**Quiz Application**

As part of your volunteering work to improve the quality of education in the country, you are tasked to implement an application that can be used to effectively create and carry out online quizzes.

The application has two main parts:

* **Teacher’s dashboard** – here, teachers can create categories, quizzes, and check students' performance
* **Student's dashboard** – students solve quizzes and view leaderboard.

**Public Part**

The **public part** of your projects should be **visible without authentication**.

* **Landing Page (must)** – contains two forms – login and register
  + **Login form (must)** – redirects to the teacher’s or student’s dashboard, depending on role. On fail, show a message and stay in landing page.
    - Requires **username** and **password**
  + **Register form (must)** – registers a user as student and redirects to student’s dashboard. On fail, show a message and stay in landing page.
    - Requires **username**, **firstname**, **lastname** and **password**

**Student’s Part**

* **Dashboard page (must) –** contains all quiz **categories**, **leaderboard**, and **quiz history.** Feel free to display information in any layout that you find appropriate
  + **Categories** – display all available categories **(must)**. Each category should be a link that leads to a dedicated category page. **(must)**
  + **Leaderboard** – show top ten students by score **(must)**. Provide a link to a dedicated leaderboard page **(should)**
  + **Quiz history** – display the last five quizzes completed by the student. **(must)** Provide a link to a dedicated history page **(should)**
* **Categories page (must)**
  + Show all quizzes that belong to the selected category. **(must)**
  + Quizzes that are solved should be at the bottom. **(must)**
  + Clicking a solvable quiz should display a confirmation that a student has only one attempt. On yes, navigate to the **Solve quiz page** with the selected quiz. **(must)**
  + If there are too many quizzes, consider pagination. **(could)**
* **Solve quiz page (must) –** attempt to solve a quiz.
  + Only one quiz can be active at a time. **(must)**
  + A quiz can be submitted only once. **(must)**
  + After submitting, display a message with the calculated score and navigate to the dashboard. **(must)**
  + Consider adding a timer, which automatically submits the quiz when expired. **(could)**
* **Leaderboard page (should)**
  + Display all students with their points in descending order. **(should)**
  + If there are too many students, consider pagination **(could)**
  + Consider adding a filter field, that enables searching a student by username **(could)**
* **History page (should)**
  + Display all quizzes solved by the student.
  + If there are too many quizzes, consider pagination **(could)**
  + Consider adding a filter field, that enables searching a quiz by name **(could)**

**Teacher’s part**

* **Dashboard page (must) –** contains a **form to create a category**, **categories** section**,** navigation to the **Create a quiz page** and **displays quizzes created** by the logged in teacher
  + **Create a category form (must)** – create a new category.
    - Contains a text field and a button.
    - Categories have a **required name. (min 2** symbols**, unique** across categories**)**
    - Display message on success/fail
  + **Create a quiz page link** **(must)** – navigate to the create a quiz page. If there are no categories, the link is inactive and a message that a category must be created first should be displayed.
  + **Categories section (must) –** display all categories by all teachers. Clicking a category navigates to the selected category page
  + **Created quizzes section** **(should)** – display the last 5 quizzes create by the teacher
* **Create a quiz page**
  + This page must not be accessible if there are no categories. (even by manually typing in the URL) **(must)**
  + A quiz has a required **name, category** and a collection of questions, time limit
    - Name is **unique** across quizzes, **min** 5 symbols
    - Category is selected from the available categories from all teachers
    - A quiz should have **at least two** questions
    - Time limit (optional). After the time is up, the quiz is automatically submitted. **(could)**
  + Questions – **text**, **type**, **points** and **answers (must)**
    - A question has a **type** – multiple or single choice
    - A question has a **text** – the question itself
    - A question awards points in the interval **[1, 6]**
    - Single choice **(must)** – add at least two possible answers, **only one** should be marked as correct.
    - Multiple choice **(should)** – add at least two possible answers, **at least one** should be marked as correct.
  + The page must have two buttons – **Add a single-choice (must)** and **Add a multiple-choice. (should)**
    - Clicking any of those generates a form for the respective type.
    - The form contains fields for **text**, **points** and **two answers**
    - There is a button **Add another answer** that dynamically adds a field for the question
* **Category page (must)**
  + Show all quizzes that belong to the selected category.
  + Each quiz has two buttons: **Solve** and **View (should)**
  + Teachers can solve each quiz **as many times** as they want. They see a score on submit, but their score is not added to the leaderboard **(should)**
  + Clicking the view button navigates to the View Quiz page **(should)**
* **View Quiz page (should)**
  + Display all students who took the quiz – **username**, **firstname + lastname**, **date** and **score**

**Scoring (must)**

A submitted quiz must be processed, and the points calculated. The total score is the sum of all correct questions' points.

* **Example** – a quiz contains 10 questions.
  + Questions 1-5 award 3 points.
  + Questions 6-10 award 6 points.
  + Student 1 answers all questions correctly – total score is 5 \* 3 + 5 \* 6 = 45 points.
  + Student 2 has q2, q6 and q10 wrong – total score is 4 \* 3 + 3 \* 6 = 30 points
* Multiple choice scoring – no partial points. Either full points or zero.
  + **Example** – a question has 4 options (A, B, C, D) and options (C, D) are correct. Question is worth 4 points.
  + Student 1 selects C and D => receives **4** points
  + Student 2 selects only D or only C => receives **0** points
  + Student 3 selects D and C, but also A or B => receives **0** points

**Import/Export (should)**

Teachers should have an **import/export quiz** functionality available at their dashboard. The imported/exported files can be in json/xml/yaml or any other format you find suitable.

* Importing a quiz must provide all the data necessary to setup a quiz with valid name, category, list of questions, etc.
* Exporting a quiz should have two options:
  + Export **quiz structure –** exports name, category, list of questions, etc.
  + Export **quiz data –** exports students’ scores for the selected quiz

**General Requirements**

Your Web application should use the following technologies, frameworks and development techniques:

* Use **ReactJS, Express, MariaDB** and **preferably Visual Studio Code**
* Create **usable** user interface.
* All the data should be loaded from a web server using **services**
* Apply proper **data validation** and **error handling**
* Your project should pass the default **ESLint linting** configuration without any errors
* Use Git as source control
* **Documentation** of the project and project architecture (as .md file, including screenshots)

**Optional Requirements**

* Take advantage of Git **branches** for writing your features.
* Originality of the implementation (uniqueness)
* CI / CD
* Host your application in the web (any public hosting provider of your choice)
* Unit test **a few components**
* Develop a **standalone mobile client**, using React Native, Ionic, Apache Cordova or any other similar technology

# Trainer Project Defenses

Each **student** must make a trainer defense of its work to the trainers (~15 minutes) It includes:

* Explain application structure, source code and design decisions
* Show the commit logs in the source control repository to prove a contribution from all team members