

MINISTERUL EDUCAȚIEI AL REPUBLICII MOLDOVA  
UNIVERSITATEA TEHNICĂ A MOLDOVEI  
FACULTATEA CALCULATOARE, INFORMATICĂ  
ȘI MICROELECTRONICĂ  
CATEDRA AUTOMATICĂ ȘI TEHNOLOGII INFORMAȚIONALE

# Raport

la disciplina Tehnologii Web

Tema: JavaScript, quiz aplicație

Lucrare de laborator 1

A efectuat: st. gr. TI-171M  
Vovc Artemie  
A verificat: lect. super.  
Vladimir Poddukin

Chișinău 2018

## Cerințele

1. Creați o structură de mape pentru a grupa fișierele după tip.
2. Adăugați fișierul de bază index.html cu elementele standard și în body să fie adăugate unele elemente în mod dinamic.
  - 2.1 Utilizatorul trebuie să vadă pe prima pagină o întrebare cu 4 variante de răspunsuri
  - 2.2 Fiecare răspuns trebuie să aibă în față câte un checkbox prin intermediu căruia se va alege răspunsul utilizatorului
  - 2.3 Mai jos de variante de răspunsuri vor fi două butoane. Primul buton va avea funcționalitatea de next (ignoră întrebarea și treci la următoarea), al doilea buton va salva răspunsul ales de utilizator și se va trece la următoare întrebare.
3. Adăugați stiluri, schimbați culoarea fonului în albă iar textul în neagră, importați fișierul de stiluri în fișierul de bază prin link element.
4. Creați un fișier în care se va stoca o mulțime de întrebări fiecare să aibă minimum câte 4 variante de răspunsuri, importați fișierul în index.html.
5. Adăugați funcționalitatea de calcul a scorului.
6. Adăugați funcționalitatea de limitare în timp.
7. Toată logica de adăugare în mod dinamic a elementelor și funcționalitățile descrise anterior se vor stoca într-un fișier main.js care se va încărca în mod asincron și la sfârșitul încărcării fișierului se va invoca o funcție de inițializare a elementelor dinamice.

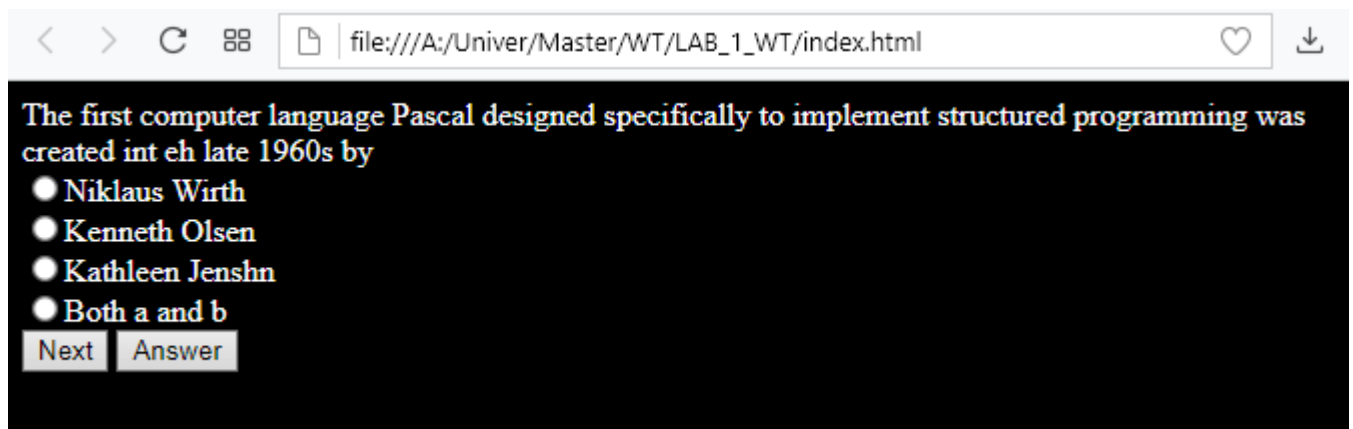


Figura 1 Interfața principală

În figura 1 este reprezentată interfața principală a aplicației, se observă câteva elemente UI ca checkbox-uri, butoane și text.

Butonul next are funcționalitatea de a ignora răspunsul curent al utilizatorului și de a trece la întrebarea următoare.

Butonul answer salvează răspunsul utilizatorului și trece la următoarea întrebare.

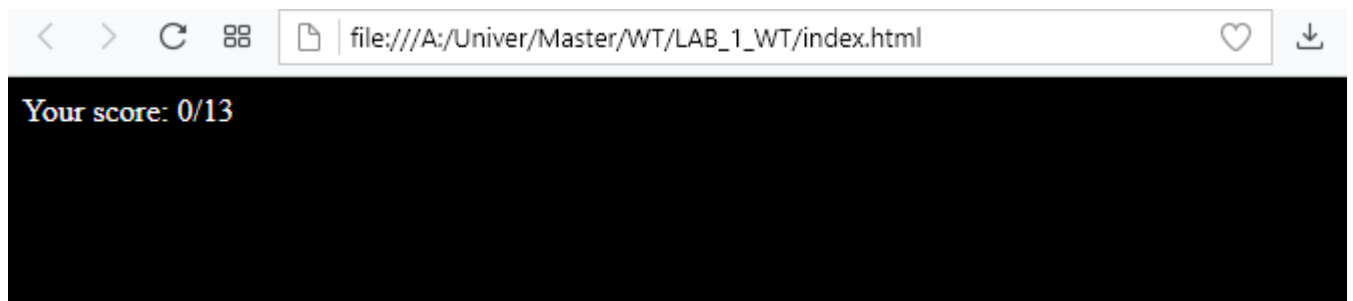


Figura 2 Reprezentarea scorului

La sfârșitul timpului acordat se va reprezenta scorul acumulat de utilizator, pe aceeași pagină.

### Descrierea structurilor de date

Vezi [https://github.com/artvovc/Lab\\_1\\_WT](https://github.com/artvovc/Lab_1_WT) toate următoarele explicații vor avea în cont fișierul main.js.

UIController reprezintă o mapă de obiecte care reprezintă elementele UI, Elementele UI din aceasta mapă pot fi acelea fixate în index.html dar și celea generate dinamic în funcția loadContentWithFollowData. Scopul este de a oferi o posibilitate de acces din oriunde a elementelor UI.

currentItemState este un obiect în care se memorează un element din structura de date "data" din fișierul data.js. Scopul acestei variabile este de a salva elementul curent din șir de întrebări pentru a-l analiza în momentul execuției funcționalității de answer.

userAnswersAccumulator obiectul principal care acumulează răspunsurile corecte la întrebări și răspunsurile utilizatorului. Scopul este de a strânge toate răspunsurile utilizatorului și de a le compara cu răspunsuri la întrebări în calitate de rezultat vom avea scorul.

## Codul sursă

```
"use strict"
```

```
const requiredUIElementForQuestion = "question"
const requiredUIElementForQuiz    = "quiz"
const requiredUIElementForAnswers = "answer"
const requiredUIElementForResult  = "result"
const requiredUIElementForAnswerBtn = "answerBtn"
const requiredUIElementForNextBtn  = "nextBtn"
```

```
const labelSuffix      = ":label"
const divSuffix         = ":div"
```

```
var UIController      = {}
var currentItemState   = {}
var userAnswersAccumulator = {
    quizCount: 0,
    answers: [],
    calculate: function() {
```

```
        return this.answers.filter(it => it.correctIs == it.userAnswerIs).length + "/" +
        this.quizCount
    }
}
```

```
var chron
```

```
function initTimeout() {
    setTimeout(function(){ reloadContent() }, 20000)
```

```
}
```

```
function load() {
```

```
    //populate UIController with required document elements for future processing
```

```
    UIController[requiredUIElementForQuestion] = document.getElementById("question")
```

```
    UIController[requiredUIElementForQuiz] = document.getElementById("quiz")
```

```
    UIController[requiredUIElementForResult] = document.getElementById("result")
```

```
    UIController[requiredUIElementForAnswerBtn] =
```

```
document.getElementById("answerBtn")
```

```
    UIController[requiredUIElementForNextBtn] = document.getElementById("nextBtn")
```

```
    //initiate content
```

```
    reloadContent()
```

```
}
```

```
function reloadContent() {
```

```
    //it should update all required elements with next content
```

```
    if (data != null && data.length != 0) {
```

```
        loadContentWithFollowData(data.shift())
```

```
        if (chron != undefined)
```

```
            clearTimeout(chron)
```

```
    else
```

```
        chron = initTimeout()
```

```
    }
```

```
    else
```

```
        clearRequiredUIElementsAndStopTheGame()
```

```
}
```

```
function loadContentWithFollowData(data) {
```

```
    //it should overlap current data into required UI elements
```

```
    UIController[requiredUIElementForQuestion].textContent = data.question
```

```
    data.cases.forEach(function(it, index) {
```

```

        let radiolIdentificator = requiredUIElementForAnswers + index

        if(document.getElementById(radiolIdentificator) == null) {

            let newDiv = document.createElement("div")
            newDiv.setAttribute("id", radiolIdentificator + divSuffix)

            let newRadio = document.createElement("input")
            newRadio.setAttribute("id", radiolIdentificator)
            newRadio.setAttribute("type", "radio")
            newRadio.setAttribute("name", "possibleAnswer")
            newRadio.setAttribute("value", it.answer)
            newRadio.checked = false

            let newLable = document.createElement("label")
            newLable.setAttribute("id", radiolIdentificator + labelSuffix)
            newLable.setAttribute("for", radiolIdentificator)
            newLable.textContent = it.answer

            newDiv.appendChild(newRadio)
            newDiv.appendChild(newLable)

            UIController[requiredUIElementForQuiz].appendChild(newDiv)
            UIController[radiolIdentificator] = document.getElementById(radiolIdentificator)
            UIController[radiolIdentificator + labelSuffix] =
document.getElementById(radiolIdentificator + labelSuffix)
            UIController[radiolIdentificator + divSuffix] =
document.getElementById(radiolIdentificator + divSuffix)

        } else {

            UIController[radiolIdentificator].checked = false
            UIController[radiolIdentificator].value = it.answer
            UIController[radiolIdentificator + labelSuffix].textContent = it.answer

        }

    })

```

```

userAnswersAccumulator.quizCount += 1
    //set current state
    currentItemState = data
}

function setUserAnswer() {

    function rec(iter) {

        if (data == UIController[requiredUIElementForQuiz].childNodes.length) return
        if (UIController[requiredUIElementForQuiz].children[iter].children[0].checked) {
            userAnswersAccumulator.answers.push({
                correctIs: currentItemState.correctIs,
                userAnswerIs:
UIController[requiredUIElementForQuiz].children[iter].children[0].value
            })
                return
        }
        rec(++iter)

    }

    rec(0)

    reloadContent()

}

function clearRequiredUIElementsAndStopTheGame() {

    //hide elements
    UIController[requiredUIElementForQuestion].style.display = "none"
    UIController[requiredUIElementForQuiz].style.display = "none"
    UIController[requiredUIElementForAnswerBtn].style.display = "none"
    UIController[requiredUIElementForNextBtn].style.display = "none"

    //show result

```

```

    UIController[requiredUIElementForResult].textContent    = "Your score: " +
userAnswersAccumulator.calculate()

}

```

```

<!DOCTYPE html>
<html>
<head>
    <title>lab1wtQuizTest</title>
    <script type="text/javascript" src="scripts/data.js"></script>
    <script async type="text/javascript" src="scripts/main.js" onload="load()"></script>
    <link rel="stylesheet" type="text/css" href="stylesheets/styles.css">
</head>
<body>
    <div id="question"></div>
    <div id="quiz"></div>
    <div id="result"></div>
    <button id="nextBtn" onclick="reloadContent()">Next</button>
    <button id="answerBtn" onclick="setUserAnswer()">Answer</button>
</body>
</html>

```

"use strict"

```

var data =
[
{
    question: "All computers execute",
    cases: [
        { answer: "BASIC programs" },
        { answer: "COBOL programs" },
        { answer: "machine language programs" },
        { answer: "FORTRAN programs" }
    ],
    correctIs: "machine language programs"
},

```



```

{
  question: "Which of the following is not a characteristic of COBOL",
  cases: [
    { answer: "it is a very standardized language" },
    { answer: "it is very efficient in terms of coding and execution" },
    { answer: "it is a very readable language" },
    { answer: "all of the above are characteristics" }
  ],
  correctIs: "it is very efficient in terms of coding and execution"
},
{
  question: "Which of the following BASIC statements is correct",
  cases: [
    { answer: "INPUT N$" },
    { answer: "60 LET C+D=Q" },
    { answer: "45 PRINT X,Y;Z" },
    { answer: "100 END,JOB" }
  ],
  correctIs: "45 PRINT X,Y;Z"
},
{
  question: "Which of the following relates to machine language",
  cases: [
    { answer: "difficult to learn" },
    { answer: "machine-dependent" },
    { answer: "instruction and data are represented by binary digits" },
    { answer: "all of the above" }
  ],
  correctIs: "all of the above"
},
{
  question: "A computer program consists of",
  cases: [
    { answer: "a completed flowchart" },
    { answer: "algorithms" },
    { answer: "algorithms written in computer`s language" },
    { answer: "discrete logical steps" }
  ],
  correctIs: "algorithms written in computer`s language"
}

```

```

},
{
    question: "Which of the following language was developed first",
    cases: [
        { answer: "PASCAL" },
        { answer: "COBOL" },
        { answer: "JAVA" },
        { answer: "BASIC" }
    ],
    correctIs: "COBOL"
},
{
    question: "The language which combines some of the basic features of the three
languages COBOL, FORTRAN and BASIC is called",
    cases: [
        { answer: "LOGO" },
        { answer: "PILOT" },
        { answer: "PASCAL" },
        { answer: "PL/1" }
    ],
    correctIs: "PL/1"
},
{
    question: "Which data base language is likely to become approved both by ANSI and
ISO?",
    cases: [
        { answer: "C++" },
        { answer: "OOP language" },
        { answer: "SQL 1" },
        { answer: "Eiffel" }
    ],
    correctIs: "SQL 1"
},
{
    question: "Which programming language was used for writing the popular operating
system UNIX?",
    cases: [
        { answer: "C++" },
        { answer: "SNOBOL" },

```

```

        { answer: "PASCAL" },
        { answer: "C" }
    ],
    correctIs: "C"
},
{
    question: "When did the high level programming language PASCAL become an ANSI
standard language?",
    cases: [
        { answer: "1983" },
        { answer: "1980" },
        { answer: "1985" },
        { answer: "1987" }
    ],
    correctIs: "1983"
},
{
    question: "The first computer language Pascal designed specifically to implement
structured programming was created in the late 1960s by",
    cases: [
        { answer: "Niklaus Wirth" },
        { answer: "Kenneth Olsen" },
        { answer: "Kathleen Jensen" },
        { answer: "Both a and b" }
    ],
    correctIs: "Both a and b"
},
{
    question: "Who is called the `mother` of COBOL",
    cases: [
        { answer: "Grace Murry Hopper" },
        { answer: "Ada Augusta Byron" },
        { answer: "Masatoshi Shima" },
        { answer: "Kathleen Jensen" }
    ],
    correctIs: "Grace Murray Hopper"
},
{
    question: "Compilers and Interpreters are themselves",

```

```
cases: [  
  { answer: "programs" },  
  { answer: "high level languages" },  
  { answer: "codes" },  
  { answer: "mnemonics" }  
],  
correctIs: "programs"  
}  
]
```