

2022-12-19 - Back-end 2nd Chance Exam

Mondragon Unibertsitatea

Computer Science Degree, 3rd course: Web Engineering I

You will have to make a CRUD for **SportResults**.

Execute the following command on MySQL for creating the table and inserting some data.

```
DROP TABLE IF EXISTS sport_result;
CREATE TABLE `sport_result` (
  `sportResultId` int(11) NOT NULL AUTO_INCREMENT,
  `team1Name` varchar(256) NOT NULL,
  `team1Result` int(11) NOT NULL,
  `team2Name` varchar(256) NOT NULL,
  `team2Result` int(11) NOT NULL,
  PRIMARY KEY (`sportResultId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
INSERT INTO `sport_result`
VALUES
(1, 'teamA', 1, 'teamB', 2),
(2, 'teamB', 1, 'teamC', 3),
(3, 'teamC', 1, 'teamA', 4);
```

Copy also `sport_result.css` to css folder and all Team images to `images` folder.

WARNING. Created classes should have meaningful names. For example, if you are creating a DAO class for SportResults, you can call it **DaoSportResult.java** or **DaoSResult.java**. Do not call it **DaoExam.java**, **DaoAdvertisement.java**, **DaoSDG.java** or anything that you had prepared before that will make it more difficult to maintain in the future.

Images

What do we want to achieve?

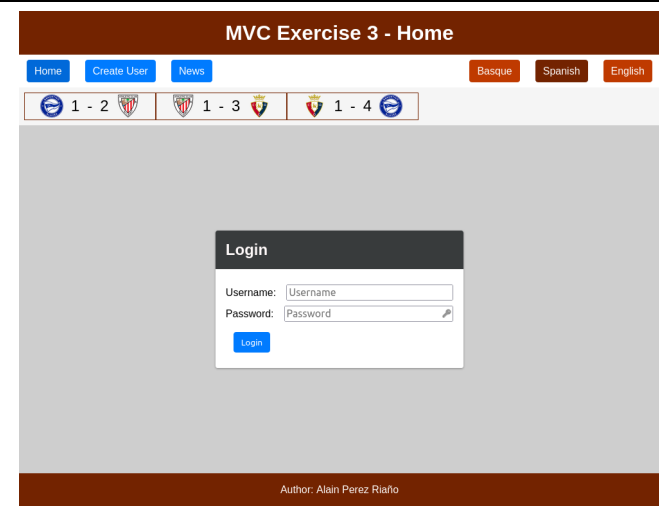


Image 1: View before login

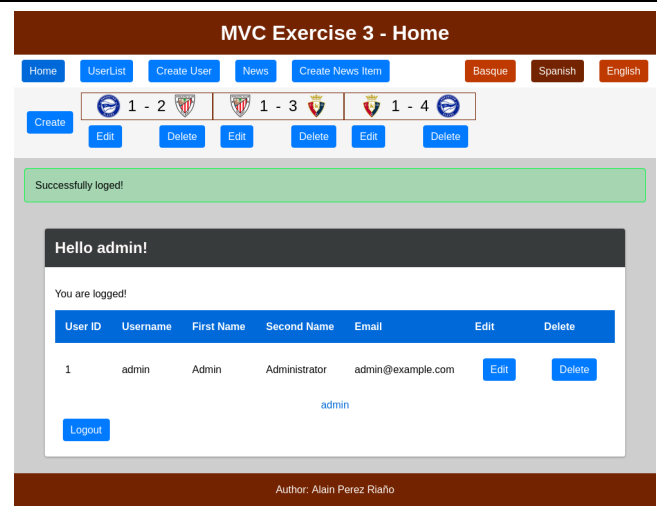


Image 2: view after login

Create form

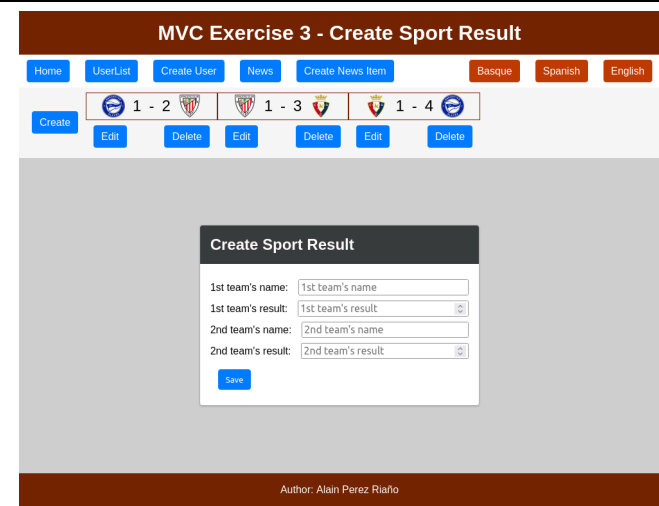


Image 4: Creation form

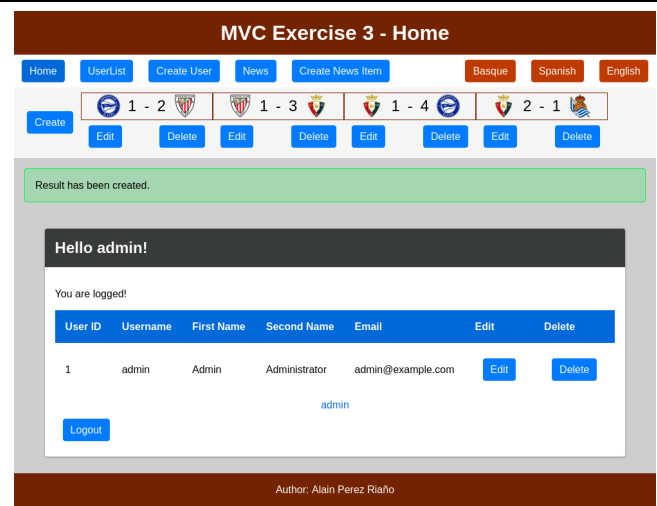


Image 4: view after creation

Edit form

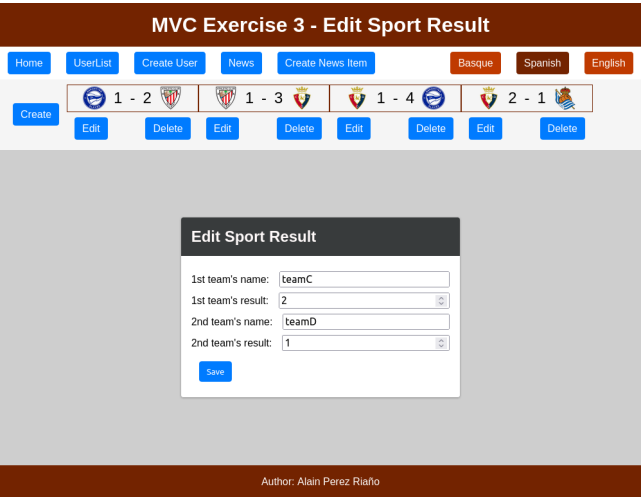


Image 4: Edit form

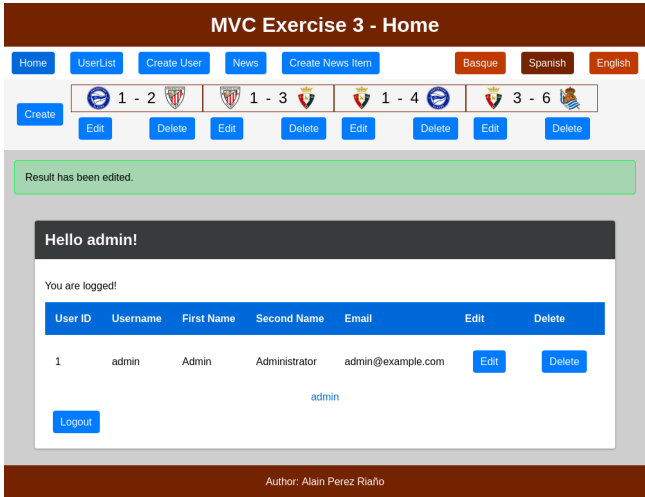


Image 4: view after edition

Create (4p)

- Add a link ("Create") that goes to SportResult creation form.
- The best place is a section between navigation and notifications, if you are not able to put it there, add the link in the header with the other buttons.
- Add error/success messages after creation.
- Follow all the patterns seen at class.

WARNING. Controllers, filters, etc. should answer to `/sportResult/xxx` or `/result/xxx` paths, no `/advertisement/xxx` nor `/exam/xxx` nor anything different not related to Sport Results (check the template below to see the paths you will have to use).

View (3 parts: 1p + 1p + 1p)

Load from database (1p)

- After SportResult has been created, send the user to somewhere where the SportResult is visible. The correct place would be `index.html` if the `sport_results` section is correctly loaded. There is no need of a view for a single SportResult, but if you are not able to make the next part of the exam, you can show it in a single SportResult view.

View template (1p)

- Modify the template(s) so SportResults look like this:

```
</nav>
<section id="sport_results">
  <a class="action_button" href="/sportResult/create">Create</a><!-- if
permitted -->
  <span class="sport_result_group">
    <span class="sport_result">
      <!--src from the name -->
      <span class="result">1</span>
      -
      <span class="result">2</span>
      
    </span>

    <span class="sport_result_actions"><!-- if permitted -->
      <a class="action_button" href="/sportResult/1/edit">Edit</a>
      <a class="action_button"
href="/sportResult/1/delete">Delete</a>
    </span>
  </span>

  <span class="sport_result_group">
    <span class="sport_result">
      
      <span class="result">1</span>
      -
      <span class="result">3</span>
      
    </span>

    <span class="sport_result_actions">
      <a class="action_button" href="/sportResult/2/edit">Edit</a>
      <a class="action_button"
href="/sportResult/2/delete">Delete</a>
    </span>
  </span>

  <!-- as many sport_result_groups as SportResults in the database -->
<div id="notifications">
...

```

- **WARNING** the code above is the resultant HTML, not what you have to add to the template.

- Both CSS files must be loaded:

```
<link rel="stylesheet" href="/css/style.css" />
<link rel="stylesheet" href="/css/sport_result.css" />
```

- The creation link can be seen in the section.
- SportResult list should also be visible in every page, not only in index.html (news list, news form, user list...).

View scope and loading (1p)

- The list of SportResults must be loaded from the database as soon as possible.
- Create a filter that is executed on `/*` path.
 - It will load the list of SportResults if they are not already loaded.
- The list of SportResults should be shared by all the users in the application (even if they are not logged in), so choose the correct scope.
- The list must be reloaded after a SportResult has been created, edited or deleted.

Delete (0.5p) & edit(1p)

- Add delete/edit buttons after SportResults (see given HTML code).
- Edit form must have all the data pre-loaded in the inputs.
- Delete button will remove the SportResult from the database.

Permissions (1p)

- Create a filter to manage SportResult permissions.
- Anyone can see SportResults.
- Only logged users can see SportResult forms.
- Only logged users can create/delete/edit SportResults.
- Only logged users can create/delete/edit buttons in the section above footer.

Multilanguage (0.5p)

- Do not use literal strings (everything with `fmt:message`).
- Remember success & error messages.
- Add strings to resource files.
- SportResult content inserted in the database is not multilingual. Ignore that part.

Tips

How to get application scope context in a controller/servlet:

```
ServletContext context = getServletContext();
```

How to get application scope context in a Filter:

```
HttpServletRequest request = (HttpServletRequest) req;  
ServletContext context = request.getSession().getServletContext();
```

or

```
ServletContext context = fConfig.getServletContext();
```

How to set/get values from context:

```
Object o = context.getAttribute("variableName");  
context.setAttribute("variableName", variableValue);
```