.NET Web Development – Final Test

Using frameworks ASP.NET Web API, Entity Framework and Bootstrap, jQuery Library and Partial View to implement a Web application for keeping records of festivals. The application should provide work with the following entities:

Place:
- Id - identifier;
- Name - text value;
- Postal code - numeric value with a maximum 5 digits.

Festival:
- Id - identifier;
- Name - text value;
- Ticket price - decimal value;
- Year of the first festival maintenance - numerical value less than 2018;
- Place - connection with the instance of the Place class (one festival can be maintained only in one place).

1) Use the working framework ASP.NET Web API and Entity Framework to implement next REST API:
   a) **GET api/mesta** - get all places;
   b) **GET api/mesta/{id}** – get place by Id;
   c) **GET api/mesta?kod={vrednost}** - get all places with a post at less than the forward value, sorted increasing by the postal code;
   d) **GET api/festivali** - get all festivals, sorted by ticket price decreasing;
   e) **GET api/festivali/{id}** – get festival by Id;
   f) **POST api/festivali** – add new festival;
   g) **PUT api/festivali/{id}** – modify an existing festival;
   h) **DELETE api/festivali/{id}** – delete an existing festival;
   i) **POST api/festivali/pretraga** - find all the festivals whose first year is holding a festival between two entered values (**start** and **end**) sorted by first year of the festival increasing;
   j) Registration and user login.

2) Using the Bootstrap framework, jQuery library and Partial View, create the following Single Page Application:

   a) During the first launch, the page contains a user registration form and a spreadsheet displaying the festivals by functionality 1) d), as shown in Figure 1.

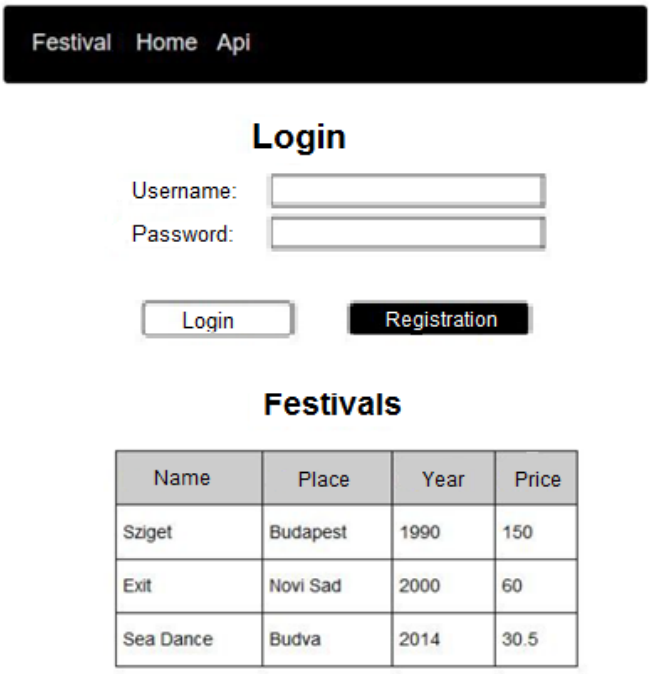

Figure 1

b) If the user presses the **Registration** button, instead of a login form,the registration form is displayed, as shown in Figure 2. If the user has successfully registered, instead of a registration form the login form is displayed (Figure 1). If an error has occurred during registration, the user will be notified with an alert.

c) If an error has occurred when the user tries to log in by pressing the **Login** button, it is necessary to inform the user withan alert. If the user has successfully logged on to the system, contents of Figure 3 will be displayed.

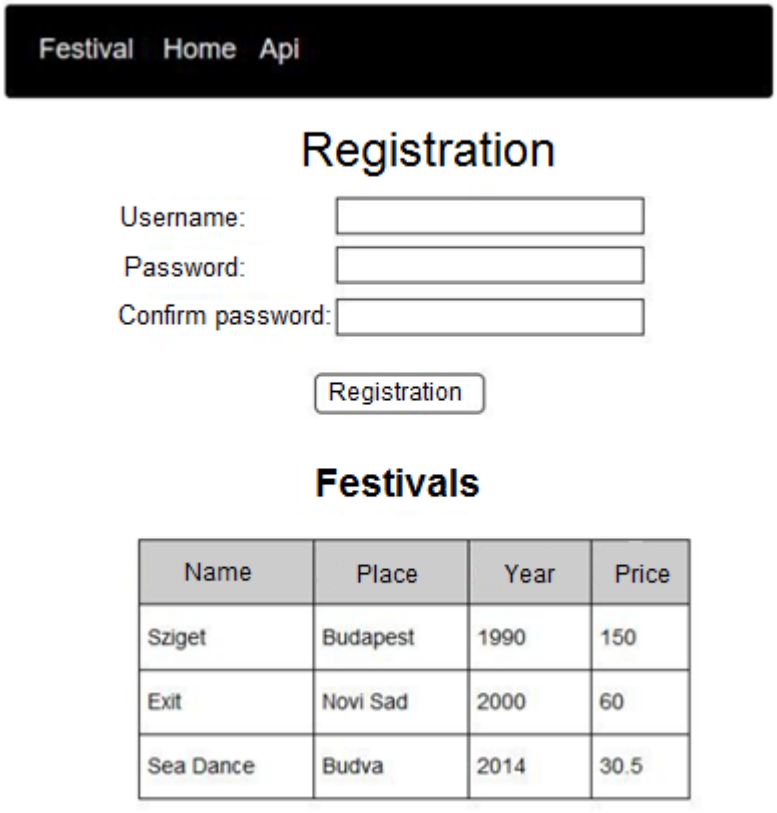

Figure 2

d) After the **Logged** in user label, the username of a logged in user is displayed. By clicking the **Logout** button, the user is logged out of the system and 3) a) content is displayed.

e) The festival table should be expanded by another column on the right, which adds a Delete button beside each festival. By clicking the **Delete** button, the selected festival is deleted and the festival table is refreshed.

f) Below the table, there is a form for searching the festivals, which follows the functionality 1) i). Festivals that meet the search criteria are displayed in the festival table.
If there was an error during the user populating a year or getting the festival from a REST API, notify the user with an alert.

g) Below the search form there is a form for adding a new festival, which follows the functionality 1) f). The festival's destination is picked from the drop-down menu that is filled with the functionality 1) a). By clicking the **Add** button, a new festival is added.If the addition was successful, the Add form fields will be cleared and the festival table will be refreshed. If an error occurs, the user will be notified via alert. By clicking the **Quit** button,Add form fields will be cleared.

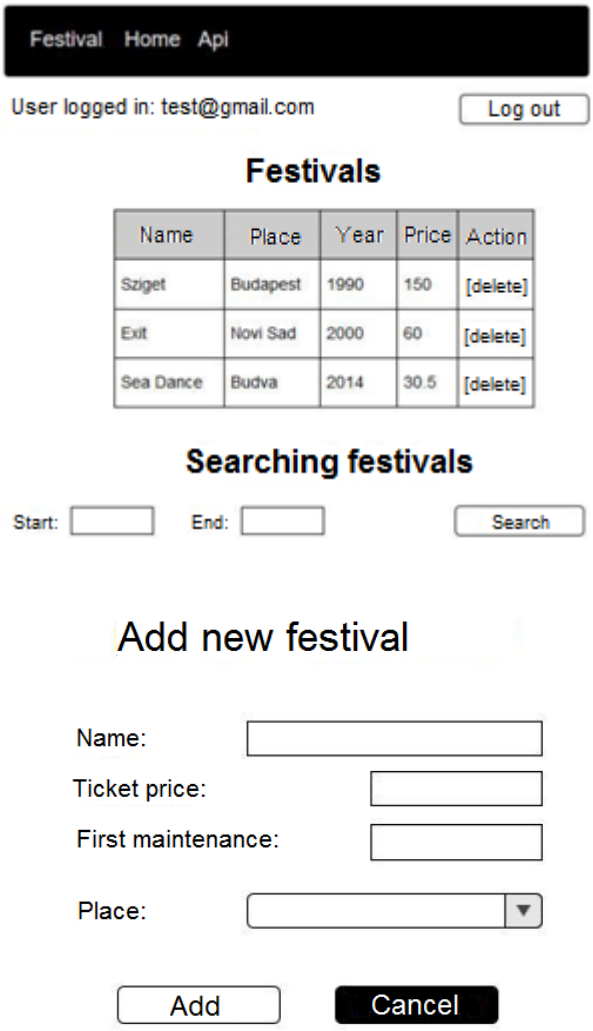

Figure 3