

VEHICLE SHARING

JAITA52 Project Work

Progettare un'applicazione web utilizzando le tecnologie trattate a lezione

(Spring, Javascript, Bootstrap, MySql) che permetta la collaborazione di diversi utenti nel creare, aggiornare e gestire una **piattaforma di VEHICLE SHARING**.

Preparare una presentazione completa di slide per esibire e spiegare il lavoro portato a termine dal gruppo.

Front-end

Sarà necessario inventare il cliente immaginario che richiede la piattaforma, creando una brand identity (nome, logo, palette di colori, font, ecc...) che detti poi lo stile del sito internet.

Per ogni pagina dell'applicazione sarà necessario creare, usando Bootstrap, un layout responsivo che contenga almeno gli elementi richiesti dalla consegna e che sia progettato con buon senso dal punto di vista della User Experience.

Back-end

I dati relativi agli utenti sono conservati in una base di dati di nome "*projectwork*", contenente una tabella di nome "*utente*". Ciascun utente sarà rappresentato attraverso un record della suddetta tabella.

Il file *projectwork.sql* allegato in appendice contiene le istruzioni SQL per la definizione della base di dati descritta.

Utenti

Esistono 3 tipologie di utenti, identificabili attraverso il campo "*ruolo*" della tabella utente:

1. **Amministratore** (es. Ruolo="ROLE_ADMIN").
Questi utenti possono caricare nuovi veicoli (auto, scooter, monopattino, bicicletta..) specificandone le caratteristiche (elettrico, benzina, diesel..) e le disponibilità.
2. **Utente Registrato** (es. Ruolo="ROLE_USER").
registrarsi al portale per prenotare un veicolo disponibile in città scegliendo da una lista divisa in categorie.
Può acquistare un noleggio prolungato ("abbonamento giornaliero") .
Può modificare una prenotazione (cambio data, cambio veicolo).
3. **Utente pubblico**, non registrato (non presente nella tabella utente). Questo utente può solo vedere i veicoli disponibili: lista totale di tutti i veicoli e funzione ricerca per categoria veicolo o modello.

Non è richiesto implementare procedure di inserimento di nuovi utenti nell'archivio, ma soltanto le necessarie procedure di autenticazione attraverso un form di login

Veicoli disponibili

Un veicolo è costituito da una serie di informazioni:

1. VeicoloID
2. Categoria (auto, monopattino, bicicletta, etc.)
3. Descrizione (modello, colore, cilindrata)
4. Alimentazione (elettrica, diesel, benzina, ibrida, etc.)
5. Indirizzo (Posizione Attuale)
6. Coordinate GPS
7. Disponibilità noleggio
8. Immagine veicolo

Ad esempio

1. 1
2. Auto
3. Fiat Panda 1200cc, colore blu, doppio Turbo compressore
4. Diesel
5. Via Carlo Alberto 22, Torino
6. 45.22325425,7.2342555
7. false
8. FiatPanda.jpg

Pagina Home

Tale pagina conterrà:

1. un carousel scorrevole (inclusione del file: carousel.html)
2. l'elenco dei veicoli disponibili (inclusione del file: disponibilita.html)
3. un link al pannello di lavoro (link alla pagina: pannello.html)
4. un link per il logout o per il login a seconda che l'utente abbia già superato o meno le procedure di autenticazione
5. un grafico sull'utilizzo dei singoli veicoli premiando i dati dei veicoli elettrici rispetto ai veicoli a carburante fossile. (info in Appendice B)

Carousel scorrevole (carousel.html)

Si tratta di un campo di testo e immagine scorrevole, aggiornato dinamicamente, in funzione del contenuto dell'archivio dei veicoli disponibili. Tale campo di testo dovrà contenere l'elenco di tutti i veicoli disponibili. Usare Slick o Swiper JS.

In tale elenco, ciascun veicolo dovrà essere rappresentato dalla sequenza:

***Descrizione – Indirizzo – link alla pagina veicolo.html con passaggio dell'ID veicolo come parametro di una richiesta http di tipo GET *** .

Disponibilità veicoli (disponibilita.html)

Dovrà essere costituito da un elenco contenente veicoli aggiornati dinamicamente (facoltativo: in funzione della data e dell'ora). Suddiviso in 2 sezioni, la prima conterrà i veicoli disponibili, la seconda conterrà l'elenco dei veicoli attualmente in fase di noleggio.

Ciascun veicolo dovrà essere rappresentato attraverso gli elementi:

Categoria – Alimentazione – Indirizzo – link alla pagina veicolo.html con passaggio dell'ID dell'evento come parametro di una richiesta http di tipo GET.

Pagina del singolo veicolo (veicolo.html)

Tale pagina viene prodotta dinamicamente in funzione dell'ID del veicolo ricevuto come parametro di una richiesta di GET. Qualora la pagina venisse richiesta senza nessun parametro, l'utente deve essere rediretto verso la pagina home.

Tale pagina dovrà contenere tutte le informazioni disponibili sul veicolo, opportunamente impaginate, e una mappa con il marker sulla posizione del veicolo, se disponibile alla prenotazione.

Pannello di lavoro (pannello.html)

Protetta da un controllo per l'autenticazione, accesso limitato ad utenti con ruolo di amministratore, deve mostrare i seguenti contenuti:

1. Se l'utente non ha già effettuato le operazioni di login, deve essere visualizzato un form di login in cui inserire UserID e password, per poi essere reindirizzati, una volta autenticati, alla stessa pagina protetta dalla quale è scaturita la richiesta di login.
2. Se l'utente è di tipo base "ROLE_USER" deve essere reindirizzato a una pagina di errore (errore.html) contenente un messaggio che specifichi che l'utente non dispone di diritti sufficienti alla modifica del parco veicoli e un link per il logout.
3. Se l'utente è di tipo admin ("ROLE_ADMIN") deve essere visualizzato un pannello di amministrazione contenente le 3 seguenti funzionalità:
 - a. Inserimento di un nuovo veicolo – link a inserisci.html
 - b. Elenco dei veicoli amministrabili – link a view.html
 - c. Logout

Pagina di inserimento di un veicolo (inserisci.html)

Protetta da un controllo per l'autenticazione. Deve prevedere un form per l'inserimento di un veicolo nell'archivio e un bottone di logout attraverso il quale tornare alla pagina home.

Tale form dovrà contenere i seguenti campi:

Categoria Veicolo, Descrizione, Alimentazione, Indirizzo, Coordinate GPS, Disponibilità al noleggio, Upload di un immagine, URL di una pagina web esterna all'applicazione per cercare indirizzo e coordinate gps (es. OpenStreetMap). L'inserimento delle informazioni relative agli ultimi tre campi del form *non* deve essere obbligatorio.

(facoltativo - implementare inserimento dinamico di indirizzo e coordinate tramite javascript)

Attraverso questa pagina l'amministratore deve poter inserire le informazioni relative ad un nuovo veicolo, che pertanto dovranno essere visibili nella home e attraverso la pagina del singolo veicolo.

Elenco dei veicoli e prenotazioni amministrabili (view.html)

Protetta da un controllo per l'autenticazione. Deve prevedere un bottone di logout attraverso il quale tornare alla pagina home. Deve permettere la visualizzazione di una lista con una riga per ciascun veicolo amministrabile e un corrispondente checkbox che, se spuntato, rende visibile il veicolo all'utente.

Se sono presenti più di 10 veicoli, la scansione dell'elenco deve essere ripartita su più pagine generate dinamicamente (eventi da 1 a 10, da 11 a 20 ecc.).

A fianco di ogni veicolo dell'elenco devono essere presenti due pulsanti con testo "modifica" e "cancella", attraverso i quali sarà possibile attivare le procedure di modifica e cancellazione del veicolo.

1. La procedura di modifica, dovrà prevedere la visualizzazione di un form, precompilato con i dati relativi al veicolo selezionato, ma editabile in tutte le sue componenti tranne VeicoloID, un pulsante di reset che ripristinerà i dati, e uno di submit per i nuovi dati.
2. La procedura di cancellazione prevederà l'eliminazione del record relativo al veicolo dall'archivio di quelli disponibili, e conseguentemente da tutte le componenti dinamiche che ne permettevano la visualizzazione.
3. L'aggiunta e modifica di un veicolo devono prevedere la possibilità di inserire un'immagine da caricare in una cartella del server, e il salvataggio del nome dell'immagine nel relativo campo del database, che verrà successivamente utilizzata nell'src del tag dell'immagine

Logout

Deve prevedere la cancellazione di tutte le informazioni che permettono il riconoscimento dell'utente dell'applicativo e il ritorno alla pagina home

Pagina utente (utente.html)

Protetta da un controllo per l'autenticazione.

Deve contenere l'elenco prenotazioni di un utente con la possibilità modifica e/o cancellare la prenotazione, ed un eventuale elenco prenotazioni chiuse/terminate.

Appendice

Si dispone di una base di dati creata attraverso questo file di definizione:
(modificare a piacere)

--- *projectwork.sql* ---

```
CREATE DATABASE IF NOT EXISTS `projectwork`;  
USE `projectwork`;
```

```
#CREATE USER 'app_generation'@'localhost' IDENTIFIED BY 'generation_2022';
GRANT ALL ON projectwork.* TO 'app_generation'@'localhost';
FLUSH PRIVILEGES;
```

```
CREATE TABLE IF NOT EXISTS `utente` (
  `utente_id` int NOT NULL AUTO_INCREMENT,
  `nome` varchar(75) DEFAULT NULL,
  `cognome` varchar(75) DEFAULT NULL,
  `data_nascita` date DEFAULT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(20) NOT NULL,
  `ruolo` enum('RUOLO_ADMIN','RUOLO_UTENTE') NOT NULL,
  PRIMARY KEY (`utente_id`),
  KEY `k_email` (`email`)
);
```

```
INSERT INTO `utente` (nome,cognome,dataNascita,email,password,ruolo)
VALUES ('Paolo','Rossi','1994-06-07','admin@email.com','admin','RUOLO_ADMIN'),
('Carlo','Verdi','2001-03-19','utente@email.com','utente','RUOLO_UTENTE');
```

Appendice B

Costruire il grafico utilizzando [Chart.js](#) (o altra libreria a scelta)

Ogni veicolo inserito ha la proprietà “alimentazione”, ci saranno quindi veicoli elettrici, benzina, diesel.

Questi dati verranno utilizzati per produrre i dati nel grafico in home page.

Ogni veicolo elettrico ha un fattore di risparmio in termini di CO2 ogni anno secondo la seguente tabella:

Bicicletta: 100

Bicicletta Elettrica: 83

Monopattino: 95

Auto Elettrica: 51

Auto Ibrida: 33

Auto Benzina: 0

Auto Diesel: 0

etc.

Questi fattori verranno moltiplicati per il numero di veicoli che l'azienda possiede, andando a determinare il risparmio di CO2 prodotto dall'utilizzo dei veicoli.

All'inserimento dei nuovi veicoli si modificherà il grafico presente nella home page.

Esempio:

30 biciclette → $30 \times 100 = 3000$ kg risparmiati

25 Monopattino → $25 \times 95 = 2375$ kg di CO2 risparmiati

10 Auto elettriche → $10 \times 51 = 510$ kg

etc.

Questo grafico rappresenta quanti kg di CO2 risparmia ogni categoria di veicolo “green” in base al numero di veicoli presenti.

Calcolo alternativo: media annuale CO2 prodotta per veicolo.
automobile benzina: 4600 kg CO2 x 10 veicoli = 46000 kg CO2/anno
automobile ibrida: 2100 kg
automobile elettrica: 800 kg
monopattino elettrico: 65 kg
bicicletta elettrica: 45 kg
bicicletta: 0 kg
etc.

Appendice C

- Trello, Figma, Canva, Unsplash
 - Organizzazione e assegnazione task <https://trello.com/>
 - <https://www.figma.com/>
 - <https://www.canva.com/>
 - <https://unsplash.com/>
- Github per condivisione codice
- Carousel
 - <https://kenwheeler.github.io/slick/>
 - <https://swiperjs.com/>
- Chart.js
 - grafici
 - <https://www.chartjs.org/docs/latest/getting-started/>
- Leaflet
 - mappa con markers (tipo google maps)
 - <https://leafletjs.com/SlavaUkraini/examples.html>
 - registrarsi su mapbox e ottenere token
- Mapbox Geocoding - facoltativo
 - ricerca indirizzi e coordinate gps
 - <https://docs.mapbox.com/api/search/geocoding/>
- Tutti i nomi delle pagine html e attributi database possono essere modificati a piacere, l'importante è mantenere le funzionalità richieste
- Realizzare il progetto usando un design pattern MVC o REST o un misto delle 2.