**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

*з лабораторної роботи № 12*

**«jQuery. Маніпулювання елементами»**

**Варіант № 21**

Дисципліна «Frontend розробка»

Спеціальність «Інженерія програмного забезпечення»

121–ЛР.12–208. 22210824

***Cтудент*** *\_\_\_\_\_\_\_\_\_\_****А. Г.******Погосян***

*(підпис)*

*\_\_30.11.2023 \_\_*

*(дата)*

***Викладач*** *\_\_\_\_\_\_\_\_\_\_\_\_\_* ***Є. О. Давиденко***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2023**

**Лабораторна робота №12**

**Варіант №21**

Закріпити навички роботи з фреймворком jQuery.

**Теоретичні відомості:**

**Завдання:** Language learning (advanced).

Використовуючи фреймворк jQuery модифікувати застосунок контролю вивчення іноземної мови. Передбачити:

* вибір рівня складності проведення контролю.

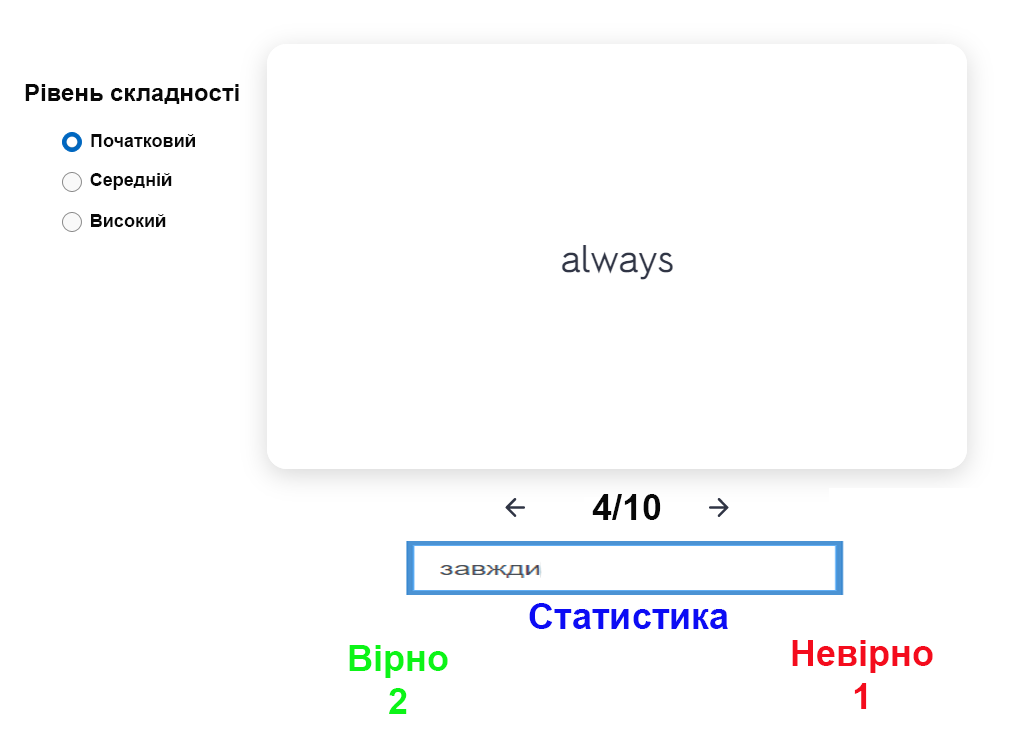


Рисунок 1 – Завдання на лабораторну роботу

Врахувати валідацію та адаптивність.

**HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Learning Language App</title>

  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

  <link rel="stylesheet" href="style.css">

  <script defer src="script.js"></script>

</head>

<body>

    <div id="app">

        <form id="difficulty">

            <label class="form-control">

                <input type="radio" name="difficulty" value="beginner" checked/>

                Beginner

              </label>

              <label class="form-control">

                <input type="radio" name="difficulty" value="intermediate"/>

                Intermediate

              </label>

              <label class="form-control">

                <input type="radio" name="difficulty" value="advanced"/>

                Advanced

              </label>

        </form>

        <div id="card">

            <h4></h4>

            <p></p>

        </div>

        <input type="text" id="translationInput" placeholder="Enter translation">

        <button id="checkButton" style="--c:#7952b3">Start!</button>

        <div id="progress">Word <span id="currentStep">1</span> of 10</div>

        <div id="score">Correct: <span id="correctCount">0</span> | Incorrect: <span id="incorrectCount">0</span></div>

    </div>

    <div id="resultModal" class="modal">

        <div class="modal-content">

            <span class="close">&times;</span>

            <p><span id="proficiencyLevel"></span></p>

        </div>

    </div>

</body>

</html>

**CSS:**

@import url("https://fonts.googleapis.com/css2?family=Nunito+Sans:opsz,wght@6..12,300&display=swap");

body{

    font-family: "Nunito Sans", sans-serif;

    --c:#7952b3;

    --e:#7952b300;

}

#app {

  text-align: center;

  max-width: 1000px; /\* Set a maximum width for larger screens \*/

  height: 60%;

  margin: 0 auto;

}

#card {

  height: 40%;

  margin: 20px;

  padding: 2%;

  font-size: xx-large;

  border: 1px solid #afafaf;

  border-radius: 3rem;

  box-shadow: 0px 0px 10px rgba(0, 0, 0, 8);

  filter: drop-shadow(0 5px 10px 0 #ffffff);

  position: relative;

  z-index: 0;

  overflow: hidden;

  transition: 0.6s ease-in;

}

#card::before {

  content: "";

  position: absolute;

  z-index: -1;

  top: -15px;

  right: -15px;

  background: #7952b3;

  height: 50px;

  width: 50px;

  border-radius: 32px;

  transform: scale(2);

  transform-origin: 50% 50%;

  transition: transform 0.6s ease-out;

}

#card:hover::before {

  transition-delay: 0.1s;

  transform: scale(40);

}

#card:hover {

    cursor: pointer;

  color: #ffffff;

}

#card p{

    font-size: 1vw;

}

#score{

    transition: background-color 0.4s;

    font-size: 1vw;

}

#score,#progress{

  margin-left: 25%;

}

.modal {

  display: none;

  position: fixed;

  top: 0;

  left: 0;

  width: 100%;

  height: 100%;

  background-color: rgba(0, 0, 0, 0.5);

}

.modal-content {

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

  background-color: #fff;

  padding: 20px;

}

.close {

  position: absolute;

  top: 0;

  right: 0;

  padding: 10px;

  cursor: pointer;

}

button {

    box-shadow: 0 0 0 .1em inset var(--c);

    --\_g: linear-gradient(var(--c) 0 0) no-repeat;

    background:

      var(--\_g) calc(var(--\_p,0%) - 100%) 0%,

      var(--\_g) calc(200% - var(--\_p,0%)) 0%,

      var(--\_g) calc(var(--\_p,0%) - 100%) 100%,

      var(--\_g) calc(200% - var(--\_p,0%)) 100%;

    background-size: 50.5% calc(var(--\_p,0%)/2 + .5%);

    outline-offset: .1em;

    transition: background-size .4s, background-position 0s .4s;

    font-size: 2vw;

  }

  button:hover {

    --\_p: 100%;

    transition: background-position .4s, background-size 0s;

  }

  button:active {

    box-shadow: 0 0 9e9q inset #0009;

    background-color: var(--c);

    color: #fff;

  }

  button:disabled {

    box-shadow: 3% inset #0009;

    background-color: gray;

    color: black;

    cursor:not-allowed;

    border: 1ps solid black;

  }

  button {

    font-size: 1rem;

    cursor: pointer;

    padding: .1em .6em;

    font-weight: bold;

    border: none;

  }

  input[type=text] {

    font-weight: bold;

  border: none;

  outline: none;

  border-bottom: .2em solid #7952b3;

  background: rgba(#7952b3, .2);

  border-radius: .2em .2em 0 0;

  padding: .4em;

  color: #7952b3;

}

input[type=text]:disabled {

  font-weight: bold;

border: none;

outline: none;

border-bottom: .2em solid #ff0000;

background: rgba(#ff0000, .2);

border-radius: .2em .2em 0 0;

padding: .4em;

color: #ff0000;

cursor: not-allowed;

}

form {

  display: grid;

  width: 30%;

  display: flex;

  flex-direction: column;

  float: left;

  margin: 1%;

}

.form-control {

  font-weight: bold;

  display: grid;

  grid-template-columns: 1em auto;

  gap: 0.5em;

}

.form-control + .form-control {

  margin-top: 1em;

}

input[type="radio"] {

  appearance: none;

  margin: 0;

  font: inherit;

  color: currentColor;

  width: 1.15em;

  height: 1.15em;

  border: 0.15em solid black;

  border-radius: 50%;

  transform: translateY(-0.075em);

  display: grid;

  place-content: center;

}

input[type="radio"]::before {

  content: "";

  width: 0.65em;

  height: 0.65em;

  border-radius: 50%;

  transform: scale(0);

  transition: 120ms transform ease-in-out;

  box-shadow: inset 1em 1em #7952b3;

  /\* Windows High Contrast Mode \*/

  background-color: CanvasText;

}

input[type="radio"]:checked::before {

  transform: scale(1);

}

input[type="radio"]:focus {

  outline: max(2px, 0.15em) solid #7952b3;

  outline-offset: max(2px, 0.15em);

}

@media only screen and (max-width: 768) {

    #card {

      font-size: 3vw;

    }

    #card p {

      font-size: 2vw;

    }

    #score {

      font-size: 1.5vw;

    }

    button {

      font-size: 1.5vw;

    }

    input[type=text] {

      font-size: 1.5vw;

    }

    #progress {

      font-size: 1.5vw;

    }

  }

**JS:**

  $("#checkButton").click(async function () {

    $("#translationInput").prop("disabled", true);

    checkAnswer();

    $("#translationInput").text = '';

  });

  const beginnerWords=["apple","banana","cat","dog","elephant","fish","grape","house","ice cream",

  "jacket","kite","lamp","moon","nest","orange","pencil","queen","rabbit","sun",

  "tree","umbrella","van","window","xylophone","yellow","zebra"];

  const intermediateWords=["ambiguous","benevolent","capitulate","deleterious","effervescent","facetious",

  "gregarious","haphazard","inquisitive","juxtapose","kaleidoscope","lament","magnanimous","nonchalant",

  "obfuscate","paradigm","quixotic","resilient","serendipity","ubiquitous","vexatious","whimsical",

  "xenophobia","yearn","zephyr","aberration"];

  const advancedWords=["aberrant","belligerent","cacophony","deleterious","efficacious","facetious","garrulous",

  "hedonistic","iconoclast","juxtaposition","kaleidoscopic","labyrinthine","mellifluous","nonplussed",

  "obfuscate","peregrinate","querulous","recalcitrant","salubrious","taciturn","ubiquitous",

  "vexatious","wraithlike","xenophobic","yeomanly","zealous"];

  var wordToTranslate = '';

  var wordTranslation = '';

  var selectedDifficulty = 'beginner';

  var maxWords = 10;

  var currentWord = 0;

  var correctCount = 0;

  var incorrectCount = 0;

function checkAnswer() {

  var userTranslation = $("#translationInput").val().trim().toLowerCase();

  if (currentWord!==0) {

    if (userTranslation === wordTranslation.toLowerCase()) {

      correctCount++;

      $("#score").css('background-color', 'green');

      setTimeout(function() {

          $("#score").css('background-color', 'initial');

        }, 500);

    } else {

      incorrectCount++;

      $("#score").css('background-color', 'red');

      setTimeout(function() {

          $("#score").css('background-color', 'initial');

        }, 500);

    }

    if (currentWord === maxWords) {

      updateScore();

      showResultModal();

      $("button").disabled=true;

    } else {

      currentWord++;

      getWordInfo();

      updateScore();

      $("#translationInput").val('');

    }

  }

  else{

    $("#checkButton").text = 'Check';

    getWordInfo();

    currentWord++;

  }

}

function loadWords(difficulty) {

  switch (difficulty) {

    case 'beginner':

      return beginnerWords.map(word => word.charAt(0).toUpperCase() + word.slice(1));;

    case 'intermediate':

      return intermediateWords.map(word => word.charAt(0).toUpperCase() + word.slice(1));;

    case 'advanced':

      return advancedWords.map(word => word.charAt(0).toUpperCase() + word.slice(1));;

    default:

      break;

  }

}

$('input[name="difficulty"]').change(function() {

  selectedDifficulty = $('input[name="difficulty"]:checked').val();

 console.log(`Chosen difficulty: ${selectedDifficulty}`);

});

async function getWordInfo() {

  var selectedWords = loadWords(selectedDifficulty);

  var selectedWord = selectedWords[Math.floor(Math.random() \* selectedWords.length)];

  $("#card h4").text(selectedWord);

  console.log(`Word: ${selectedWord}`);

  wordToTranslate = selectedWord;

  describeFetchedWord(selectedWord);

  await translateFetchedWord(selectedWord);

}

async function describeFetchedWord(word) {

  const url = `https://wordsapiv1.p.rapidapi.com/words/${word}/definitions`;

  const options = {

    method: 'GET',

    headers: {

      'X-RapidAPI-Key': '7ba5fcb28bmshf0bc269820defaap1c19b9jsn49823a4484be',

      'X-RapidAPI-Host': 'wordsapiv1.p.rapidapi.com'

    }

  };

  try {

    const response = await fetch(url, options);

    const result = await response.json();

    if (result.definitions && result.definitions.length > 0) {

      const definition = result.definitions[0].definition;

      console.log(`${word}'s definition: ${definition}.`);

      $("#card p").text(definition);

      return definition;

    } else {

      console.log('No definition found.');

      $("#card p").text('No definition found.');

      return 'No definition found.';

    }

  } catch (error) {

    console.error(error);

    return 'Error fetching definition.';

  }

}

async function translateFetchedWord(word) {

  const options = {

    method: "POST",

    url: "https://api.edenai.run/v2/translation/automatic\_translation",

    headers: {

      authorization: "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiMjAxZGJjZmMtYjIzNS00MWM2LWEyZjUtNmI5Mjc5MjFhOGFjIiwidHlwZSI6ImFwaV90b2tlbiJ9.3ZeJD0L7iDNwh4REex0GofmEW-Ftmz-Lew9s6ViMJgw",

    },

    data: {

      show\_original\_response: false,

      fallback\_providers: "",

      providers: "amazon,google,ibm,microsoft",

      text: word,

      source\_language: "en",

      target\_language: "uk",

    },

  };

  axios

  .request(options)

  .then((response) => {

    console.log(`Переклад: ${response.data.google ? response.data.google.text : response.data[0].text}` );

    wordTranslation = response.data.google ? response.data.google.text  : response.data[0].text ;

  $("#translationInput").prop("disabled", false);

  })

  .catch((error) => {

    console.error(error);

  });

}

document.addEventListener('keypress', function (e) {

    if (e.key === 'Enter') {

      $("#checkButton").click();

    }

});

function updateScore() {

  $("#progress").html(`Word <span id="currentStep">${currentWord}</span> of 10`);

  $("#correctCount").text(correctCount);

  $("#incorrectCount").text(incorrectCount);

}

function showResultModal() {

    $(":button").prop("disabled", true);

    $(":input").prop("disabled", true);

  $("#proficiencyLevel").text(`You've scored ${correctCount} out of 10!`);

  $("#resultModal").show();

}

$(".close").click(function () {

  $("#resultModal").hide();

});

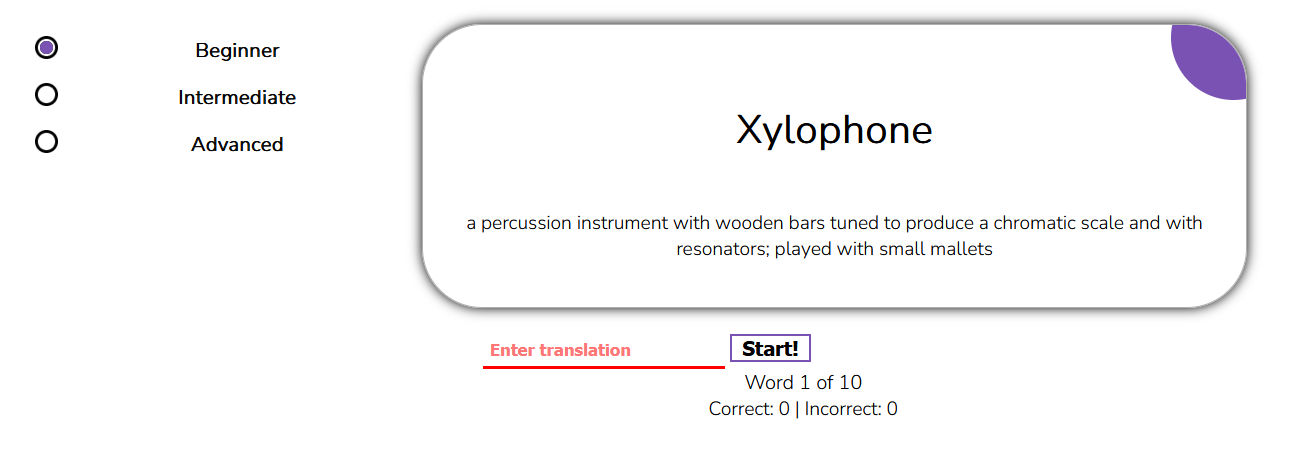


Рисунок 2 – Готова сторінка

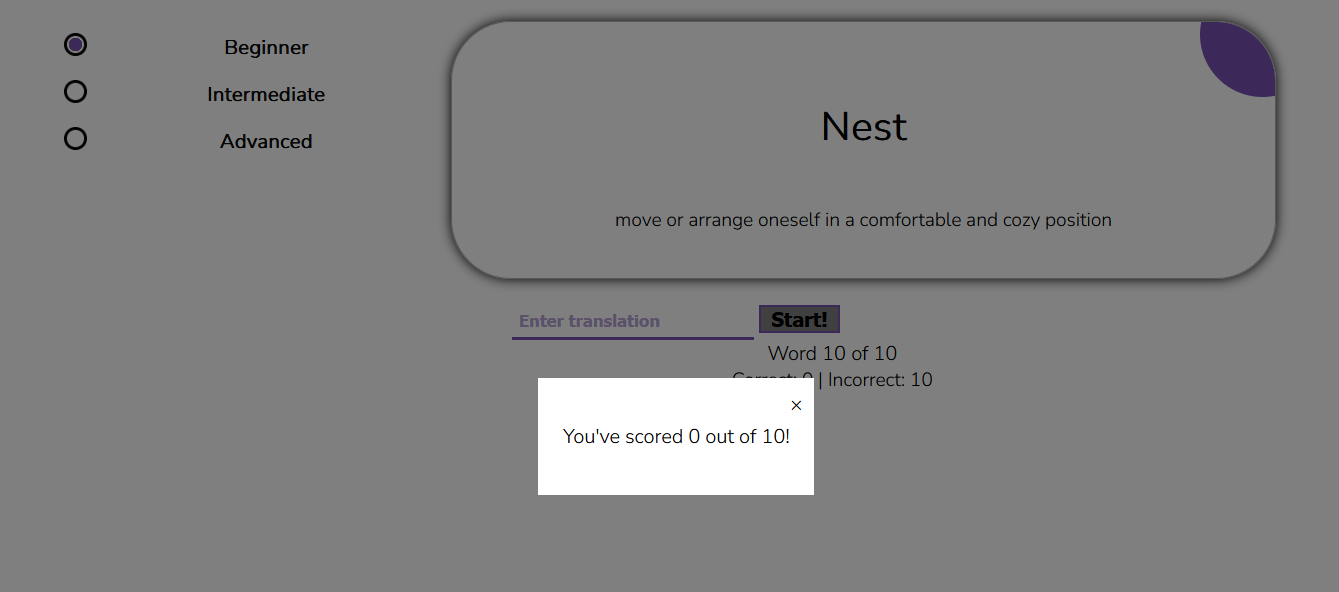


Рисунок 3 – Результат відпрацювання функцій

Посилання на сторінку: [сторінка GitHub Pages](https://trenbolone-acetate.github.io/CardWordTranslateWithLevels/)

Висновок: Під час виконання лабораторної роботи було проведено ознайомлення з способами отримання доступу до HTML елементів за допомогою JQuery. Було повторено умовні оператори, регулярні вислови та масиви. Також було проведено роботу з адаптації сторінки під інші дисплеї за допомогою корегування параметрів CSS. Також було проведено роботу з API для отримання потрібної інформації з інтернету автоматично.