


Web Applications, A.Y. 2019/2020  
Master Degree in Computer Engineering  
Master Degree in ICT for Internet and Multimedia

Homework 1 – Client-side Design and Development

Submission date: 24 April 2020

Group	Project	
The Atomics	SkiAdvisor	
Last Name	First Name	Student Number
Basana	Francesco	1237329
Camuffo	Elena	1234370
Cibin	Nicola	1237330
Crosara	Laura	1234372
Moro	Matteo	1234368

## Objectives

Thanks to the territorial shape of the Italian peninsula, Italian people have the possibility to spend their holidays in many mountain villages, where more than a thousand ski facilities have been leased to make their stay during the winter period pleasant and enjoyable.

In this context, a team of developers is entrusted to realize a web application (SkiAdvisor) to help the skier in selecting and personalizing their own ski experience. SkiAdvisor requires a Database Management system (DBMS) to hold, handle and manage the data relative to ski localities and provide an instrument to efficiently satisfy the needs of each kind of user.

That applications' objectives are the following:

- Providing a complete and well-organized collection of data about ski areas, ski tracks, ski lifts, ski facilities and all the offered related services on the whole Italian territory.
- Allowing each kind of user to satisfy their own needs with an adaptable and flexible service combined with a friendly and intuitive visualization of the data.
- Giving the possibility to the owners of hotels and services to manage their activities.
- Yielding a feedback on all the elements involved in the ski system with a system of surveys and evaluations of users' own experiences.

## Main functionalities

The website is thought to realize many functionalities to allow the users a complete knowledge and organization of their winter holidays on ski.

The main functionalities of the website are the following:

- The users have the possibility to customize their experience registering on the website, logging-in and having the further possibility to access their private area where they can store all the booked lessons and, if they are teachers, the lessons they have to hold.
- It is possible to book a lesson on a specific ski school, with the possibility to choose the desired time slot and the preferred teacher.
- In addition, the possibility to choose the ski area holds, in order to provide an instrument to satisfy the needs of the user.
- A map holding all the ski areas in the Italian territory is present, in order to guarantee a precise localization of each element in the ski area of interest.
- Finally, there is also the possibility to visualize the features, elements and details relative to every Italian ski area, i.e. the tracks, the huts and the ski lifts offered and information about them all.

## Design choices

As regards the design of the webpage, we decided to employ a responsive design bootstrap based, in such a way to make the visualization clean and fancy and to make it adaptable for desktop devices, as well as for mobile smartphones and tablets.

The choice of colors reflects the theme as the website is about Ski, the main colors are blue, white and light grey, to make the user immerse into the winter's environment.

We chose to not use a pre-made template, in order to improve our abilities in designing the pages. They had to reflect our own theme and be personal; for this reason, we created our personal logo and style, bootstrap based as anticipated, to speed up the workflow.

## Header and Footer

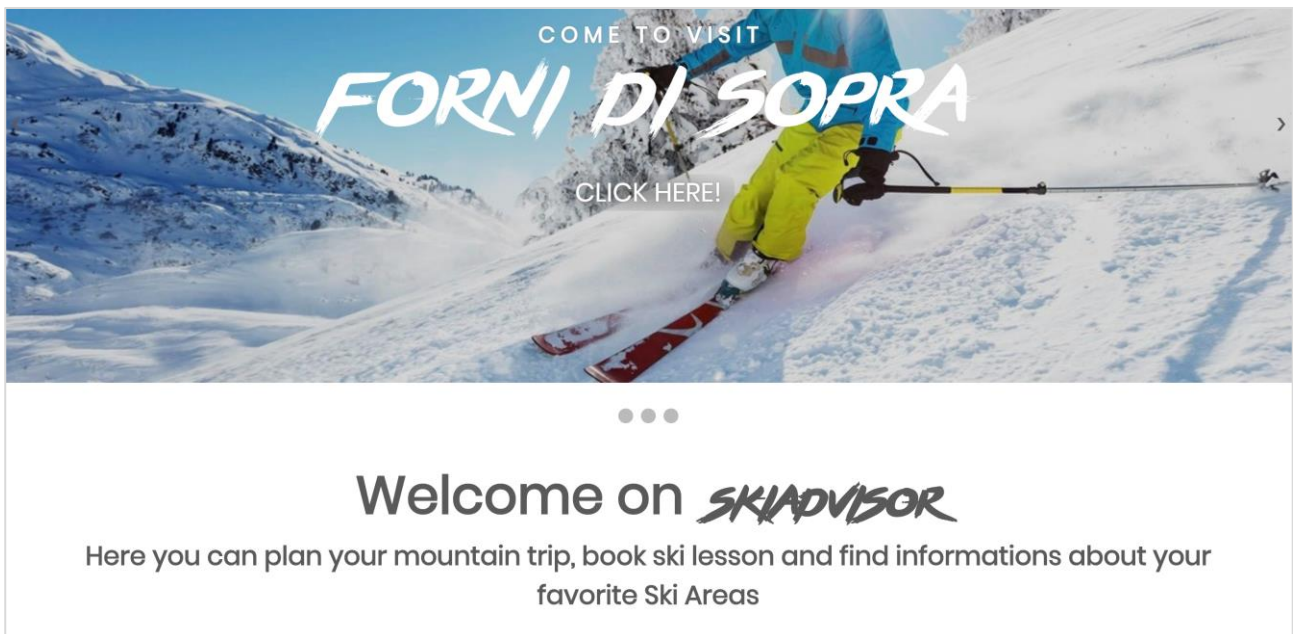
The header and the footer are dynamically included into each page, if the website is running on the server.

Note that if we try to open any html page without loading the website on Tomcat, we'll get a Cross Origin Error. If we would open directly the html page, we should disable the header and footer inclusion in the utilities.js file.

- The header** is fixed on the top of the page and it is composed of an upper navbar, includes the application logo, the title, and three links respectively to the homepage, to the sign-up page and to the login page; there is in addition a lower navbar, containing the links to all the other pages.

In the mobile visualization the upper navbar collapses in a hamburger menu while the lower navbar shows only the three main links: the explore page, the research page and the ski area page.
- The footer** is fixed on the bottom of each page and contains the names and ids of team workers (more information about them in the about us page). Also, a link to the top of the page is present.

## Home Page



The home page is designed in such a way to emphasize the main features and services offered by the website. The page is structured in two main sections, separated by a short presentation of the website, i.e. a slideshow and a set of images with descriptions presenting the main functionalities of the website.



#### Ski Areas' Huts

Once you have chosen your destination, you can look for the best hut where you can have lunch and rest!



#### Book Ski Lessons

Learn skiing with ski lessons with qualified ski instructors. Check their rates and choose your favorite!



**The slideshow**, in which the most important Italian ski Areas are shown, is positioned at the top of the page. It works both in automatic and manual way. The images automatically change after 5 seconds, but there are also two buttons (prev/next buttons) available to allow the user to do it manually; also some dots at the bottom of the slideshow are present to indicate how many images are there inside it and they can also be used as navigators.

When the User clicks one of the slides, he is automatically redirected to the Ski Area page.

**The presentation** part is made up of a short welcome message addressed to all the web page visitors. After the presentation there are four images in which the main services of the website are presented:

- Help users to search for the destination that better fits their needs.
- Booking ski lessons.
- Finding huts located in a certain Ski Area.
- Exploring the map in order to find Ski Areas in a desired location or find the location of some Ski Areas of interest.

When the user clicks one of these images, he is redirected to web page that provides the selected service.

## Ski Area Page

name	open	color	length	avg.ev.LS	avg.ev.HS
track0	No	blue	632m	1.70	4.52
track1	No	green	146m	3.71	3.12
track2	No	black	349m	0.04	5.10
track3	No	red	390m	0.71	5.13

The Ski Area page is the page dedicated to the specific Ski Area, which gives the possibility to view information about each hut, ski lift or ski track of every ski facility owing to every Italian ski area. The page is structured in the following three main sections:

- **The title**, which shows the name of the page, together with the name of the selected ski area. Under the title there's a link to redirect to the Explore page where the map is located.
- **The selection menu**, which comprises a dropdown to select the desired ski area from a predefined list, and a booking button which redirects to the Booking Page.
- **The content**, which is composed of a side tab menu, from which is possible to select the tab you want to visualize among ski lifts, ski tracks and huts of the selected ski area, and a section where all the details about them are visualized. That visualized content depends on the ski area selected from the dropdown, and on the tab chosen from the side menu. The ski lifts, the ski tracks and the huts are ordered according to their membership of a specific ski facility of that ski area. In addition, also an image of the selected ski area is visualized.

The tab menu is managed exploiting bootstrap functions, while the selection dropdown is managed via JavaScript, making use also of some jQuery functions. The content is ephemeral, retrieved from an array, generated and randomly populated into the JavaScript code. The real content will be given as a query sent to the database from the server, with the following Json structure.

The filter will be sent to the server with a POST request structured as follows:

```
{
  "data": {
    "skiareaname": "San Martino di Castrozza"
  }
}
```



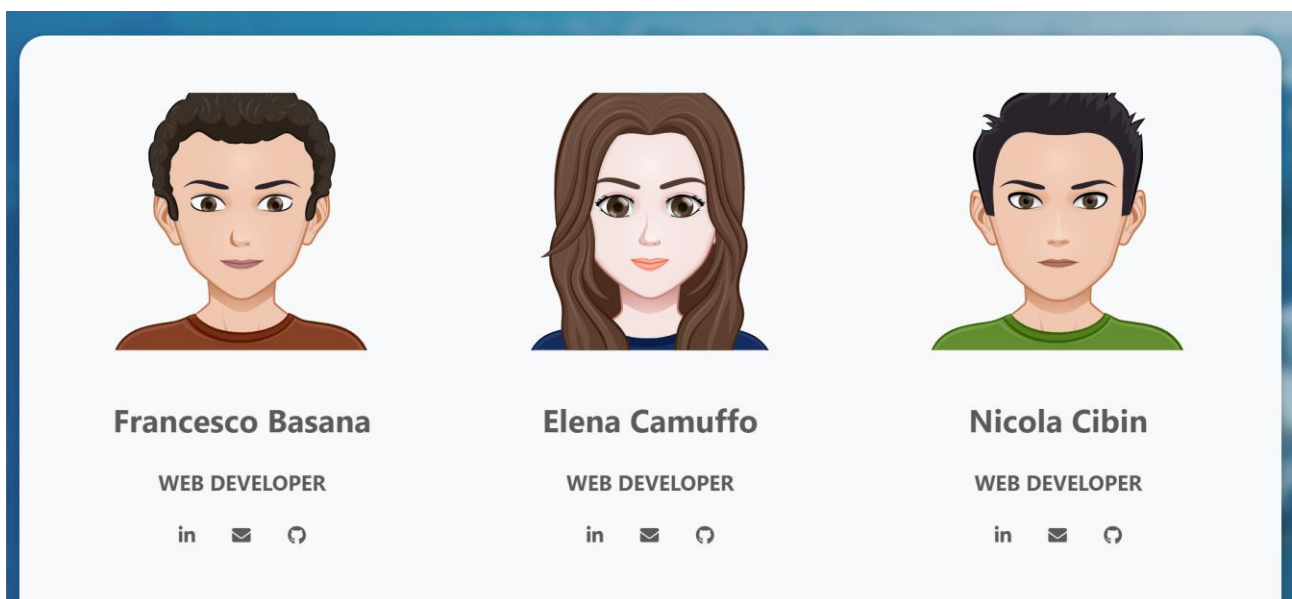
and we will get an answer structured as:

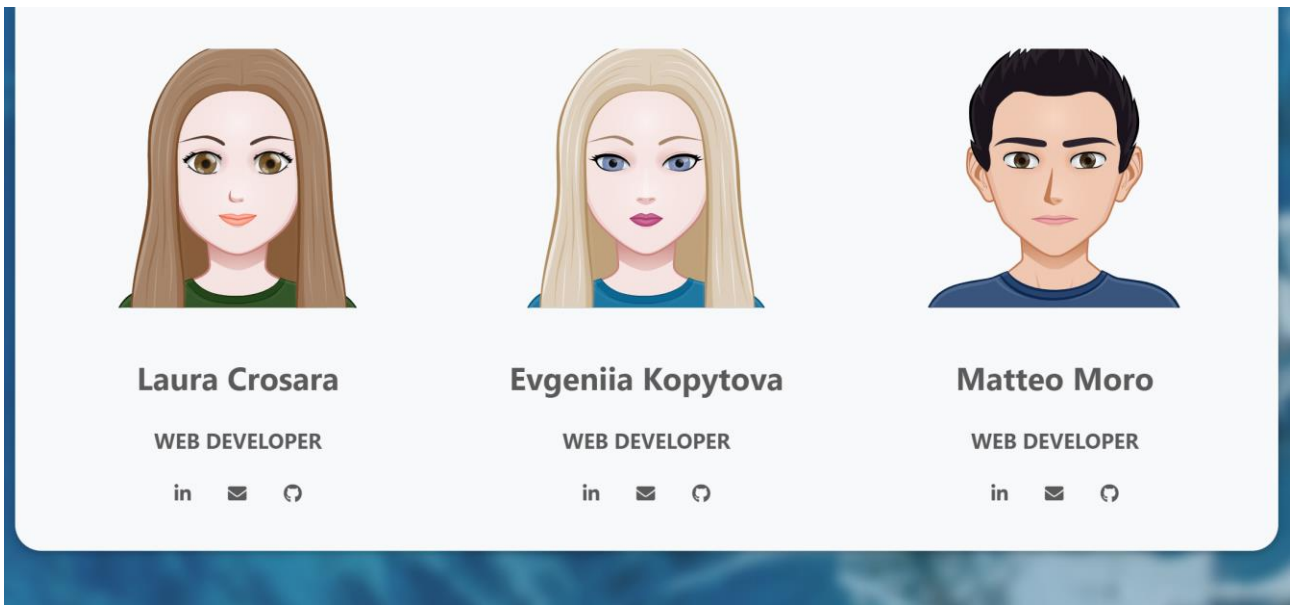
```
{
  "data": {
    "skifacility": [
      {
        "name": "Ces e Tognola",
        "skitracks": [
          {
            "name": "Valbonetta",
            "open": "Yes",
            "color": "blue",
            "length": 250,
            "avgevalLS": 4.5,
            "avgevalHS": 4
          },
          {
            "name": "Rekord",
            "open": "No",
            "color": "red",
            "length": 400,
            "avgevalLS": 3.5,
            "avgevalHS": 3.2
          }
        ],
        "skilifts": [
          {
            "name": "punta Ces",
            "open": "Yes",
            "capability": 4,
            "tipology": "chairlift"
          }
        ],
        "huts": [
          {
            "name": "Malga Ces",
            "open": "Yes",
            "avgeval": 4,
            "accbyfoot": "Yes"
          }
        ]
      }
    ]
  }
}
```

## About Us Page

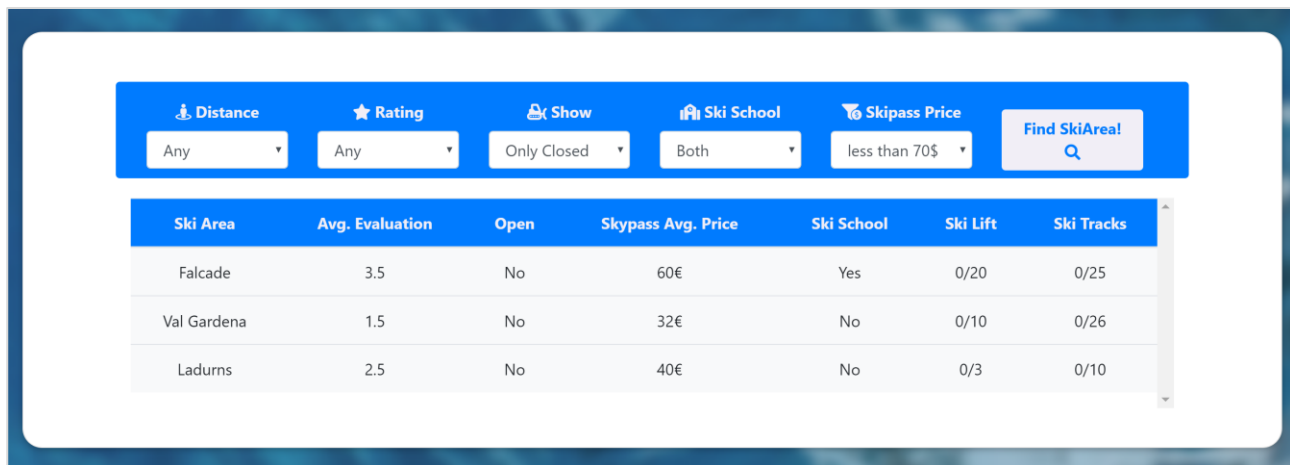
This is the page which gives information about the team workers. We decided to put this page into the website to make the users know who we are. Because of privacy we design these avatars for this page, instead of putting our photos.

From this page it will be possible in a future improvement to be linked to our LinkedIn and GitHub profiles. It will be also possible to visualize the email addresses of ours.





## Search Page



The search page is structured in such a way to make possible to find the most appropriate ski area w.r.t the users' needs.

In the head of the page we can find a **brief introduction** to the page functionalities and immediately below **some filters** are available to be applied to the ski area table.

The distance filter is the only one that now is only a mock-up, it will be effectively implemented in the back-end part of the website, instead all the other ones are effectively applicable to the table below. Users can select the minimum average rating of the ski areas, select to show only the open ones or the ones that have a ski school available, and set a maximum ski pass price.

When a user has set the desired filter, he can simply click on the search button and look for the ski area that satisfy best his own needs.

Given the fact that we don't have a back end and a database for retrieving the data so far, the ski area information is stored in an array in the JavaScript file. Obviously, in the final release, the filtering will be done through a query sent to the database from the server.

If the filters applied are too restrictive and no ski areas can be found, a notification is shown to the user and he is invited to relax the parameters in the search bar.

The following lines are the ones used to automatically generate the table starting from the data stored.

```
78 function generateTable(table, data) {
79     var tbody = document.querySelector("tbody");
80     for (var element of data) {
81         var row = tbody.insertRow();
82         for (key in element) {
83             var cell = row.insertCell();
84             var text = document.createTextNode(element[key]);
85             cell.appendChild(text);
86         }
87     }
88 }
```

When a user has set the filters and wants to filter the table, the filters value will be sent to the server with a POST request with the following structure:

```
{
  "data": {
    "distance": 20,
    "rating": 1,
    "open": 0,
    "skiSchool": 2,
    "avgPrice": 60
  }
}
```

In the "open" and "skischool" fields, the value range is [0,2], where:

- "0" stands for "show only closed ski area"
- "1" stands for "show only opened ski area"
- "2" stands for "show both closed and opened ski area"

After that the server has received the previous request, it will query the database in order to retrieve the ski area that satisfy that parameters and will respond to the client with the ski area list structured as follows:



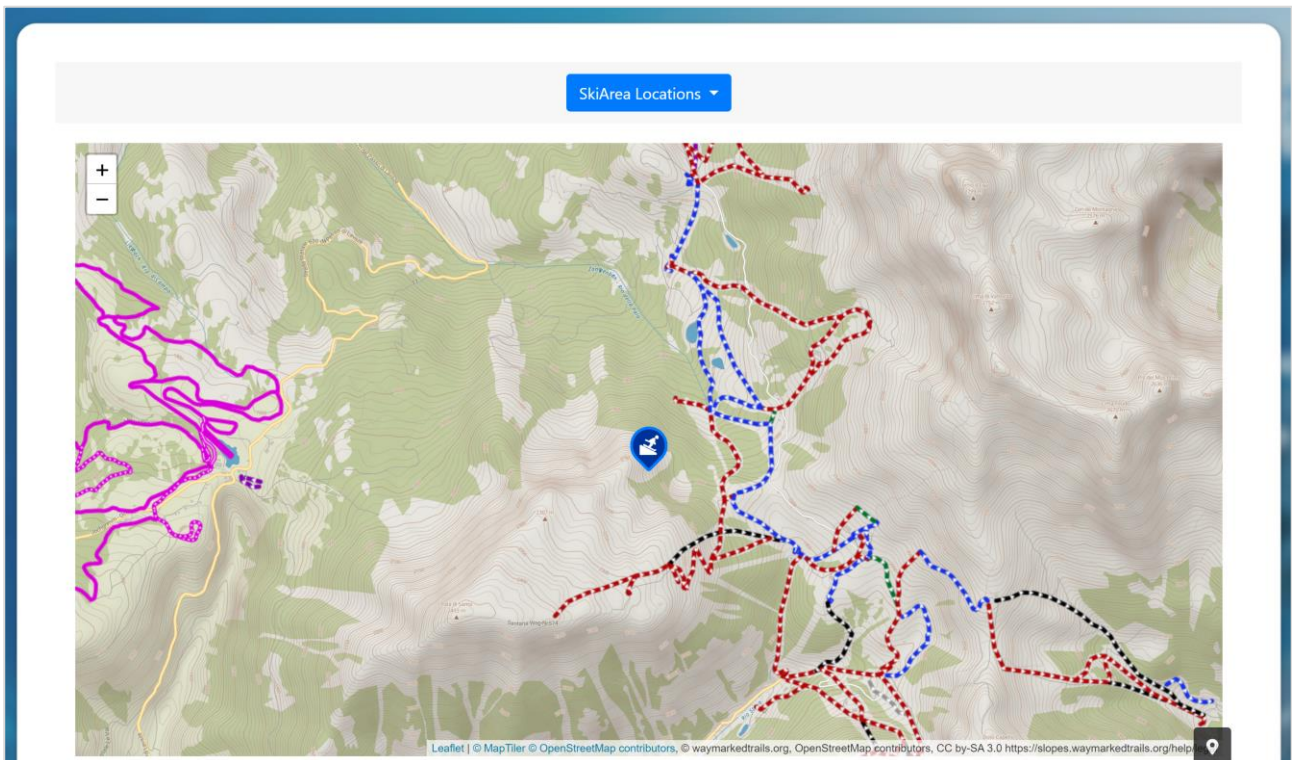
```

{
  "data":{
    { "name": "Civetta", "avgEval": "4.5", "open": "Yes", "avgPrice": "65€", "skiSchool": "Yes", "skiLift":"10/10", "skiTrack":"20/20"},
    { "name": "Falcade", "avgEval": "3.5", "open": "No", "avgPrice": "60€", "skiSchool": "Yes", "skiLift":"0/20", "skiTrack":"0/25"},
    { "name": "Latemar", "avgEval": "3.7", "open": "Yes", "avgPrice": "55€", "skiSchool": "No", "skiLift":"7/10", "skiTrack":"27/30"},
    { "name": "Falcade", "avgEval": "3.2", "open": "Yes", "avgPrice": "72€", "skiSchool": "No", "skiLift":"18/18", "skiTrack":"30/30"}
  }
}

```

In the response we will obtain the number of ski lift and ski track for each ski area, expressed as “open”/ “total”.

## Explore Page



The explore page is structured in such a way to provide the user an easy interaction with the map.

The page is pretty simple and consists in two parts:

- In the header we can find the title and a link back to the search page. Under the title there is a dropdown menu in which are stored all the ski area location. For the moment, the data displayed are only mockups but, when the backend will be implemented, the real data will show up. The additions of the dropdown items are achieved dynamically.
- In the content area of the card layer we can find the map, which is implemented using Leaflet, a JS library for interactive map implementation. The map consists of two layers: the topological one, that acts as the base layer, and the ski tracks one, that shows all the tracks of all the ski areas. The map, provides in addition the following features: geolocalization of the user,

markers on all the ski areas stored in the server and clusters of markers to have an estimation of how many ski areas are in a region. In addition, by selecting an item in the dropdown menu, the map will go on the coordinate of the location clicked, plus when a marker is clicked the user will be redirect on the page related to that ski area.

When the page is loaded, the client sends a GET request in order to obtain the name of the ski area, its coordinates and the link that redirects to Ski Area page of that ski area.

```
{
  "data": {
    "name": "Civetta",
    "coords": {
      "lat": 46.4126997,
      "lng": 12.0393272
    },
    "href": "../html/skiarea.html"
  }
}
```

## Booking Page

The booking page is used by the users to book a ski lesson in a ski school.

Users are redirect to this page from the “Book Lesson” button on the ski area page and on the personal page. In order to book a lesson, skiers have to choose a date through the “date picker”, a time slot and a teacher through the relative dropdown menu.

In the final website version, we will perform a check with the database in order to propose to the user only the available data, time and teachers. By now, only an alert message is visualized to notify if the form is compiled correctly.

After the booking has been done, the user can check the lesson booked in his profile page. The lesson appears also on the reserved page of the selected teacher.

For the data selection, we’ve chose to implement the data picker offered by “gijgo.com/datepicker” because the one available in html5 will be rendered in different ways dependently on the selected browser. We’ re still looking for a better calendar because this one has too few functionalities and it would limit the function that we will provide to the user.

When the page is loaded, the client sends a GET request in order to obtain date and hour where there are already too many bookings and these time slots will be disabled in the calendar.

The request structure follows:

```
{
  "data":{
    "skiSchool": "Civetta Ski School",
    "month": "12",
    "year": "2020"
  }
}
```

In the response from the server we will have the day number as label of the array that will contain the starting hour of the time slots already full.

```
{
  "data":{
    "10": [9, 13],
    "22": [9, 12, 15],
    "30": [10, 11, 12, 13, 14]
  }
}
```

## Personal Page

The personal page of the website visualizes the information about registered and logged in users. It consists in two parts, i.e. the normal user part and the administrator part, which is hidden for normal users within back-end implementation.

**Part “About”** corresponds to both – normal user and the ski teacher. It includes general information such as: username, name and surname of the person, age, ski level and fiscal code. Under the general information block there is a table which describes all the bookings done by user. It includes ski school's name, date, time slot and the teacher name. In addition, there is a button which allows users to delete their bookings. At the end of the page there is a button – “New booking” which is linked to the booking page in order to make a new booking.

**Part “Teacher Info”** will be available only for teachers and labeled as an administrator as well. The main information block includes teacher license, average evaluation, lesson capacity, available mark, and number of attended lessons. These parameters are supposed to be dynamic except for the teacher license. Moreover, the table below allows to see all the bookings made by users to that particular teacher.



## Test Test

LABEL: Administrator

About Teacher Info

User Name test123  
Name and Surname Test Test  
Age 22  
Ski level 6  
Fiscal Code TST TST 91M15 L151K

#	Ski school	Date	Hour	Teacher name	
1	Scuola Civetta	01/02/2020	10:00 - 13:00	Andrea Bianco	
2	Scuola Civetta	02/02/2020	10:00 - 13:00	Andrea Bianco	
3	Scuola Civetta	03/02/2020	10:00 - 13:00	Andrea Bianco	
4	Scuola Civetta	04/02/2020	10:00 - 13:00	Andrea Bianco	

New Booking



## Test Test

LABEL: Administrator

About Teacher Info

License 1406516513132  
Average Evaluation 6.5  
Lesson Capacity 36h  
Available ✓  
Number of Attended Lessons 6

#	Ski school	Student	Date	Hour	
1	Scuola Civetta	Mario Rossi	01/02/2020	10:00 - 13:00	
2	Scuola Civetta	Marco Bianchi	01/02/2020	10:00 - 13:00	
3	Scuola Civetta	Evgenia Bettarello	01/02/2020	10:00 - 13:00	

New Booking

As soon as the page get loaded, the client sends a GET request in order to obtain a data table such as name of the school, date and time slot of the class, teacher name. Considering the administration part of the page, the process is the same.

The delete button, when clicked, shows a popup message to the user in order to confirm that he wants to delete the booking. A POST request is sent to the server in order to delete row with information which is one of the elements of the GET array.

```

{
  "data": [
    {
      "SkiSchool": "Scuola Civetta",
      "Date": "01/02/2020",
      "Hour": "10.00 - 13.00",
      "Teachername": "Andrea Bianco"
    },
    {
      "SkiSchool": "Scuola Civetta",
      "Date": "02/02/2020",
      "Hour": "10.00 - 13.00",
      "Teachername": "Andrea Bianco"
    },
    {
      "SkiSchool": "Scuola Civetta",
      "Date": "03/02/2020",
      "Hour": "10.00 - 13.00",
      "Teachername": "Andrea Bianco"
    }
  ]
}

```

```

"SkiSchool": "Scuola Civetta",
"Student": "Mario Rossi",
>Date": "02/02/2020",
>Hour": "10.00 - 13.00"
},
{
>SkiSchool": "Scuola Civetta",
>Student": "Mario Rossi",
>Date": "01/02/2020",
>Hour": "10.00 - 13.00"
},
{
>SkiSchool": "Scuola Civetta",
>Student": "Marco Bianchi",
>Date": "02/02/2020",
>Hour": "10.00 - 13.00"
},
{
>SkiSchool": "Scuola Civetta",
>Student": "Evgenia Bettarello",
>Date": "03/02/2020",
>Hour": "10.00 - 13.00"
},
}

```

```

{
  "data": {
    "username": "testtest",
    "nameandsurname": "testtest",
    "age": 22,
    "skilevel": 6,
    "fiscalcode": "TST TST 91M15 L151K"
  }
}

{
  "data": {
    "licence": "1406516513132",
    "averageevaluation": 6.5,
    "lessoncapacity": 36,
    "avaible": true,
    "numberofattendlessons": 6
  }
}

```

The information stored in "About" and "Teacher Info" slots is also sent through a GET request in order to obtain information. Client send the GET request for "Teacher info" and the booking available on the teacher part only if user authorized as a teacher. On the teacher info part Boolean, string, integer and double types are illustrated.

## Signup

Name  
 Enter Your Name

Surname  
 Enter Your Surname

Fiscal Code  
 Enter Your Fiscal Code

Date of Birth  
 gg/mm/aaaa

E-Mail  
 Enter Email

Username  
 Choose your username

Password  
 Enter Password

Repeat Password  
 Repeat Password

By creating an account you agree to our [Terms & Conditions](#).  
 Are you a ski teacher? [SignUp as a SkiTeacher](#).

[Sign Up](#)  
 If you already have an account [Log In](#).

Name  
 Enter Your Name

Surname  
 Enter Your Surname

Fiscal Code  
 Enter Your Fiscal Code

Date of Birth  
 gg/mm/aaaa

E-Mail  
 Enter Email

License  
 License Number

Username  
 Choose your username

Password  
 Enter Password

Repeat Password  
 Repeat Password

By creating an account you agree to our [Terms & Privacy](#).  
 If you are not a SkiTeacher [SignUp as a normal user](#).  
 If you already have an account [Log In](#).

[Sign Up](#)

The Sign-Up page allows the users to register and have the full access to all the functionalities of the website. There are two different types of registration: one for the **normal users** and one for the users that are also **ski teachers**.

The registration forms ask only for such kind of information that can be useful in order to improve the users' experiences. In particular the form asks the user to insert the following information:

- **Name**.
- **Surname**.
- **Fiscal Code**, which provides a certification about the user's information.
- **Date of birth**, to know what the age of the user is.
- **E-mail**, which is used to communicate with the user for anything related to the website.
- **Username**, which needs to be unique in order to be able to distinguish between different users.
- **License** (only for ski teacher registration), is a certification that the user is a ski teacher.

Once they have correctly entering all the required data, the users must press the Signup button to submit the form. If the user forgets to enter some data, an alert dialog is opened; there the user can know which fields are wrongly filled or incomplete.

If a user enters a username that is already used, an alert dialog is opened; there the user is informed about this issue and he needs to change it before completing their registration. If all the field are correctly completed the Sign-Up button routes the user to the Login page.

If the User is just registered, he can access the website through the Login page.

To change the typology of the registration (normal to teacher or viceversa), the user has to press the "ski teacher"/ "normal user" blue link.

Using files containing JSON data we simulate the communication with the server. When the new user presses the "Sign Up" button, the website sends a request formatted in the following way:

```
{
  "data":
  {
    "username": "mariorossi1",
    "password": "prova1",
    "license": "FEURILKJDASD"
    "name": " Mario",
    "surname": "Rossi",
    "FC": "RSSMRA48H10H562S",
    "birthdate": "10/06/1948",
    "email": "mariorossi@gmail.com"
  }
};
```

If the user Signs up as a "normal user" the license field will be empty.

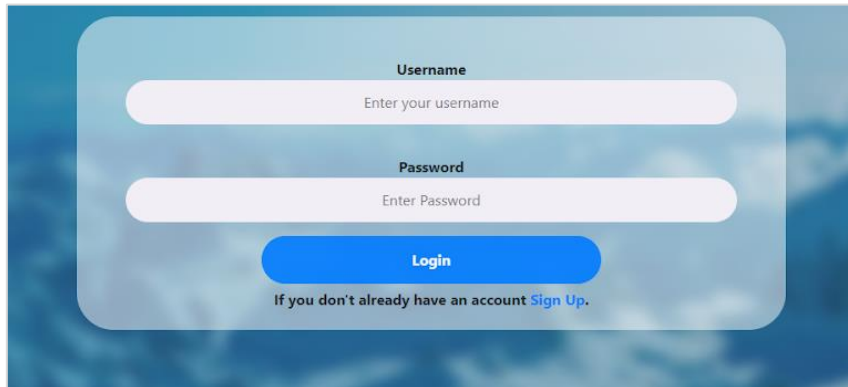
If the request is successful, a unique token Key is generated via script to have a safe reference to the user. After this generation, the user is created and inserted into the database. The following lines are the ones used to automatically generate the token key:

```
function generateKey() {
  var res = "";
  var char = "ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
  var length = char.length;
  for ( var i = 0; i < 20; i++ ) {
    res += characters.charAt(Math.floor(Math.random() * length));
  }
  return res;
}
```



Finally, when the user completes the submission, he acknowledges the “Terms & Condition” can be downloadable from the Sign-Up page press the “Terms & Condition” blue link. At the moment the “Terms & Condition” is a mockup.

## Login



The image shows a login form with a light blue background. It features two input fields: "Username" with the placeholder text "Enter your username" and "Password" with the placeholder text "Enter Password". Below the fields is a blue "Login" button. At the bottom, there is a link that says "If you don't already have an account [Sign Up](#)."

The Login page allows all the registered user to access all the functionalities of the website and have a personal area reserved.

It is structured as a simple form with two fields, Username and Password, and it is submitted by means of the apposite button.

If the user enters a wrong data an alert dialog showing what was wrong is opened. If the user is not yet registered, he can access to the sign-up page pressing the sign-up blue link. If the two fields are correctly completed, the Login button routes the user to the personal page.

Here is a simulation of a connection with the server. When the user presses the "Login" button the website sends a request like:

```
{
  "data":
  {
    "username": "mariorossi1",
    "password": "prova1",
  }
};
```

If the User already exists, the request is successful, and the response of the server will have the following structure:

```
{
  "data":
  {
    "username": "mariorossi1",
    "password": "prova1",
    "token_key": "J1Hj2kjH7LI5nb3jk4fK"
  }
};
```

The “token\_key” is the reference key generated in the sign-up process, which allows to know whether the user has just logged in.