

## Project description

The aim of this project is to create an online tournament system. The system should allow its users to conduct tournaments in a discipline selected by students. The organizers can set a time and place of the tournament, limits on the number of participants and sponsor logos while competitors can apply to participate and enter the results of their encounters.

### Functional requirements

1. Each user must register in the system providing his first name, last name, email address (used also as a login) and a password. After registering, the system sends an email to the given address with a confirmation link which expires in 24 hours. The user account becomes active only after it is confirmed by the link.
2. The system allows its users to log in (+ *I forgot my password* option).
3. The main page displays a list of upcoming tournaments (with paging, 10 tournaments per page) and a simple search mechanism (the list is also visible to users who aren't logged in).
4. After clicking on any given tournament, a details page is displayed which basic tournament information (name, discipline, organizer, time, google maps with location, max participants, participation application deadline, sponsor logos, number of ranked players) with the option of applying to participate in the tournament. If the logged in user is this event's organizer, he can also edit the tournament information.
5. After selecting the option to sign up for the tournament the system asks for a license number and current ranking (at this point, users who aren't logged in are asked to log in). Assume that both license number and ranking need to be unique. For simplicity, you can assume that this data is not verified by the organizer.
6. There can only be as many participants as the organizer specified via the limits.
7. After the participation application deadline passes, the system selects the tournament ladder, seeding the seeded players according to their ranking. The way the ladder is organized and the players selected as well as the system of play should correspond to a discipline selected by the student.
8. After selection, users can view the ladder on the tournament page. From now on, the users can enter the results of their games played. For simplicity, you can assume that it is sufficient to select the winner of the encounter, however, both participants have to select the same winner. If the participants enter conflicting results, both results are withdrawn and they can reenter the result.
9. When the result is entered successfully, the tournament ladder (scoreboard) is updated.
10. The ladder/scoreboard should be presented with some sort of visualization (a table is not enough to fulfill this point).
11. Users can easily view their upcoming games and tournaments for which they are registered.
12. Each user can organize his own tournament. All required information needs to be provided. You can't host tournaments in the past.

### **Non-functional requirements**

- ORM mechanism, but using queries in certain specific cases is allowed
- Irreversible operations require additional confirmation in the form of dialog boxes
- Remember about variable binding, data validation, post/redirect/get, and proper password management (<https://crackstation.net/hashing-security.htm>)
- An important element of grading will concern concurrent data access (e.g., participants limits, entering game results, etc.)

### **Remarks**

To achieve the highest score of 4 points, the system needs to be complete and flawless. However, partial solutions are also accepted according to the following thresholds:

- 1 point – requirements 1, 2, 3 (without search), 4, 12
- 2 points – the above requirements + requirements 5, 6, 7, 11
- 3 points – the above requirements + requirements 8, 9

Solutions need to be presented personally. Plagiarism results in both students failing the entire course.