

Universitatea Tehnica a Moldovei

Departament ISA

Disciplina:
Tehnologii Web și aplicații

RAPORT

Lucrare de laborator nr. 1

A efectuat:

studentul gr. TI-171M, Lealin Serghei

A verificat:

Vladimir Poddukin

Chișinău 2018

Task:

Create a question-based quiz using HTML and JavaScript.

List of specifications:

- List of question are predefined in source code of the project;
- Questions and answers are showed in English;
- User choose, how many questions he wants to answer;
- Questions for the user are selected randomly from the pool of questions;
- When the user press “Start”, the timer starts to count seconds;
- User can choose only one answer from the list;
- All answers must be answered;
- Only after pressing “Answer”, user gets his final result in form “right number of answers from the selected number of questions”, and also user sees, how much seconds passed from the start of the quiz.

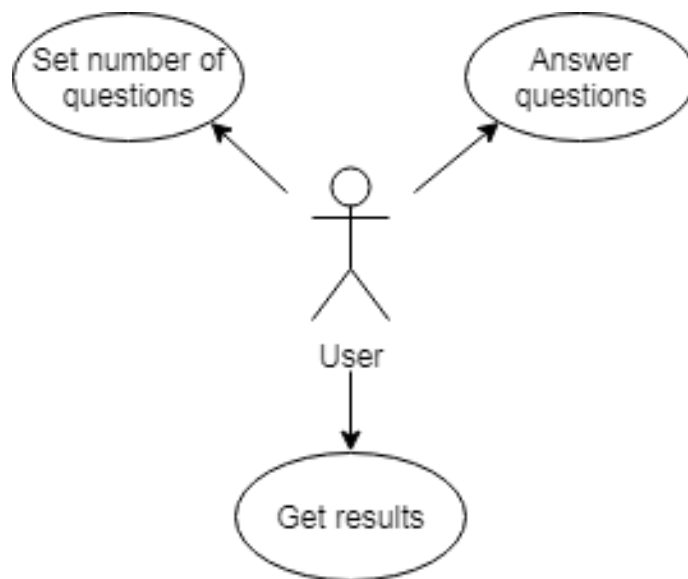
Use case diagram:

Image 1 – Use case for user

As shown on diagram (Image 1), user doesn't have so many options to do in the project, but the main idea of quiz is answering question as fast as possible. User can get his final results, only if all questions were answered. The number of questions can vary from one to all possible questions from the pool, user decides himself, how difficult he wants his quiz.

Source code:

Main code of JavaScript, which Node.js compile:

```
function getQuizScript(){
    var scriptText = fs.readFileSync("./quiz.js");
    var questions = fs.readFileSync("./questions.json");

    scriptText = scriptText.toString().replace("var jsonQuestions;", "var jsonQuestions = " + questions);

    return scriptText;
}

http.createServer(function (req, res) {
    if (req.url == "/quiz.js")
    {
        res.writeHead(200, {'Content-Type': 'text/javascript'});
        res.write(getQuizScript());
    }
    else
    {
        res.writeHead(200, {'Content-Type': 'text/html'});
        res.write(fs.readFileSync("./index.html"));
    }

    res.end();
}).listen(8080);
```

The first function gets raw JavaScript code and then loads from file questions in JSON format. After loading all files, one string from raw code is replaced by question from JSON file.

Second function is main for Node.js and it's necessary to receive and respond via http packets, acting as a server for this project.

Generation random numbers of questions:

```
var elem;
for (i = 0; i < QUESTIONS_SHOW_COUNT; i++){
    do{
        elem = Math.floor((Math.random() * QUESTIONS_COUNT));
    }
    while (questionIDs.indexOf(elem) != -1);

    questionIDs.push(elem);
}
```

Creating and appending elements dynamically to HTML page:

```
sQuestionCount = document.createElement("SELECT");

for (i = 0; i < QUESTIONS_COUNT; i++){
    var opt = document.createElement("OPTION");
```

```
        opt.value = i+1;
        opt.innerHTML = i+1;
        sQuestionCount.appendChild(opt);
    }

    document.body.appendChild(sQuestionCount);
```

This is an example code of creation and appending elements to main web page. In this example, there is a “Select” object, which contains sub objects: “Option”. The sub objects also created dynamically and append to the first object, after that the first object appends to document of web page.

Conclusion

In this project were studied bases of JavaScript and HTML. In JavaScript were used object like: basic variables, functions, arrays and objects. Also were studied communication between HTML and JavaScript: adding objects, getting objects by theirs ID, hiding and showing needed elements. For this project was used a well-known library: “Node.js”, this library provides a server-driven JavaScript and was used for loading files from the Server. This project is using a common-known repository: “git”, for proper using this repository, were studied main commands: copy, get, add, commit and push.