

Software testing

Creating a new team

When a new user opens our site, there is a button for the user to create a team. This button redirects the user to a form for creating an account and team. The form asks for the following information: team name, name, 2x password, email, number, allergies and if the user has a car. When the user has filled everything in correctly it can click on the create team button. Then, the client performs several checks, to ensure correctness. First of all it checks if the two passwords are equal. Then, a check is performed to ensure that the email and the team are unique, thus not yet in the database. Lastly it checks if an answer is chosen for the question if the participant has a car. We have tested these checks thoroughly. multiple weird inputs have been checked, such as filling in nothing at all or using teams and emails that are already in the database. After having some incorrect results, the methods have been updated. At this point in time all the checks are working properly, except the one that checks if one of two boxes has been checked for the car question (this is always true at this moment). We will fix this bug later.

After all the checks succeed, the data should be stored in the database, and the user will be redirected to the participant pages. The database should contain a new row in the team table, and a new row in the person table. The tests show that this is also working correctly. All the data is properly stored inside the database, after which the user is correctly redirected.

joining an existing team

A new user can also make a new account by joining a friend's existing team. This friend can invite the user via a link. This link contains the name of the team as a value for now, but this will be changed later to a more secure code. When a new user opens this link it will open a form in which it needs to fill in the same information as for creating a new team, except the team name. Instead it needs to accept that it wants to join this team. Then, when the user submits the form, the same checks get performed. The information will again be stored in the database, but only a new row in the person table will be added, as no new team is created.

Creating an admin account

We have not implemented this functionality yet, for now you need to create admin accounts via the database.

Login

when a user opens the website, it will land on the login page. For logging in the user needs to fill in their email address with their password. The client checks if the email is in the database and if the password corresponds with the password that belongs to the user in the database. If one of both checks fails, an error message will be shown. We have tested these checks thoroughly by inserting invalid input, such as a wrong password, a non existing email address, no input, etc. All of these checks passed. When the checks return true, the user will be logged in to either the admin pages, or the participant pages. If the user is admin or participant is stored inside the database. Also a session key is created for the user, which is a random integer between 1 and 1000000000. This number is stored at the client and in the database. The client receives this information and redirects the user to the correct pages.

Admin can see all existing teams, their members and their information

On the team overview page for the admin, all teams should be shown with their corresponding members. We have chosen to have the tables fold in so that it is easier to navigate through the teams. When clicking on a team, it opens a table with all the members. For every number we show their name, number, email, car and allergies. We have tested this by creating multiple teams with different amounts of users. At first, the result was that only the team captains were shown, but after fixing this bug everything works correctly and all the correct data is shown.

Admin can edit information from users

We have not yet implemented this functionality, only the button for this functionality has been created.

Admin can see all graded and ungraded submissions from participants

From the submissions page, admin will have the options to go to the graded or ungraded submissions. While the pages exist, no functionality has been implemented here yet.

Admin can grade submissions

Since the submission page is not operational yet, this functionality hasn't been implemented.

Admin can see all challenges, puzzles and crazy88

We implemented all three pages for the different challenges the same. On load they fetch the data about the challenges and create a table on the page as an overview. Here the admins can proceed to add new challenges, edit the existing ones or delete one. Both functionalities edit the table on the page and send a request to the server for the required changes at the same time.

Admin can see a scoreboard

On the scoreboard page of the admins all the teams and their scores should be shown. There is a table which shows the place, name and score for every team in descending order. The client fetches the team table via the server to return all teams. These results will be put into an array. This array is sorted such that the highest score is on top. Then the results are put in a table. We have tested the scoreboard by adding different teams with different scores into the database. The result of these tests is positive: the table updated correctly and sorted everything properly. This scoreboard is always visible, even if the admin disables the scoreboard.

Admin can make the scoreboard visible/invisible

On the scoreboard page admins have a button that changes the visibility of the scoreboard for the participants. This fetches a boolean from the server, and sends one back according to the information in the message (If it got true it returns false, if it got false it returns true). However this functionality has not yet been tested, so there might be errors with it.

User can see an overview of their current team members

On the team overview page for participants, all team members should be shown. A table is shown with for all members their name, number, email, car and allergies. The client fetches the database via the server for all users with the same team as them. The result of this fetch is shown in the table. We tested this functionality by creating a new team and adding several members. The result of this test is positive: the table updates correctly and shows all the correct data.

The team captain can delete members from their team

We have not yet implemented this functionality, only the button for this functionality has been created.

User can see all challenges, puzzles and crazy88

We have not yet implemented this functionality, only the pages and the table for the results of this functionality is created

User can hand in a submission for a certain challenge, puzzle or and crazy88

The user will pick a challenge to submit from a table of challenges on the respective page: Challenges, Puzzles, Crazy88. The user will then get redirected to the submission page with the name of their challenge as parameter, so it will be saved in the database as a submitted challenge with the same name, for that specific team(if accepted as a valid submission). After the player presses submit, a page will pop up confirming their submission and offering them to go back to the main page(maybe we could add other page options as well).

User can see an overview of their handed in submissions

We have not yet implemented this functionality, only the space for the results of this functionality is created.

User can see the current state of the scoreboard if visible

On the scoreboard page of the participants all the teams and their scores should be shown. There is a table which shows the place, name and score for every team in descending order. The client fetches the team table via the server to return all teams. These results will be put into an array. This array is sorted such that the highest score is on top. Then the results are put in a table. We have tested the scoreboard by adding different teams with different scores into the database. The result of these tests is positive: the table updated correctly and sorted everything properly.

The scoreboard should not be shown if the admins disabled it. We have implemented some backend functionalities for this already, but when we tested these functionalities it was not working yet. We are still working on fixing this functionality.

User can see all photos of the event in the gallery and download them

We have only partially implemented this user story. The user can see photos which are stored in a folder on the client side. But there is no way to download them or to add photos.