Tecnologie Web - Project report

# SKItalia

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

Daniel Talaj

Tecnologie Web, University of Turin, 2019

# Contents

# 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 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 such as 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). Thanks to these, users can compare different ski resorts and find the ones that best suit their preferences as well as their budgets.

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 resorts to the list 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 (**index.php)**. If the user submits incorrect credentials, he receives a message and stays on the landing page where he can try to log in again. If the login is successful, user is redirected to the main 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 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. Upon successful registration, user is redirected to the landing page where he can log in.

### 2.4 Management of user generated content

All users can browse all of the generally available data about ski resorts the same way.

Users can personalize the generally available content by adding one or more items to a personal list of favourite ski resorts. They can do it by selecting the ski resorts (by checking checkboxes next to each ski resort they want to add) and then clicking the "**Add selected ski resorts to favorites**" button. Users can later request their list by clicking the "**My favorite ski resorts**" button which loads their list to the page.

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 the username of currently logged user that is included in the session data.

## 3 Features

### 3.1 Usability, interaction

Site follows the principles of usability, user is informed about success/failure of his actions. For example, when user attempts to log in he receives a message if his login was successful or not. User also receives similar feedback when he attempts to add ski resorts to his list of favorites. If there are no resorts selected and he clicks the button to add them to favorites, he will receive a message that no resorts are selected and if he should select some resorts first if he wants to add them.

### 3.2 Sessions

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 personalized 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 handling the personalized content.

User generated output and requests for personalized content are bond to the user name that is stored in current session and accessed by:

```
$UserName = $_SESSION["name"];
```

If the user completely closes the browser window, he loses the session and has to log in again to use the main page.

## 3.3 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)
- Requesting ski resort data
- Adding one or more ski resorts to a personal list of favourite ski resorts
- Requesting personal list of favourite ski resorts

## 3.4 Security

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

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

Password hash function is used to hide original passwords from database users, thanks to hash function, none of the passwords are directly visible in the database. Used function:

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

The password hash function is used both when creating a new account and when logging in (it compares hash of entered password with hash in DB) – user's real password practically is not visible.

### 3.5 Presentation

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 with all details or list of favourite ski resorts, again with all details. This table is dynamically built based on the size of the data to be displayed. The style of the table provides a very good readability of the data. There are checkboxes at the end of each row, user can selected the ski resorts, later he can add the selected ones to his personal list.

There are several buttons on the page that provide basic functionality of the page:

- "**Biggest skiresorts**" – loads list of skiresorts ordered from the biggest to smallest based on total length of slopes
- "**My favorite skiresorts**" **–** loads the personal list of favorite ski resorts of the currently logged user
- **"Add selected skiresorts to favorites"** – adds selected ski resorts to list of favorites

## 4 Front-end

### 4.1 Separation of content and behaviour

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. There are no styles defined directly in the HTML. Instead, links to both script files and style sheets are included in the `<head>` section of the pages.

Front-end behaviour is handled by JavaScript with high usage of JQuery functions and features. Actions on the page are handled by JQuery event handling features. These event handlers are included in the $(document).ready() function of each JavaScript file.

Examples of event handlers:

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

```
$('#chkParent').click(function () {});
```

### 4.2 Organization of files and folders

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 folder **images**.

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

## 5.1 Communication

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 parsed/encoded 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. JSON example (response from server request for list of favorite skiresorts):


```
[["Alta Badia","Trentino-South Tyrol (Trentino-Alto
Adige\/S\u00fcdtirol)","1226","1324","2550","130","53","56.00","7"],["Cortina
d'Ampezzo","Veneto","1700","1224","2924","120","35","50.00","8"]]
```


On the server side (PHP functions/scripts), errors are handled by try/catch statements. If an operation was not possible or if an error occurred, a proper response is returned.

Every AJAX call that is triggered by an event on the front-end, requests execution of a PHP script which later returns a proper response. For example, clicking the button to add ski resorts to favorites executes an AJAX call that sends data to **AddToFavoritesScript.php** that attempts to insert the data and returns a success/failure message.

## 5.2 Scheme of database

The site uses a custom built database talaj_web_tech_project. This database includes 3 tables that are used in this project:

- SKIRESORTS (data about ski resorts – data only requested, not manipulated by user)
- USERS (user account data – requested on login and new records created upon registration)
- USERS_SKIRESORTS (storing data for personalized content – combines data from SKIRESORTS and USERS – new records created when user adds ski resorts to his personal list; on request for personal list, it is provided thanks to a JOIN on this table)

The application has been developed in Microsoft Visual Studio 2012, run on WAMP Server (version 3.1.4). The used MySQL database has been operated by 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.