

TestingSystem – Frontend documentation

This documentation is organized by React components that were used in developing frontend support for TestingSystem project. Each component is described why it was used, followed by explanation of state variable and also binded functions.

Start

This component is first component that we see when we open application. It is used for welcoming visitor and offer to him to login in case visitor has created account prior. If visitor doesn't have account, he can create one.

State:

- Option – Used to determine if visitor clicked login submit button or clicked register button. It is used for navigating on Start component.
- Name – Used for keeping username if logged for sending as props to Login component.

Binded functions:

- loginSuccessCallback()
- loginFailureCallback()
- registerCallBack()
- registeredSuccessfullyCallback()
- registeredCancelled()

All this functions were used for notification and navigating purposes

- logout() – This function is used for deleting token and other user data from session storage.

Login

This component is used as login form, for user to enter his credentials and hit server. It displays necessary fields that to be filled: email, password, and optionally remember me checkbox. First two fields are validated by HTML5.

State:

- Email – Used to keep currently entered email address.
- Password – Used to keep currently entered password.
- RememberMe – Used to keep if checkbox is checked.

Binded functions:

- emailChanged()
- passwordChanged()

- rememberMeChanged()

These functions are used to react on changes on frontend. They all call setState for appropriate field and updates it accordingly.

- submit()

Used for submitting form to server for login. In case it succeeds it returns authorization token.

Register

This component used as registration form for visitor to enter his data and register at server. Visitor can have two types of roles: Student and Examiner. Visitor also need to fill his name and surname, also to provide email and password. All fields are validated through HTML5

State:

- FirstName – Contains currently set first name.
- LastName – Contains currently set last name.
- Email – Contains currently set email address.
- Password – Contains currently set password.
- Type – Contains either Student(2) or Examiner(1) type.

Binded functions:

- firstNameChanged()
- lastNameChanged()
- emailChanged()
- passwordChanged()
- typeChanged()

All this functions were used for updating according fields with setState function.

- submit() – This function is used for sending data to server, which responds with Ok(Status 200) in case everything went right (not existing email for example) or BadRequest(Status 400) if something went wrong

Separation of Examiner and Student use case

Previous components were used for both Examiner and Student, and it ending result was storing token in browser's session storage. From now, examiner will open Home component that reassembles some kind of dashboard where he can use functionalities. Student is redirected to TakeTest component.

TakeTest

This component is used like dashboard for Student where he can see his results for tests he taken, and also a dropdown menu for choosing tests that he didn't take.

State:

- `currentState` – Contains in which stadium of displaying component is. Used for navigation
- `tests` – This variable is placeholder for tests that we will fetch from server.
- `selectedTestId`
- `totalScore`
- `studentScore`

Last three variables were used for notification when student is taken back from exam to this main panel.

Binded functions:

- `selectedChange()` – Sets appropriate value in `currentState` variable, which is used for nav
- `testResults()` – This is callback that is being used for taking back to main panel.

TakeResults

This component is used for fetching and displaying results of tests that Student has taken. This is one of lastly created components, and in differs than others, it's been written using most modern React Hooks API.

TakeQuestions

Most important component in Student part of frontend. It is used to display questions, take answers from student and submit them back to server. Displaying questions have several difficulties like: showing variable number of answer options, use checkbox (for multiple answer options questions) radio button (for yes and no questions) or nothing (for free texted questions that need to be filled).

State:

- `testId` – Used for fetching all questions for corresponding test.
- `questions` – This variable is placeholder fetched questions.

Beside this state variables, one Map was declared for purpose of keeping answered data in map. Map is organized in shape: [*questionId*, [*ans1*, *ans2*,...]]. Keys are `questionId`, and values are arrays of answer options that Student provided. This array will be different in case of question type:

- YesNo – Array will hold only one value, on change it will dispose previous value
- Multiple choice – Array must be checked if it contains that value, if not add it, if yes just filter array
- Free text – This case will be processed like YesNo

Binded functions:

- `answeredYesNo()` – Procedure for taking new response for question of type YesNo
- `answeredMultiple()` – Procedure for taking new response for question of type Multiple
- `answeredFreetext()` – Procedure for taking new response for question of type FreeText

CreateTestForm

This component is responsible for creating new test. Test name, starting date and time should be provided to it, and result should be successfully created test and redirection to Home component (Dashboard) of Examiner.

State:

- testName – Contains user inputted value for test name.
- startDate – Contains user inputted value for starting date.
- startTime – Contains user inputted value for test starting time.

Binded functions:

- testNameChange()
 - startTimeChange()
 - startDateChange()
- All three functions are used for setting appropriate values using setState

CreateGroupForm

This component is responsible for creating new group.

State:

- groupName – Contains user inputted value for test name
- Binded functions:

Binded functions

- groupNameChange() – Used for setting newly entered name
- submit() – Used for sending form to server

TestList

This component for displaying list of created tests by Examiner.

State:

- examinerId – Contains examiner id that is logged in.
- examinerName – Contains examiner username from session storage.
- allTest – Placeholder for fetching tests.
- showQuestionId – Callback hook for dropdown menu
- showQuestionName – Callback hook for sending as props.
- changeStartDateTestId – Callback hook for dropdown menu
- changeStartDateTestTitle – Callback hook for sending as props.

Binded functions

- activateOrDeactivateButton() – callback for activating test
- listAllQuestionsButton() – callback for displaying all question in this test
- changeStartDateButton() – callback for displaying change start date component

Test

This component is responsible for showing Test and its details.

State:

- testId
- testTitle
- testExaminerId
- testStartDateTime
- testStartTime
- testIsActive
- testScore

All fields are for holding Test parameters.

Binded functions

This component is for displaying purposes only, it doesn't have any interactivity.

TableHeader

This component is for displaying purposes only. It displays header of table that contains all tests.

EditTestStartDateTime

This component is responsible for changing test start date and time.

State:

- testId – Contains test ID for hitting server with new data.
- testTitle – Contains test title for displaying.
- startDate – New date
- startTime – New time

Binded functions

- startDateChange() – callback for setState on startDate state variable
- startTimeChange() – callback for setState on startTime state variable
- submit() – Used for sending form to server

Student

This component represents one row in StudentList table.

State:

- id – Contains Student ID
- firstName – Contains Student first name
- lastName – Contains Student last name.
- checked – Contains checkbox if student will be added to group.

Binded functions

- checkChange() – callback for setState on checked state variable

StudentList

This component represents table with Students.

State:

- students – Contains all students that were fetched.
- studentsAddList – Contains all students that were selected (checked)
- groupId – Contains groupId for sending selected students.
- groupId – Contains group title for displaying.

Binded functions

- finishClick() – callback for sending selected students to server to add to group
- getChecked() – function for getting if checkbox is checked

TableHeader (QuestionList)

This component is used for purpose of displaying table header for question list.

Question

This component represents one row in QuestionList.

State:

- questionId – Contains question ID for displaying.
- questionText – Contains question text for displaying.

QuestionList

This component represent table with all questions in one test.

State:

- testId – Contains test ID for displaying and listing questions.
- testTitle – Contains test title for displaying.
- allQuestions – Placeholder for fetching all questions for test with given ID.
- editQuestionId – In case this is not null, we should render edit question comp.
- deleteQuestionId – In case this is not null, we should render delete question comp.
- addQuestion – In case this is not null, we should render add new question comp.

Binded functions

- editClicked() – Callback for editing question
- deleteClicked() – Callback for deleting question
- addClicked() - Callback for add new question
- createClicked() – Callback for create new question component

DeleteQuestion

This component is confirm form for deleting question

State:

- questionId – Contains question ID that should be deleted.
- testId – Contains test ID that contains ID

All this state were sent as props.

Binded functions:

- delete() – Confirmed deleted and hitting server

CreateStatistics

This component is used for general creating all statistic use cases.

State:

- tests – Placeholder for all tests that we fetch.
- groups – Placeholder for all groups that we fetch.
- selectedTestId – Value of selected test from dropdown
- selectedGroupId – Value of selected group from dropdown

Binded functions:

- selectedTestIdChange()
- selectedGroupIdChange()

These functions are used for updating states.

Statistics

This component is main component for showing all created statistics for selected test and group.

State:

- tests – Placeholder for all tests that we fetch.
- groups – Placeholder for all groups that we fetch.
- selectedTestId – Value of selected test from dropdown
- selectedGroupId – Value of selected group from dropdown

Binded functions:

- selectedTestIdChange()
- selectedGroupIdChange()

These functions are used for updating states.

StatisticsDetail

This component is responsible for handling user inputs for creating statistics. It has three options: create statistics now, schedule for date and schedule monthly statistics. It's also responsible for hitting database with inputted data.

State:

- testId – Parent component send Id as props
- groupId – Parent component send Id as props
- date – Entered date in case of scheduling for date
- changed – Indicator if state is changed

Binded functions:

- createNow() – Function that executes if user click create statistic now
- scheduleDate() – Function that executes if user click schedule statistic for date
- scheduleMonthly() – Function that executes if user click schedule monthly.
- changedDate() – Calls setState for date state variable.

StatisticsTable

Component used for displaying statistic that were created in database. They are represented in form of rows in a table, and this component contains all that rows. After fetching, they are displayed. This component is created using Hooks.

Group

This component presents one row in GroupLayoutComponent. It is used for displaying groups alongside button for selecting group in parent component.

State:

- groupId – Represent group ID for holding as value in button
- groupTitle – Represent group title for displaying.

Bindend functions doesn't exist in this component.

GroupList

This group contains multiple Group components and is used as list of groups for students to add in.

State:

- examinerID – Contains currently logged in examiner ID
- allGroups – Placeholder for fetching data from server
- listStudentsGroupId – Selected group id from Group component
- listStudentsGroupName – Name of selected group

Binded function:

- listAllStudentsButton() – This functions is called when group is selected from Group component.

CreateQuestion

This is main component for creating new question. It contains question text as generic field for all question types. There is dropdown list to choose which question type user want to create, and creates adequate form bellow dropdown.

State:

- testID – ID of test where to create new question.
- answerType – Contains which type of question user is creating.
- questionText – Text of question placeholder.

Binded functions:

- questionTextChanged() – setState function for questionText variable
- answerTypeChanged() – setState function for answerType variable
- getAnswerText() – hook to child component to get inputted text
- setCorrectAnswer() – hook to child component to connect with checked answer option

MultipleAnswers

This component represents variable multiple-choice questions. It offers placeholders for entering answer options, checking them if they are correct or not, and to delete sufficient placeholders if needed.

State:

- options – Array for holding all answer options
- optionNumber – Length of options array.

Binded functions:

- delete() – Callback for deleting answer option
- correct() – Callback for setting correct answer option
- text() – Callback for registering change in answer option text
- addNewOption() – Callback for creating placeholder for new answer option

Answer

This component presents Yes/No question answer option.

State:

- value – Value of input field for answer option text
- logo – If answer option should have radiobutton or checkbox
- isCorrect – Used for setting correct value on answer option
- orderNumber – [Deprecated] Used for ordering answer options

Binded functions:

- ValueChanged() – setState for value state variable
- CorrectChanged() – setState for correct state variable

AnswerWithDelete

This component is used to display and fill Multiple choice questions. It has same state and binded functions as Answer, with different implementation of binded functions.