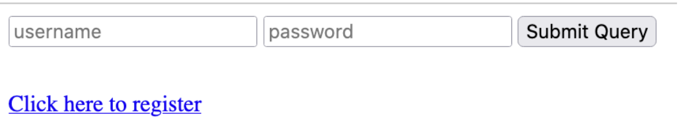
ASSIGNMENT 2

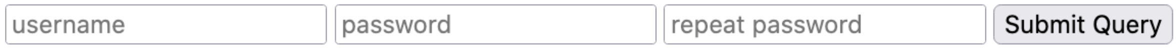
The following webapp must be developed by using an MVC pattern based on Servlets + Beans + JSPs. Code in the JSP must be minimal. The main point is to demonstrate understanding of how and where data have to be kept.

1) A game on the web requires the user to identify him/herself (Authentication), and if not yet known to the system to register (Registration).
Authentication is required if the last access occurred more than T seconds ago, where T is a parameter of the WebApp [1], otherwise a known user need not to authenticate and proceeds to the start page.

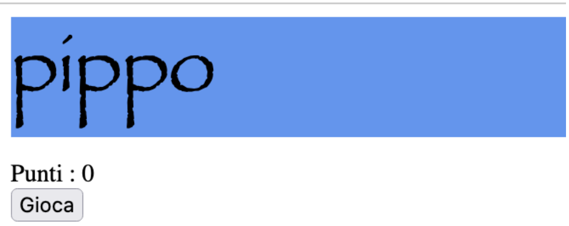
Authentication:



Registration:



2) After authentication the user (not admin) sees the start page that says: "your score is" and the score (initially zero: it resets every time the user passes from login) and shows the button "Play" ("Gioca").

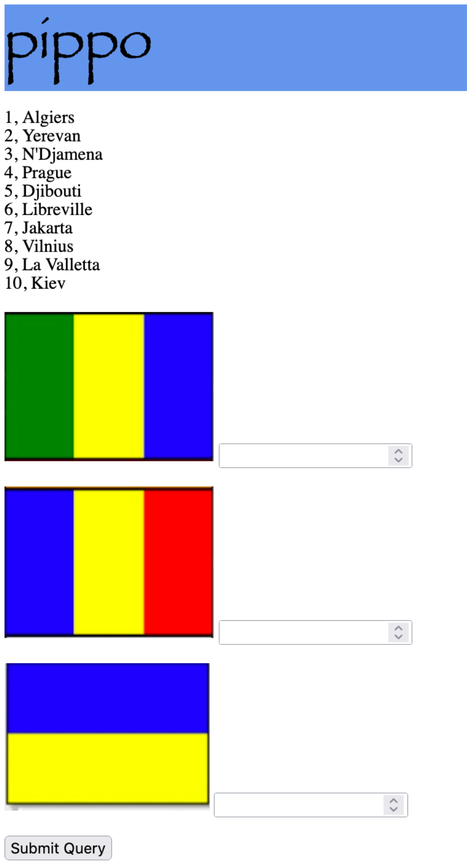

pippo

Punti : 0

3) All pages (except authentication / registration pages) show the same header containing the user name.

4) Each game session consists of associating the respective capitals with three flags of states. The three flags are chosen at random from a given pool.
By pressing the Play button the game page is presented, which shows the numbered list of all the capitals of the pool, and a form with three flags chosen at random and arranged vertically, and next to each flag an empty text field. At the bottom of the form there is a "Submit" button.

ASSIGNMENT 2


pippo

1, Algiers
2, Yerevan
3, N'Djamena
4, Prague
5, Djibouti
6, Libreville
7, Jakarta
8, Vilnius
9, La Valletta
10, Kiev

5) The user must enter the number indicating the capital corresponding to the flag in the text fields (the number is chosen from the list above). The text field is restricted to accept only numbers[2]. The submit button performs the submission only if all the fields have been filled in[3]. These checks are performed on the client side, and re-checked on the server side.

6) At this point the system checks the answers: if they are all correct it increases the score by three points, if at least one is wrong it decreases it by one point. In both cases it returns to the start page.

7) Attempts to access the home and game pages by unauthenticated users redirect to the authentication page.

8) The manager (predefined user, authenticated with admin, pw nimda) instead of going to the start page goes to a control page where s/he can see the list of all active users (i.e. currently connected), and for each of them what the current score is. (To update the list a page reload in needed) .
Attempts to access the control page by users other than admin generate a 401 error.

9) All data are maintained as long as the webapp is running. At webapp shutdown [4] user identities and scores are saved to a file (use serialization [5]). After redeploy data read back from file.

The images of the flags can be found on latemar.science.unitn.it/esameWeb/bandiere.zip

ASSIGNMENT 2

Suggestions:
[1] https://www.baeldung.com/servlet-session-timeout
[2] https://www.w3schools.com/html/html_form_input_types.asp
[3] https://www.w3schools.com/html/html_form_attributes.asp
[4] https://mkyong.com/servlet/what-is-listener-servletcontextlistener-example/
[5] https://mkyong.com/java/how-to-read-and-write-java-object-to-a-file/