

SKIItalia

„The perfect place for all lovers of winter sports and Italy”

Daniel Talaj

Tecnologie Web, University of Turin, 2019

Contents

1	General Information.....	3
1.1	The Site Theme	3
2	Functionality	3
2.1	Login.....	3
2.2	Logout.....	4
2.3	Registration.....	4
2.4	Management of user generated content	4
3	Features	4
3.1	Database interrogation	4
4	Front-end.....	5
5	Back-end, communication between front-end and back-end	6

1 General Information

1.1 The Site Theme

SKItalia is a site for all lovers of winter sports who would like to spend their winter holidays in Italy, but need some information to help them decide about which of the available destinations to choose. It is especially useful for skiers and snowboarders. It offers detailed information about a large variety of Italian ski resorts including all of the biggest and most popular resorts, represented by Val Gardena or Cervinia, through smaller, local resorts that are ideal for families, like Alpe Lusia, but also includes hidden gems like the glacier ski resort of Val Senales.

Information include details about size of the ski resorts (total length of slopes, number of lifts), location (region), snow security (maximum elevation), but also information about pricing (price of daily ticket).

Users have an option to personalize the content of the page by adding their favourite ski resorts to their list of favourites. Users can access their personal list of favourite ski resorts and add new ones at any time.

2 Functionality

2.1 Login

SKItalia is a site where the user has to log in to his personal account to gain access to the features of the site. Without being logged in, user cannot do anything except attempting to log in or creating a new account.

Login is handled by a login form on the landing page. If the user submits incorrect credentials, he receives a message and stays on the landing page. If the login is successful, user is redirected to the main/home page of the site (**mainpage.php**).

Manually addressing **mainpage.php** without being logged in redirects the user to the landing page **index.php** with a message to log in first. This is possible thanks to sessions.

2.2 Logout

User has an option to Logout at any time while using the site. There is a logout button available on **mainpage.php** that triggers the logout action when clicked.

After logging out, user of the page loses his initially created session and is redirected to the landing page **index.php**. If the user wants to use features of the page, first he needs to log in again.

2.3 Registration

If the user does not have an account yet, he can create a new account on the **registration.php** page. User is asked to enter his credentials (user name, password) and also to enter his e-mail address and gender. Later, user will be using the selected user name and password as credentials to access the site.

2.4 Management of user generated content

Content management is possible thanks to sessions. When the user is logged in, he can access personalized content. No user can access personalized content of another user. Accessing this is strictly connected to username that is included in the session data.

3 Features

Sessions are an important feature of the site. Sessions help to assure that no one can view anything that he/she should not see or should not have access to. Having access to personal content is strictly bond to a personal account and user has his own session while using the page. Session data (user name) is also crucial for

3.1 Database interrogation

Database interrogation is handled by a large library of functions stored in file **database.php** that handle:

- Connection to a MySQL database
- User credentials verification on login

- Creating a new user account (including a check if the selected user name already exists)
- Password hash function to hide original passwords from database users -> thanks to hash function, none of the passwords are directly visible in the database

```
function password_hash_custom($password){
    $hash = password_hash($password, PASSWORD_DEFAULT);
    return $hash;
}
```

- Adding one or more ski resorts to a personal list of favourite ski resorts
- Requesting ski resort data or the personal list
- Requesting ski resort data

SQL Injection is prevented by frequent use of PDO::quote() function, example:

```
$name = $db->quote($name);
```

The content of the site is mainly presented in the form of a table on the main page that provides generally available ski resort data or list of favourite ski resorts.

4 Front-end

Front-end behaviour is handled by JavaScript with high usage of JQuery functions and features.

The site follows the unobtrusive style of programming, when content of the page is separated from the behaviour. There are no direct calls of JavaScript functions or code from the HTML elements of the page.

All actions on the page are handled by JQuery event handling features. These event handlers are that are included in the \$(document).ready() function of each JavaScript file.

Example of an event handler:

```
$("#AddToFavoritesButton").on("click", function (){});
```

Source files of the project are organized in folders – JQuery library is located in folder named **libraries**, JavaScript files are located in folder named **scripts**, CSS files are in folder **styles**, images in **images** folder.

5 Back-end, communication between front-end and back-end

Data transfers between front-end and back-end are handled by AJAX calls in JavaScript with usage of JQuery. JSON is the only data format that has been used in these transfers. Data in form of JSON is handled by PHP on the back-end and JavaScript on the front-end. When sending data from front-end to back-end or vice versa, first the data is serialized to JSON format, then sent to the other side using an AJAX call and when received by the other side, it is further manipulated with.

The application has been developed in Microsoft Visual Studio 2012, run on WAMP Server (version 3.1.4). Database has been operated with the use of phpMyAdmin.

During the development process, the application has been tested mainly in Firefox, but also briefly tested in Internet Explorer and Chrome, worked on both.