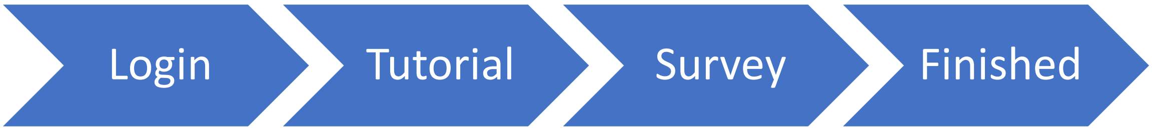
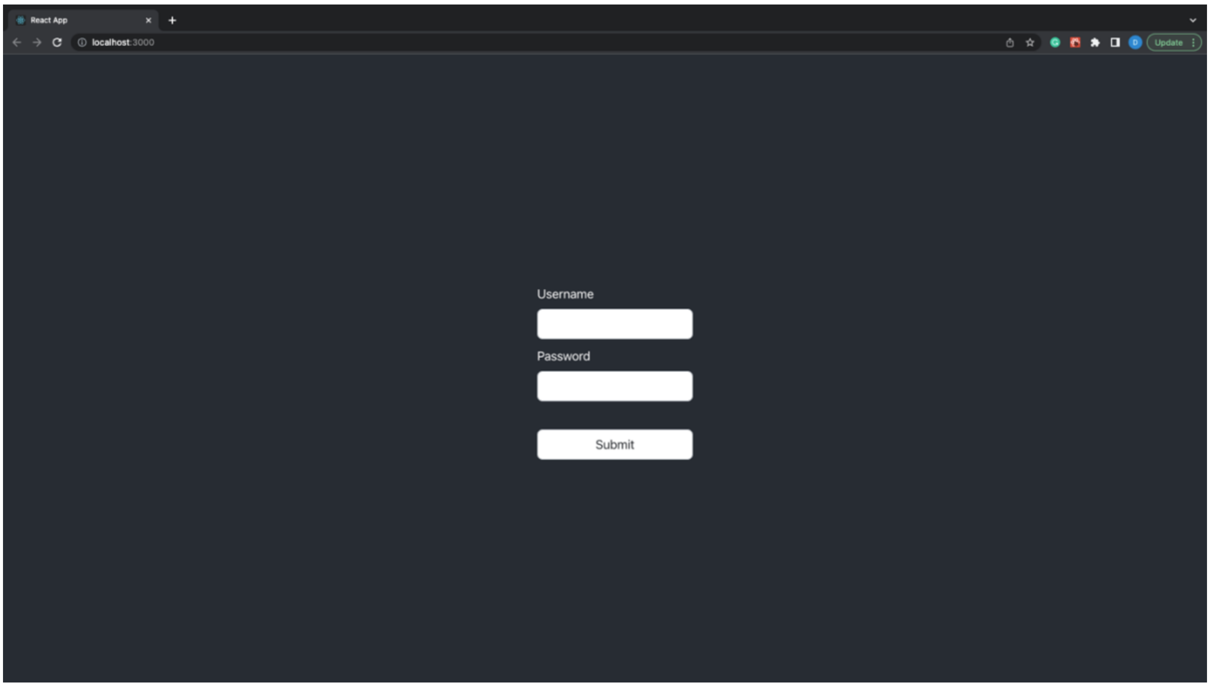
Application

The interface is implemented as a React web application. Hence, it can be accessed through a web browser and the experts do not have to install anything. It contains four sequential screens.


Login
Tutorial
Survey
Finished

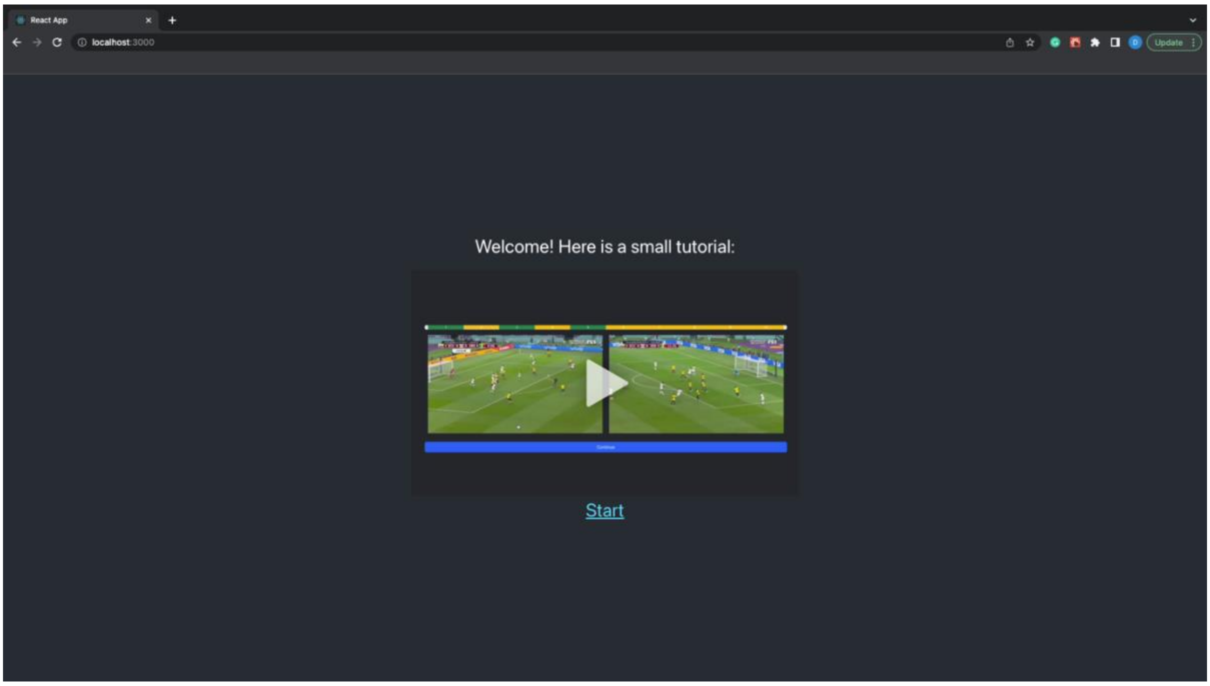
Login screen

We require the user to enter credentials to access the app. This allows us to assign a unique survey to each expert and compute inter-rater agreement metrics. Each expert will be provided with a username and access token when he/she is invited to participate.



Tutorial screen

The tutorial screen consists of a tutorial video and a start button. The start button opens a new tab with the survey.

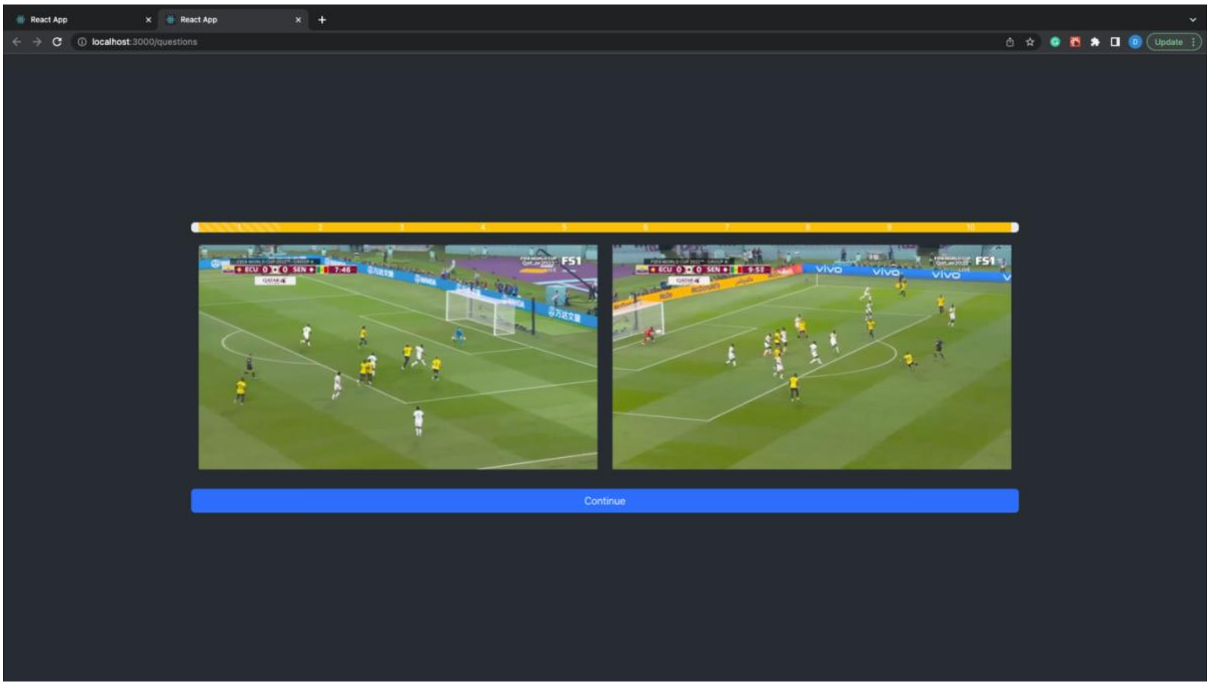


Welcome! Here is a small tutorial:

Start

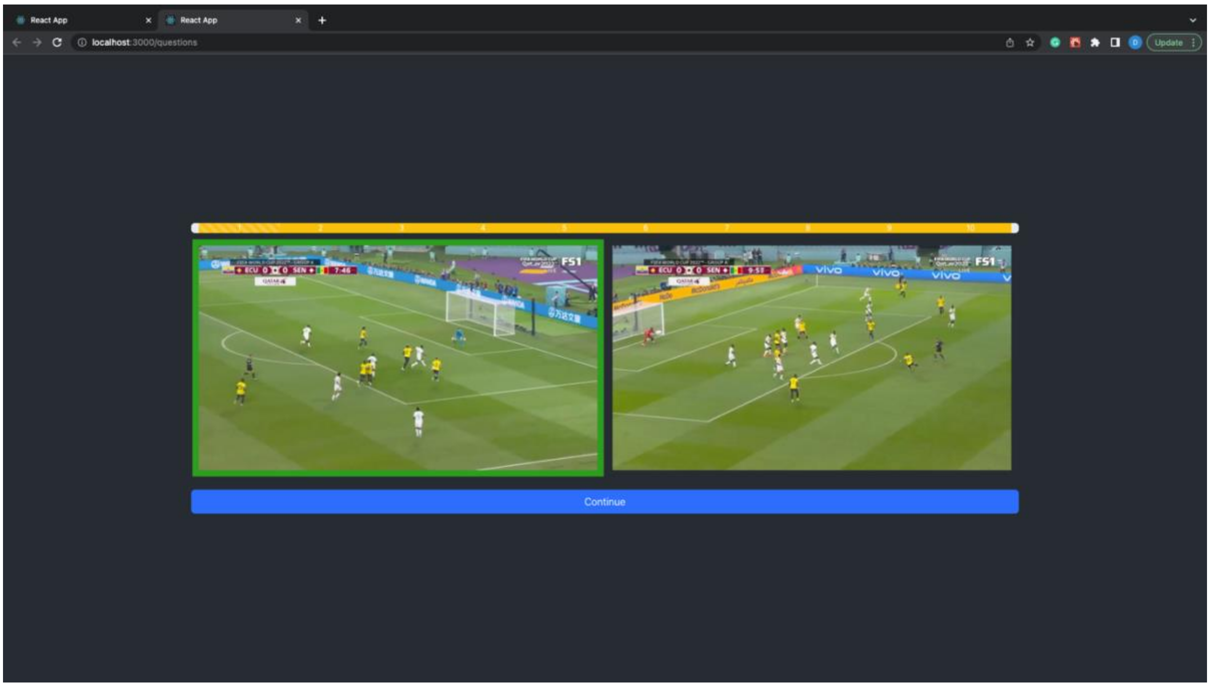
Survey screen

The survey in the prototype consists of 10 questions. This number can be adjusted according to the time available to experts. In each question, the expert is asked to select the shot with the highest scoring probability.

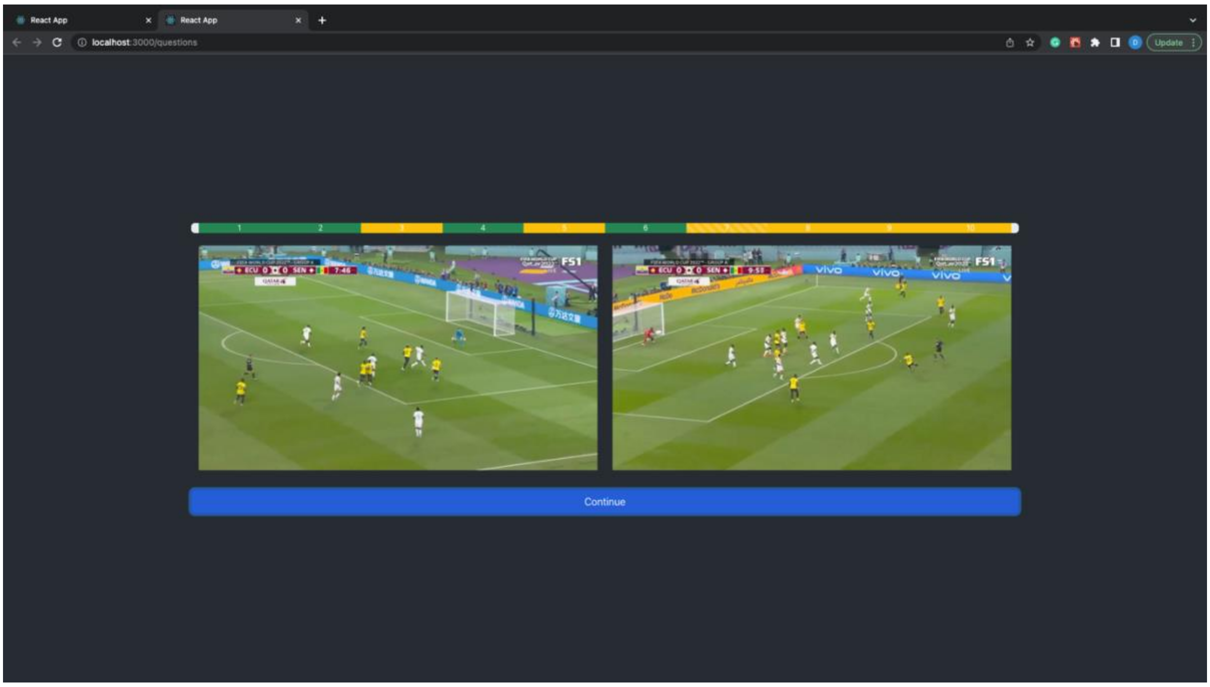


To judge each shooting opportunity, the expert is shown a screenshot taken right before the shot is performed. When the expert hovers the mouse over one of these screenshots a clip plays showing 5 seconds of video that preceded the shot.

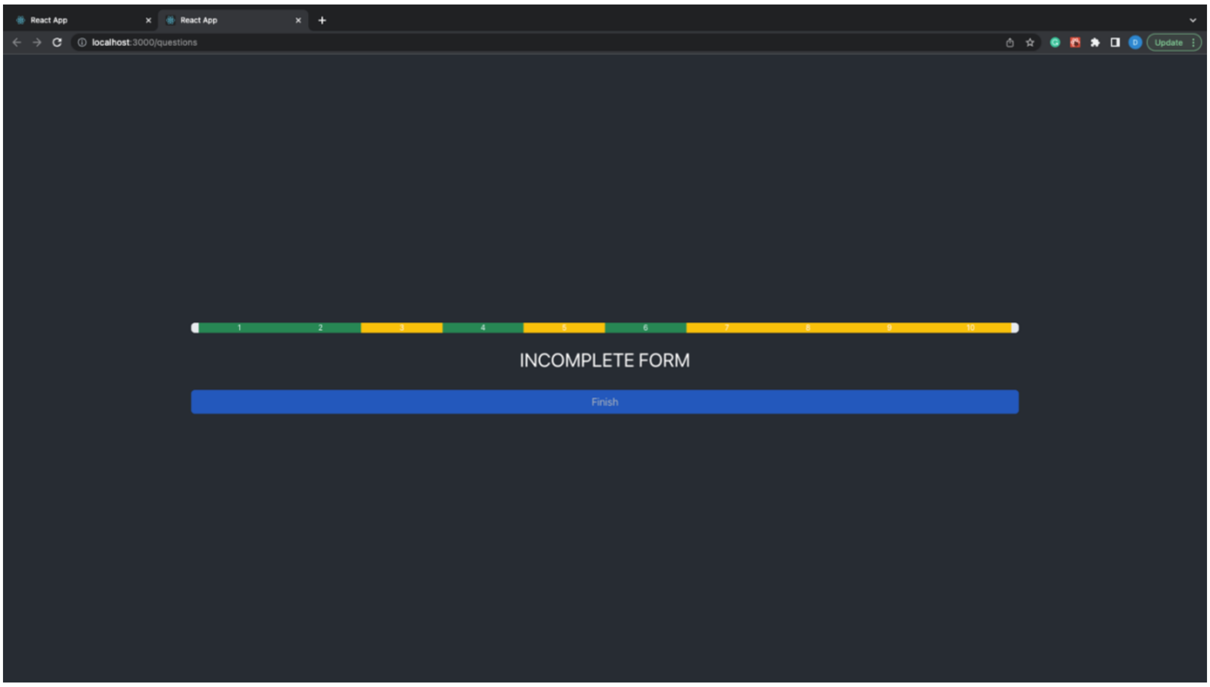
The expert makes his/her choice by clicking on one of the thumbnails. The selected shot is then highlighted by drawing a green frame around it.



The progress bar shows the current question and how many unanswered questions remain. It also enables the expert to navigate to any other question by clicking on a number on the progress bar.



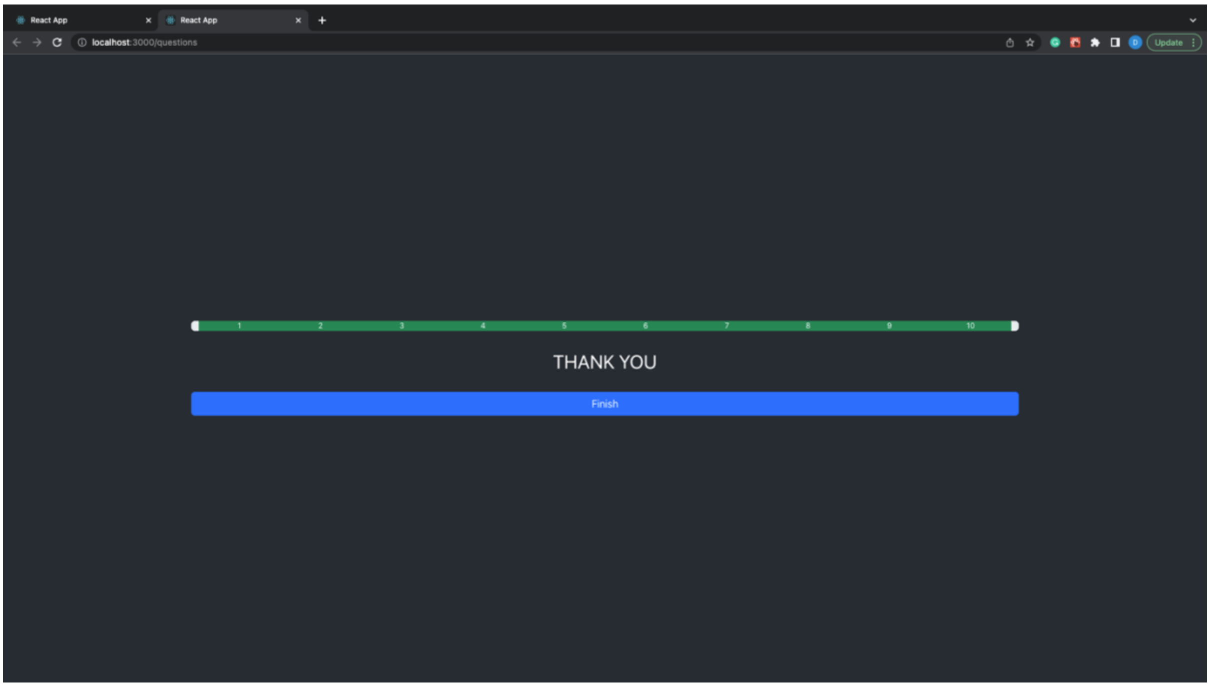
It is possible to skip questions by not answering and clicking continue, however, to end the survey one must answer all questions. The finish button is disabled until all questions have been answered.



INCOMPLETE FORM

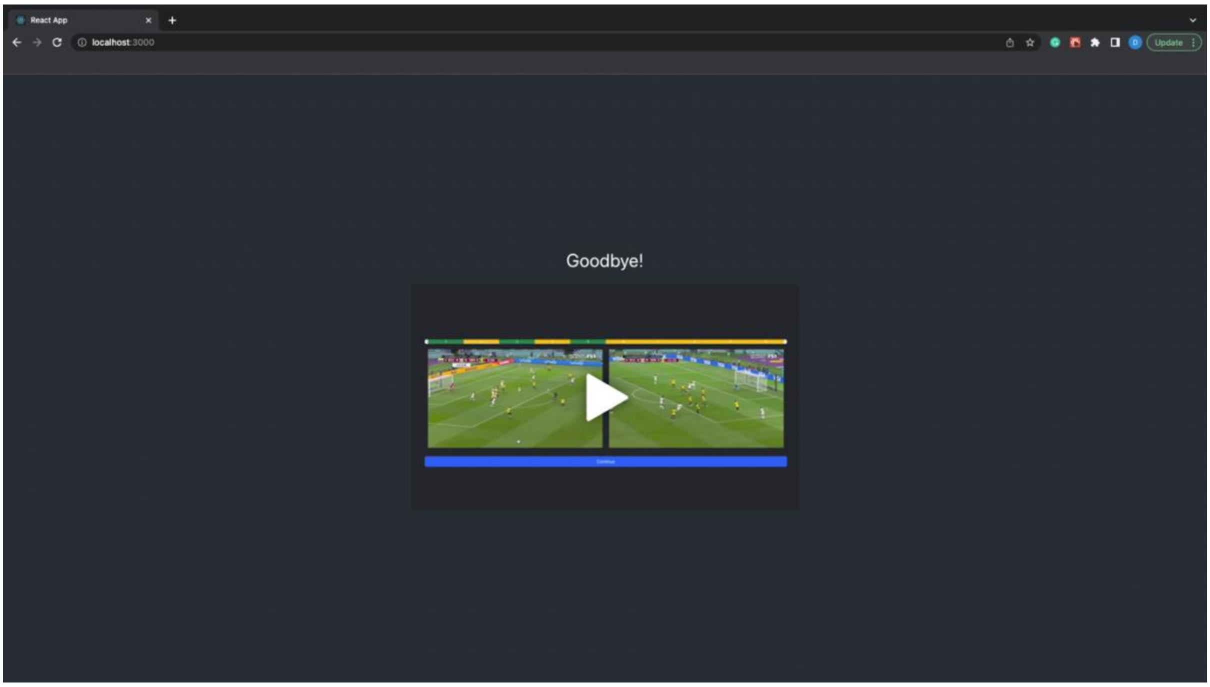
Finished screen

After answering all questions and clicking on the Finish button on the last question, the survey tab closes.



THANK YOU

The start button on the home page disappears and the user is logged out.



Goodbye!

Design Questions:

- **How to select questions:** The selection of questions could be static or dynamic. In the static option, a series of questions will be created for each expert and stored in the database. In the dynamic option, clips will be selected at random, and questions will be created during runtime. For both options, we should make sure to alternate the left/right position of shots when we show them to multiple experts.

- **Duplicate Resolution:** We should decide how many different experts we will show each pair of questions.

- S**election of shot pairs:** How will we sample pairs of shots? We could select pairs with a similar position and the same body part, and then use the judgment of the experts to determine the importance of less obvious contextual factors.

- **Update answers:** Do we allow the experts to update their answers?

- **When to save answers:** Answers can be saved after each question or when finishing the entire series. Saving every answer at the time of answer makes the app more resilient to failures. However, this design needs to resolve the problem of changing records when experts change his/her answer during a session.

- **Number of questions:** Do we require each expert to answer a pre-determined number of questions, or do we let experts rate as many questions as they wish?

- **Source of questions:** Are we going to fetch questions from a known API like Wyscout during runtime or create a database with questions? Another concern connected with this is, where and how are we going to store the videos. This concern is also connected with front-end implementation, which will be resolved after the decision.

- **Local vs session storage:** Storing the user's token after login in local or session storage has a tradeoff. The local storage option makes it possible to continue to survey after closing the tab but makes the app more vulnerable to errors if the expert does end his/her session successfully and the same computer/machine will be used by the next expert. Session storage is more resistant to error, but the expert needs to start over to survey if the tab closes. The current prototype is using local storage.