le informazioni sono contenute in un file *JSON*, le quali vengono salvate nel *localStorage* alla prima visualizzazione della pagina `index.html`.

Il vettore `aerei` memorizza al suo interno le informazioni di base relative agli aerei, come il nome, la lunghezza, velocità massima, passeggeri, ecc ...

Di seguito allego un esempio di oggetto del vettore `aerei` presente sul *localStorage*:

```
[
  {
    "id":2,
    "name":"Boeing 767-300",
    "tipologia":"Internazionale",
    "lunghezza":54.9,
    "apertura alare":47.6,
    "vel":851,
    "vel max":913,
    "superficie alare":283.3,
    "peso a vuoto":86070,
    "peso max al decollo":158760,
    "passeggeri":{"1classe":350,"2classe":269,"3classe":218},
    "capacità combustibile":90.625,
    "motore":"2 turbofan General Electric CF6-80C2"
  }, ...
]
```

5. Implementazione delle pagine Web

Ogni pagina web è composta da una navbar generata staticamente in *HTML* alla quale viene effettuata una operazione di *append*, nella quale tramite la funzione *JavaScript* `stampaNavbarRight()`, se è stato effettuato l'accesso, viene aggiunto dinamicamente il nome dell'utente loggato e il pulsante per effettuare il logout, altrimenti verranno visualizzati i pulsanti per accedere alla pagine di login e registrazione.



Il resto della pagina viene generato dinamicamente o staticamente in base alla necessità.

index.html

In questa pagina viene visualizzato un semplice riassunto delle operazioni effettuabili sul sito web e tramite la navbar è possibile raggiungere le altre pagine de "Il Volo".



[register.html](#)

Questa pagina consente di registrare l'account di una compagnia aerea.

È composta da un form suddiviso in due *panel*: il primo dedicato all'inserimento delle informazioni della compagnia (nome, password, nazione, valore azionario e partner) e il secondo invece dedicato alla sua flotta aerea.

All'apertura della pagina, sarà presente solamente il panel di sinistra. Dopo aver inserito i dati e cliccato sul pulsante "Avanti", verrà visualizzato il panel di destra.

Quando viene cliccato il bottone "Avanti", viene effettuato un controllo tramite *jQuery* atto a verificare che tutti i campi siano stati riempiti. In caso negativo, viene visualizzata un'animazione che trasforma il bordo inferiore dell'input di rosso e fa comparire la scritta "Mancante" per 2 secondi, per poi ripristinare la situazione iniziale.

The diagram illustrates the transition between two panels in the registration process. On the left, the 'Informazioni compagnia:' panel contains five input fields: 'Nome:' (placeholder: 'Inserisci nome compagnia'), 'Password:' (placeholder: 'Inserisci password' with a toggle icon), 'Nazione:' (placeholder: 'Inserisci nazione'), 'Valore azionario:' (placeholder: 'Inserisci valore azionario'), and 'Partner:' (placeholder: 'Inserisci partner'). Below this panel is a blue 'Avanti' button. A red arrow points to the right panel, which is also titled 'Informazioni compagnia:'. This panel shows the same fields but with data entered: 'Nome:' is 'EasyJet', 'Password:' is masked with '.....', 'Nazione:' is 'Inghilterra', 'Valore azionario:' is '0,35', and 'Partner:' is 'Partner mancante!'. The 'Partner' input field has a red border and the text 'Partner mancante!' is displayed below it. Below this panel is another blue 'Avanti' button.

Una volta inseriti i dati in modo corretto, comparirà il panel di destra atto all'inserimento della flotta aerea.

The diagram illustrates the transition between two panels in the registration process. On the left, the 'Informazioni compagnia:' panel contains five input fields: 'Nome:' (placeholder: 'EasyJet'), 'Password:' (placeholder: '.....' with a toggle icon), 'Nazione:' (placeholder: 'Inghilterra'), 'Valore azionario:' (placeholder: '0,35'), and 'Partner:' (placeholder: 'Delta Airlines'). Below this panel is a blue 'Avanti' button. A red arrow points to the right panel, which is titled 'La tua flotta:'. This panel contains a table with two columns: 'Tipologia aereo' and 'Numero'. The table lists five aircraft types with their corresponding counts: Boeing 767-300 (0), Boeing 737-800 (3), Boeing 777 (0), Airbus A380 (2), and Airbus A340 (2). Below this panel is a green 'Completa registrazione' button.

Tipologia aereo	Numero
Boeing 767-300	0
Boeing 737-800	3
Boeing 777	0
Airbus A380	2
Airbus A340	2

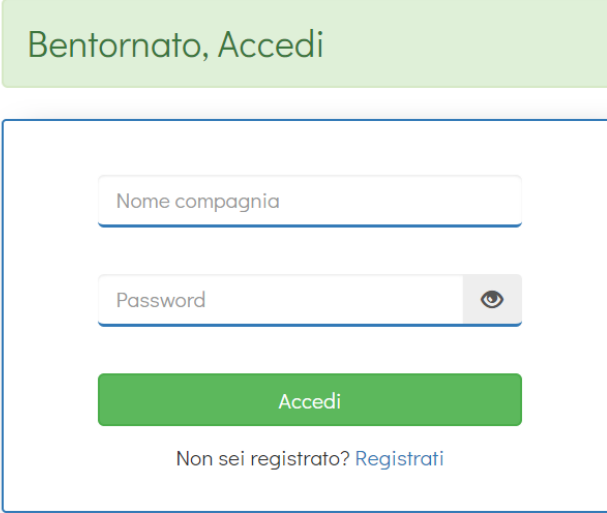
Cliccando su “Completa registrazione” viene richiamata la funzione `registrazioneUtente()` che permette di inserire la compagnia all’interno del vettore `register` del `localStorage`.

I dati inseriti vengono salvati all’interno del vettore `dati`, viene controllato che il nome della compagnia non sia già presente nel sistema tramite la funzione `checkNome()` ed infine salvati sul `localStorage`.

Successivamente vengono eliminate le righe degli aerei con quantità uguale a 0 per avere un `localStorage` più facilmente comprensibile. Nel caso in cui tutti gli aerei hanno quantità 0, viene visualizzato un messaggio di errore per consentire all’utente di registrare la compagnia in modo corretto.

[login.html](#)

Questa pagina consente alle compagnie registrate di accedere al sito per poter godere dei servizi del portale.



The image shows a login form titled "Bentornato, Accedi". It contains two input fields: "Nome compagnia" and "Password". The "Password" field has a toggle icon (an eye) to the right of it. Below the fields is a green button labeled "Accedi". At the bottom, there is a link that says "Non sei registrato? [Registrati](#)".

In questa pagina sono presenti 2 funzioni JavaScript fondamentali:

- `checkLogin()` : che reindirizza l’utente alla pagina `account.html` nel caso abbia già effettuato l’accesso;
- `loginUtente()` che dopo aver controllato che i campi nome e password siano stati riempiti, salva i dati all’interno del vettore `dati` e controlla se essi hanno una corrispondenza all’interno del vettore `register` del `localStorage`; in caso positivo l’utente viene reindirizzato alla pagina `account.html`, altrimenti verrà visualizzato un messaggio di errore e consentirà all’utente di effettuare un nuovo tentativo.

Cliccando sulla glyphicon dell’occhio, l’input `type="password"` viene trasformato tramite `jQuery` in `type="text"` per consentire all’utente di controllare la password inserita.

[tratte.html](#)

In questa pagina viene stampato l'elenco delle tratte disponibili in modo dinamico attraverso la funzione *JavaScript* `inizializzaTratte()` che prende i valori dal *localStorage* e li stampa all'interno di specifici *div* predisposti da Bootstrap chiamati *panel*.

Elenco tratte:

Tratta 1

Bangkok → Milano

Internazionale

Boeing 767-300, Boeing 777, Airbus A380

Prenota tratta

Visualizza dettagli

Tratta 2

Milano → Roma

Nazionale

Boeing 737-800, Airbus A340

Prenota tratta

Visualizza dettagli

Ricerca tratta:

Filtro per città:

Filtro per tipologia aereo:

Filtro per tipologia tratta:

☐ Internazionale ☐ Nazionale

Cerca

Annulla

I dati relativi alle tratte vengono inizialmente importati dal *localStorage* in un oggetto JavaScript tramite il comando: `var tratteNR = JSON.parse(localStorage.getItem("tratte"));`

L'oggetto viene iterato tramite un *for* e i vari attributi di esso vengono disposti all'interno del codice *HTML*. All'interno di questo *div* è presente il bottone per prenotare la tratta scelta. Cliccando su tale bottone, verrà richiamata la funzione `effettuaPrenotazione(i)` che rimanderà l'utente alla pagina di prenotazione di tale tratta.

Un'altra funzione presente in questa pagina è quella che permette di effettuare la ricerca.

Una volta compilato il form sulla destra con uno o più campi, cliccando sul bottone "Cerca" vengono inviati i dati presenti alla funzione `cercaTratta()`. Tramite questa funzione controllo se i valori inseriti all'interno del form corrispondono a quelli presenti sul *localStorage*.

La prima cosa da effettuare è quella di controllare se i campi sono stati riempiti dall'utente oppure no. In caso negativo imposto le variabili di appoggio a *true*.

Esempio:

```
if(document.cercatratte.Partenza.value == "")
    City = true;
else
    City = document.cercatratte.Partenza.value;
```

Una volta settate le variabili di appoggio, tramite la funzione `RegExp()` con parametri "ig" viene effettuata la ricerca ignorando la distinzione tra maiuscole e minuscole (attributo *i*), e eseguendo una ricerca globale (attributo *g*). Infine, tramite il comando `test()` viene controllato che il contenuto di `RegExp()` abbia una corrispondenza nel *localStorage*.

```
c = (RegExp(City, "ig").test(tratteNR[i].from) | RegExp(City, "ig").test(tratteNR[i].to));
```

Una volta effettuati questi controlli, se tutti i valori sono stati impostati a *true*, verrà attivato il procedimento di stampa delle tratte che rispettano la ricerca.

[book.html](#)

Cliccando sul bottone prenota all'interno della pagina `tratte.html`, viene effettuato un reindirizzamento a `book.html` salvando sul `sessionStorage` l'id della tratta da prenotare.

Una volta visualizzata tale pagina, tramite la funzione `getID()`, viene estratto l'ID della tratta prescelta e viene richiamata a sua volta la funzione `stampaTratta()` che permetterà di stampare il *panel* relativo ad essa per procedere alla prenotazione definitiva.

Effettua prenotazione:

Tratta 1

Bangkok → Milano

✈ Internazionale

🌐 Boeing 767-300, Boeing 777, Airbus A380

Seleziona aereo:

Seleziona slot:

Seleziona aereo ▼

Seleziona slot ▼

Prenota

Dettagli slots

Slot AK-101 ▼

Giorno: Lunedì

Orario partenza: 15:00

Orario arrivo: 16:00

Slot AK-102 ▼

Disponibilità:

Modello aereo	Disponibilità	Tipologia volo
Boeing 737-800	2	Nazionale
Boeing 777	1	Internazionale

Nome slot	Disponibilità	Giorno Partenza
AK-101	2	Lunedì
AK-102	0	Giovedì

Al centro, tramite uno script presente nel body verranno visualizzati gli slot di tale tratta per poterne visualizzare le informazioni complete.

Infine, sulla destra vi è un riepilogo delle disponibilità di aerei e slot compatibili con la tratta prescelta.

Questi dati vengono stampati rispettivamente tramite le funzioni `stampaAerei()` e `stampaSlots()`, le quali prendono i dati direttamente dal `localStorage`.

L'ultima funzione presente su questa pagina è la più importante: `stampaTratta()`.

Questa funzione permette all'utente, dopo aver selezionato aereo e slot e aver controllato la loro disponibilità, di procedere alla prenotazione della tratta, decrementando i valori di disponibilità dell'aereo e dello slot scelto all'interno del `localStorage`.

[account.html](#)

Dopo aver prenotato una tratta, l'utente verrà reindirizzato alla seguente pagina.

Le tue prenotazioni:

Prenota tratta

Prenotazione # 2

Tratta 1

Bangkok → Milano

✈ Internazionale

🌐 Boeing 777

Slot AK-101

Giorno: Lunedì
Orario partenza: 15:00
Orario arrivo: 16:00

Elimina

Alitalia

Nazione:
Italia

Valore azionario:
€ 0.2 per azione

Partner:
Barilla

Modello aereo	Disponibilità	Tipologia volo
Boeing 737-800	2	Nazionale
Boeing 777	1	Internazionale

Sul lato sinistro di essa all'interno del relativo *panel*, vengono visualizzate, tramite la funzione `stampaPrenotazioni()`, le informazioni delle tratte prenotate presenti all'interno del sottovettore *prenotazioni* della compagnia che ha effettuato l'accesso sul *localStorage*.

Vi è inoltre la possibilità di cancellare ogni prenotazione effettuata tramite `cancellaPrenotazione(j)`. Questa funzione rimuoverà la prenotazione dal vettore *register* e incrementerà la disponibilità dell'aereo e dello slot utilizzati nella prenotazione cancellata.

Sul lato destro, invece, vi è un riepilogo dei dati della compagnia aerea (`stampaDatiUtente()`). Cliccando sul *dropdown menu button* sarà possibile modificare i propri dati, oppure cancellare l'account.

La funzione `cancellaAccount()` tramite il comando `regUsers.splice(i, 1)` permetterà all'utente di eliminare il proprio account. I parametri passati nella funzione *splice* sono rispettivamente l'indice dell'utente che sta richiedendo l'operazione presente sul *localStorage*, e la quantità di oggetti di cancellare.

[change.html](#)

Questa pagina permette all'utente di modificare i dati del proprio account.

The image shows a web interface for modifying account data. It is divided into two main sections: 'Dati attuali:' (Current data) and 'Modifica dati:' (Modify data).

Dati attuali:

- Nome: **Alitalia**
- Password: *********
- Nazione: **Italia**
- Valore azionario: **€ 0.2 per azione**
- Partner: **Barilla**

Modifica dati:

- Nome:
- Password:
- Nazione:
- Valore azionario:
- Partner:

At the bottom, there are two buttons: 'Annulla modifica' (Cancel modification) in red and 'Conferma modifica' (Confirm modification) in green.

È composta principalmente da due *panel*, nei quali, sulla sinistra vengono visualizzati i dati attuali, e sulla destra vi è la possibilità di modificarli.

Una volta compilato il form a sinistra, cliccando su “Conferma modifica”, verrà richiamata la funzione `modificaDati()` che, dopo aver controllato che il valore degli input sia diverso da NULL, li sovrascriverà sul relativo vettore del *localStorage*.