

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки і комп'ютерних технологій

Лабораторна робота №2
З курсу “Веб програмування на стороні клієнта”
“ Використання HTML, CSS, JavaScript для проходження тесту з 5 питань з
рисунками SVG і збереженням результату в JSON-файл”

Виконав:
студент ФЕІ-25
Кайда Матвій
Перевірив:
доц. Анохін В. М.

Львів 2025

Мета: Ознайомитися з принципами створення інтерактивних тестів за допомогою форм HTML, CSS, JavaScript. Використати SVG-графіку для візуального оформлення тесту. Реалізувати збереження результатів тесту у JSON-файл. Використати локальне сховище (LocalStorage) для того, щоб локально зберігати результат попереднього проходження тесту і показувати його поруч з наступним.

Хід роботи:

1. Описуємо html-структуру сторінок:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styleTest.css">
  <title>Тест з SVG</title>
</head>
<body>
  <h1>Тест з SVG</h1>
  <div id="question-container"></div>
  <button id="next-btn">Наступне питання</button>
  <button id="submit-btn">Завершити тест</button>
  <div id="result"></div>

  <script src="testScript.js"></script>
</body>
</html>
```

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styleLogin.css">
  <title>Bxiд</title>
</head>
<body>
  <h2>Bxiд</h2>
  <form id="login-form">
    <label>Email: <input type="email" id="email" required></label><br>
    <label>Password: <input type="password" id="password" required></label><br>
    <button type="submit">Sign in</button>
  </form>
  <p id="error" style="color: red;"></p>
  <button id="register-btn">Registration</button>
  <script src="loginScript.js"></script>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styleLogin.css">
  <title>Client registration</title>
  <script defer src="registrationScript.js"></script>
</head>
<body>
  <h2>Registration form</h2>
  <form id="registrationForm">
    <input type="text" name="name" placeholder="Ім'я" required>
    <input type="email" name="email" placeholder="Email" required>
    <input type="password" name="password" placeholder="Пароль" required>
    <input type="tel" name="phoneNumber" placeholder="Телефон">
    <label><input type="checkbox" name="agreement" required> Allow to process personal data</label>
    <label>Profile photo:<input type="file" name="photo"></label>

    <button type="submit">Register</button>
  </form>
</body>
</html>

```

2. Описуємо фронтенд частину самого тесту:

```

document.addEventListener("DOMContentLoaded", async () => {
  const questionContainer = document.getElementById("question-container");
  const nextBtn = document.getElementById("next-btn");
  const submitBtn = document.getElementById("submit-btn");

  let questions = [];
  let currentQuestionIndex = 0;
  let answers = [];

  async function loadQuestions() {
    const response = await fetch("questions.json");
    questions = await response.json();
    showQuestion();
  }

  function showQuestion() {
    if (currentQuestionIndex >= questions.length) {
      nextBtn.style.display = "none";
      submitBtn.style.display = "block";
      return;
    }

    const q = questions[currentQuestionIndex];
    questionContainer.innerHTML = `
      <h2>${q.question}</h2>
      
      <div>
        ${q.options.map((opt, index) =>
          `<label>
            <input type="radio" name="answer" value="${index}">
            ${opt}
          </label><t>`
        )}.join("")
      </div>
    `;
  }

  nextBtn.addEventListener("click", () => {
    const selected = document.querySelector("input[name='answer']:checked");
    if (!selected) return alert("Виберіть відповідь!");

    answers.push(parseInt(selected.value, 10));
    currentQuestionIndex++;
    showQuestion();
  });

```

```

submitBtn.addEventListener("click", async () => {
    const email = localStorage.getItem("loggedInUser");

    if(!email) {
        location = "login.html";
    }

    const correctAnswers = questions.map(q => q.correct);
    const score = answers.filter((ans, i) => ans === correctAnswers[i]).length;

    const result = {
        email: email,
        date: new Date().toISOString().split("T")[0],
        score
    };
    await fetch("/results", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(result)
    });
    if(email) {
        alert('Тест завершено! Ваш результат: ${score}/${questions.length}\n Попередній результат: ${localStorage.getItem("lastScore")}/${questions.length}');
        localStorage.setItem("lastScore", score);
    }
});

loadQuestions();
});

```

3. Реалізуємо передачу даних введених на форму реєстрації на сервер:

```
document.getElementById('registrationForm').addEventListener('submit', async function(e) {
    e.preventDefault();

    const formData = new FormData(this);
    const client = {
        name: formData.get('name'),
        email: formData.get('email'),
        password: formData.get('password'),
        phoneNumber: formData.get('phoneNumber'),
        agreement: formData.get('agreement') === 'on',
    };

    const file = formData.get('photo');
    if (file && file.size > 0) {
        const reader = new FileReader();
        reader.readAsDataURL(file);
        reader.onload = async function() {
            client.photo = reader.result;
            await sendClientData(client);
        };
    } else {
        await sendClientData(client);
    }
    location = "index.html";
});

async function sendClientData(client) {
    await fetch('/clients', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(client)
    });

    loadClients();
}

async function loadClients() {
    const res = await fetch('/clients');
    const clients = await res.json();
    document.getElementById('clientsList').innerHTML = clients.map(c => `
        <p>
            <strong>${c.name}</strong> - ${c.email}
            <br> Телефон: ${c.phoneNumber}
            <br> Фото: ${c.photo ? `` : 'Немає фото'}
        </p>
    `).join('');
}

loadClients();
```

4. Описуємо передачу даних з логін-сторінки на сервер.

```

document.getElementById("login-form").addEventListener("submit", async (e) => {
  e.preventDefault();

  const email = document.getElementById("email").value.trim();
  const password = document.getElementById("password").value.trim();
  const errorText = document.getElementById("error");

  try {
    const response = await fetch("/clients");
    if (!response.ok) throw new Error("Не вдалося завантажити клієнтів");

    const clients = await response.json();
    const user = clients.find(client => client.email === email && client.password === password);

    if (user) {
      localStorage.setItem("loggedInUser", email);
      window.location.href = "index.html";
    } else {
      errorText.textContent = "Неправильний логін або пароль.";
    }
  } catch (error) {
    errorText.textContent = "Помилка авторизації!";
    console.error(error);
  }
});

document.getElementById("register-btn").addEventListener("click", async (e) => {
  e.preventDefault();
  location = "regForm.html";
});

```

5. Додаємо до серверу, написаному в лаб. №1 метод для збереження результатів тесту:

```

const res_file = path.join(__dirname, "public/result.json");
const loadResults = () => {
  if (!fs.existsSync(res_file)) return [];
  const data = fs.readFileSync(res_file, "utf-8");
  return JSON.parse(data);
};

const saveResults = (results) => {
  fs.writeFileSync(res_file, JSON.stringify(results, null, 2));
};

app.post('/results', (req, res) => {
  const { email, date, score } = req.body;
  if (!email || !date || typeof score !== "number") {
    return res.status(400).json({ error: "Невірний формат даних" });
  }

  const results = loadResults();
  results.push({ email, date, score });
  saveResults(results);

  res.json({ message: "Результат збережено" });
});

```

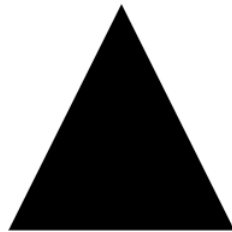
6. Створюємо самі питання тесту та надаємо йому охайнішого вигляду.

```
[
  {
    "question": "Яка фігура на рисунку?",
    "options": ["Квадрат", "Коло", "Прямокутник", "Трикутник"],
    "correct": 3,
    "svg": "triangle.svg"
  },
  {
    "question": "Якого кольору фігура?",
    "options": ["Червоний", "Зелений", "Синій", "Фіолетовий"],
    "correct": 0,
    "svg": "red_figure.svg"
  },
  {
    "question": "Яка фігура зображена на рисунку?",
    "options": ["Квадрат", "Коло", "Трикутник", "Прямокутник"],
    "correct": 1,
    "svg": "circle.svg"
  },
  {
    "question": "Якого кольору об'єкт на рисунку?",
    "options": ["Червоний", "Синій", "Зелений", "Жовтий"],
    "correct": 2,
    "svg": "color.svg"
  },
  {
    "question": "Скільки кутів має фігура на зображенні?",
    "options": ["3", "4", "5", "6"],
    "correct": 2,
    "svg": "pentagon.svg"
  }
]
```

Результати тестування:

Тест з SVG

Яка фігура на рисунку?



☐ Квадрат ☐ Коло ☐ Прямокутник ☐ Трикутник

Наступне питання

Завершити тест

Вхід

Email:

Password:

Sign in

Registration

Registration form

☐

Allow to process personal data

Profile photo:

Choose File

No file chosen

Register

localhost:3000 says

Тест завершено! Ваш результат: 2/5
Попередній результат: 0/5

OK

Висновок: Під час виконання лабораторної роботи було написано тестувальну програму, яка зберігає результати зареєстрованих та авторизованих користувачів.