

Diario di lavoro

Luogo	Canobbio
Data	21.11.2019

Lavori svolti

Durante la giornata di oggi ho finito il filtraggio tramite un range di date e ho modificato i percorsi del progetto per fare in modo di averli uguali a quelli di Infomaniak. Ho inoltre aggiornato il Gantt.

Il risultato finale è il seguente:

The screenshot shows a web browser window with the URL `127.0.0.1:8080/gestione_parcheggi/2019/ricerca/index`. The page has a dark theme and a navigation bar with 'Gestione Parcheggi', 'Home', 'Offerta', and 'Ricerca' (active). A search form is displayed with two date pickers: 'Inserisci le date da cercare:' (Dal: , al:) and 'Inserisci la disponibilità:' (Tutto il giorno). A 'Cerca' button is next to the first date picker. Below the form is a table with the following data:

Disponibilità	Data disponibilità	Numero di targa	Prenota
Mattina	02-12-2019	TI-234567	<button>Prenota</button>
Mattina	16-11-2019	TI-234234	<button>Prenota</button>
Tutto il giorno	20-11-2019	GR-12345	<button>Prenota</button>
Tutto il giorno	23-11-2019		<button>Prenota</button>
Proprietario	Disponibilità	Data disponibilità	Prenota

Per riuscire a fare il tutto ho aggiunto i nuovi datepicker e un pulsante per eseguire il filtraggio alla pagina dei risultati.

```
<div class="col-6">
  <p class="text-white font-weight-bold">Inserisci le date da cercare:</p>
  <div class="input-group justify-content-center">
    <p class="text-white font-weight-bold ml-3 mr-3">Dal </p>
    <input type="text" id="startDate" name="startDate" class="datepicker">
    <p class="text-white font-weight-bold ml-3 mr-3">al </p>
    <input type="text" id="endDate" name="endDate" class="datepicker">
    <input type="submit" value="Cerca" class="button btn-primary ml-3" onclick="filterDate()">
  </div>
</div>
```

In seguito ho creato il metodo di filtraggio nel rispettivo Model.

```
/**
 * Funzione che filtra i posteggi disponibili in base alle date scelte dall'utente.
 */
public static function filterByDate($startDate, $endDate)
{
    $startDateFormat = date_format(date_create($startDate), format: "Y-m-d H:i:s");
    $endDateFormat = date_format(date_create($endDate), format: "Y-m-d H:i:s");

    if($startDateFormat > $endDateFormat)
    {
        $_SESSION['minDateError'] = 'Errore: la data iniziale è più grande di quella finale';
    }
    else
    {
        unset($_SESSION['minDateError']);
    }

    self::$statement = Database::get()->prepare( statement: "select * from posteggio where data_disp between :start_date and :end_date");
    self::$statement->bindParam( parameter: ':start_date', &variable: $startDateFormat, data_type: \PDO::PARAM_STR);
    self::$statement->bindParam( parameter: ':end_date', &variable: $endDateFormat, data_type: \PDO::PARAM_STR);
    self::$statement->execute();

    self::$parcheggi = self::$statement->fetchAll( fetch_style: \PDO::FETCH_ASSOC);

    self::formatCarAndDate();
}
```

In questo metodo faccio la stessa select di base che carica tutti i posteggi disponibili, ma gli aggiungo anche il valore data_disp da controllare.

Ho modificato il codice presente nel controller per fare in modo di visualizzare il cambiamento dei dati una volta filtrati.

```
/**
 * Funzione che richiama il metodo per ricevere i posteggi offerti e poi reindirizza verso la pagina che li mostra.
 */
public function index()
{
    $selected = null;
    if(isset($_GET['filter_disp']))
    {
        $disp = Validator::testInput($_GET['filter_disp']);
        RicercaModel::filterByDisp($disp);
        $selected = $disp;
    }
    elseif (isset($_GET['startDate']) && isset($_GET['endDate']))
    {
        $startDate = $_GET['startDate'];
        $endDate = $_GET['endDate'];

        RicercaModel::filterByDate($startDate, $endDate);
    }
    else {
        RicercaModel::getParcheggiDisponibili();
    }
    ViewLoader::load( template: 'ricerca/index', array('parcheggi'=>RicercaModel::$parcheggi, 'selected' => $selected));
}
```

Tutto ciò per funzionare ha bisogno della seguente funzione javascript che prende i valori selezionato nei datepicker e li invia tramite GET al controller.

```
function filterDate() {  
    var startDate = document.getElementById('startDate').value;  
    var endDate = document.getElementById('endDate').value;  
    window.location.href = "/gestione parcheggi2019/ricerca/index/?startDate=" + startDate + "&endDate=" + endDate;  
}
```

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

In linea.

Programma di massima per la prossima giornata di lavoro

Caricare il sito online e fare il metodo per la prenotazione.