

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

Звіт
про виконання лабораторної роботи №4,5,6
з курсу «Веб програмування на стороні клієнта»
«4-6 лабораторна робота»

Виконав:
Студент групи ФЕІ-24
Хомич О.Я.

Перевірив:
асистент
Чмихало О.С.

Блок 2

Завдання 1

Мета: вивчення html та html5 форм.

Практичне завдання

Хід роботи

1. Використовуючи теги форм, створюємо форму для опитування з 10 питань.

Для кожного питання використовуємо інший тип форми:

```
<form id="surveyForm">
  <label for="faculty">На якому факультеті ви навчаєтесь:</label>
  <select id="faculty" name="faculty">
    <option value="engineering">Інженерія</option>
    <option value="humanities">Гуманітарні науки</option>
    <option value="science">Природничі науки</option>
  </select><br>
```

Рис.1 Випадаючий список

```
<label>Виберіть потік на якому ви навчаєтесь:</label><br>
<input type="radio" id="stream1" name="stream" value="stream1">
<label for="stream1">Потік 1</label><br>
<input type="radio" id="stream2" name="stream" value="stream2">
<label for="stream2">Потік 2</label><br>

<label>Вкажіть групу:</label><br>
<input type="checkbox" id="groupA" name="group" value="groupA">
<label for="groupA">Група А</label><br>
<input type="checkbox" id="groupB" name="group" value="groupB">
<label for="groupB">Група В</label><br>

<label for="interviewTime">Виберіть зручний час для проходження співбесіди:</label><br>
<input type="datetime-local" id="interviewTime" name="interviewTime"><br>

<label for="averageScore">Вкажіть середній бал:</label><br>
<input type="number" id="averageScore" name="averageScore" min="0" max="100"><br>
```

Рис.2 Введення тексту, вибір дати, введення числа

```
<label for="averageScore">Вкажіть середній бал:</label><br>
<input type="number" id="averageScore" name="averageScore" min="0" max="100"><br>
```

Рис.3 Змінне значення

```
<label>Виберіть потік на якому ви навчаєтесь:</label><br>
<input type="radio" id="stream1" name="stream" value="stream1">
<label for="stream1">Потік 1</label><br>
<input type="radio" id="stream2" name="stream" value="stream2">
<label for="stream2">Потік 2</label><br>
```

Рис.4 Радіо кнопка

```

<label>Вкажіть групу:</label><br>
<input type="checkbox" id="groupA" name="group" value="groupA">
<label for="groupA">Група А</label><br>
<input type="checkbox" id="groupB" name="group" value="groupB">
<label for="groupB">Група В</label><br>

```

Рис.5 Багатоопційний список

2. Проходимо опитування. Результати опитування записуємо в LocalStorage:

```

<script>
  document.getElementById('surveyForm').addEventListener('submit', function(event) {
    event.preventDefault();
    const formData = new FormData(this);
    const surveyData = {};
    for (const [key, value] of formData.entries()) {
      surveyData[key] = value;
    }
    const surveys = JSON.parse(localStorage.getItem('surveys')) || [];
    surveys.push(surveyData);
    localStorage.setItem('surveys', JSON.stringify(surveys));
    alert('Дякуємо за участь у опитуванні!');
    this.reset();
  });

```

Рис.7 Функція запису даних в LocalStorage

```

▼ [{faculty: "engineering", stream: "stream1", group: "groupA", interviewTime: "2024-04-03T14:01",...},...]
  ▶ 0: {faculty: "engineering", stream: "stream1", group: "groupA", interviewTime: "2024-04-03T14:01",...}
  ▶ 1: {faculty: "humanities", stream: "stream1", group: "groupB", interviewTime: "2023-05-12T20:32",...}
  ▶ 2: {faculty: "engineering", stream: "stream1", group: "groupA", interviewTime: "2024-04-01T22:24",...}
  ▶ 3: {faculty: "humanities", stream: "stream2", group: "groupB", interviewTime: "2024-04-02T22:25",...}

```

Рис 8. Приклад збереження даних у LocalStorage

3. Пишемо 3 запити – фільтри, використовуючи відповідно підібрані форми:

```

const surveys = JSON.parse(localStorage.getItem('surveys')) || [];

// Вивести на сторінку учасників опитування лише з одного факультету
const engineeringStudents = surveys.filter(survey => survey.faculty === 'engineering');
console.log('Учасники з факультету Інженерії:', engineeringStudents);

// Вивести учасників, які можуть прийти на співбесіду в певний день і час
const specificInterviewTime = '2024-04-04T12:00'; // Припустимо, це конкретний день і час
const availableParticipants = surveys.filter(survey => new Date(survey.interviewTime) > new Date(specificInterviewTime));
console.log('Учасники, які можуть прийти на співбесіду:', availableParticipants);

// Вивести учасників з середнім балом від 3 до 4
const filteredByScore = surveys.filter(survey => survey.averageScore >= 3 && survey.averageScore <= 4);
console.log('Учасники з середнім балом від 3 до 4:', filteredByScore);
</script>

```

Рис.9 Вибір опції фільтрації

Рис.10 Результати фільтрацій

Учасники з факультету Інженерії:

```
▼ Array(1) 1
  ▶ 0: {faculty: 'engineering', stream: 'stream1', group: 'groupA', interviewTime: '2024-04-03T14:01', averageScore: '100', length: 1}
  ▶ [[Prototype]]: Array(0)
```

Учасники, які можуть прийти на співбесіду: ▶ Array(0)

Учасники з середнім балом від 3 до 4: ▶ Array(0)

Висновок: вивчено форми html та html5. Створено опитування за допомогою форм, дані якого зберігаються в LocalStorage. HTML5 відрізняється від HTML наявністю новими елементами вводу, такими як `<input type="date">`, `<input type="email">`, `<input type="url">`, `<input type="number">`, `<input type="range">` та інші.

Завдання 2

Мета: створити форму тестування.

Практичне завдання

Хід роботи

1. Створюємо json з тестовими питаннями та правильними відповідями на них:

```

{
  "testName": "Тест з WEB дизайну",
  "questions": [
    {
      "question": "Вкажіть тег для блоку?",
      "answers": [
        {
          "answer": "a",
          "isCorrect": false
        },
        {
          "answer": "div",
          "isCorrect": true
        },
        {
          "answer": "img",
          "isCorrect": false
        },
        {
          "answer": "p",
          "isCorrect": false
        }
      ]
    },
    {
      "question": "Вкажіть тег для гіперпосилання?",
      "answers": [
        {
          "answer": "a",
          "isCorrect": true
        },
        {
          "answer": "div",
          "isCorrect": false
        },
        {
          "answer": "span",
          "isCorrect": false
        },
        {
          "answer": "p",
          "isCorrect": false
        }
      ]
    }
  ]
},

```

Рис.1

2-3. Пишемо html, css та js код для відображення на web сторінці та проходження тесту з 5 питань. Перевяємо і показуємо правильність відповіді на кожне питання, виводимо на сторінці сумарний результат проходження тесту:

```

<div id="test-container">

</div>

<button id="submit-btn">Перевірити відповіді</button>
<button id="restart-btn">Пройти тест заново</button>

```

Рис.2 HTML web

```

#test-container {
|   margin-bottom: 20px;
| }

.question {
|   margin-bottom: 20px;
| }

ul {
|   list-style-type: none;
| }

li {
|   margin-bottom: 5px;
| }

#submit-btn {
|   display: block;
|   margin-bottom: 20px;
| }

#result-container {
|   font-weight: bold;
| }

.correct {
|   color: ■ green;
| }

.incorrect {
|   color: ■ red;
| }

```

Рис.3 CSS web

```

const testContainer = document.getElementById('test-container');
const resultContainer = document.getElementById('result-container');
const submitBtn = document.getElementById('submit-btn');
const restartBtn = document.getElementById('restart-btn');

let correctAnswers = 0;

// Підключення JSON файлу з питаннями
fetch('questions.json')
  .then(response => response.json())
  .then(data => renderTest(data));

// Функція для відображення тесту
function renderTest(data) {
  data.questions.forEach(question => {
    const questionElement = document.createElement('div');
    questionElement.classList.add('question');
    questionElement.innerHTML = `
      <p>${index + 1}. ${question.question}</p>
      <ul>
        ${question.answers.map(answer => `<li><input type="checkbox" data-correct=${answer.correct}</li>`)}
      </ul>
    `;
    testContainer.appendChild(questionElement);
  });
}

// Функція для перевірки відповідей
function checkAnswers() {
  correctAnswers = 0;
  const checkboxes = testContainer.querySelectorAll('input[type="checkbox"]');
  checkboxes.forEach(checkbox => {
    if (checkbox.checked && checkbox.dataset.correct === 'true') {
      correctAnswers++;
      checkbox.parentNode.classList.add('correct');
    } else if (checkbox.checked && checkbox.dataset.correct === 'false') {
      checkbox.parentNode.classList.add('incorrect');
    }
  });
  resultContainer.textContent = `Правильних відповідей: ${correctAnswers}`;
}

```

Рис.4 Частина JS web (відображення тесту та обробка відповідей)

```

// Функція для перевірки відповідей
function checkAnswers() {
  correctAnswers = 0;
  const checkboxes = testContainer.querySelectorAll('input[type="checkbox"]');
  checkboxes.forEach(checkbox => {
    if (checkbox.checked && checkbox.dataset.correct === 'true') {
      correctAnswers++;
      checkbox.parentNode.classList.add('correct');
    } else if (checkbox.checked && checkbox.dataset.correct === 'false') {
      checkbox.parentNode.classList.add('incorrect');
    }
  });
  resultContainer.textContent = `Правильних відповідей: ${correctAnswers}`;
}

// Функція для очищення вибраних відповідей і підкреслення
function restartTest() {
  const checkboxes = testContainer.querySelectorAll('input[type="checkbox"]');
  checkboxes.forEach(checkbox => {
    checkbox.checked = false;
    checkbox.parentNode.classList.remove('correct', 'incorrect');
  });
  resultContainer.textContent = '';
}

// Перевірка відповідей при кліці на кнопку "Перевірити відповіді"
submitBtn.addEventListener('click', checkAnswers);

// Очищення вибраних відповідей та підкреслення при кліці на кнопку "Пройти тест заново"
restartBtn.addEventListener('click', restartTest);

```

Рис.5 Друга частина JS web (обробник натискання на кнопку здачі тесту)

1. Вкажіть тег для блоку?

- ☐ a
- ☒ div
- ☐ img
- ☐ p

2. Вкажіть тег для гіперпосилання?

- ☐ a
- ☒ div
- ☐ span
- ☐ p

3. Яка команда використовується для відображення тексту на веб-сторінці?

- ☐ <div>
- ☒
- ☐ <p>
- ☐ <h1>

4. Яка мова використовується для стилізації веб-сторінок?

- ☐ HTML
- ☒ CSS
- ☐ JavaScript
- ☐ PHP

5. Як правильно підключити зовнішній CSS файл до HTML сторінки?

- ☐ <link rel="stylesheet" type="text/css" href="style.css">
- ☒ <style>@import url('style.css');</style>
- ☐ <style src="style.css"></style>
- ☐ <script src="style.css"></script>

Перевірити відповіді

Пройти тест заново

Правильних відповідей: 2

Рис.6 Результат

Висновок: створено форму тестування. Для зберігання питань та відповідей зручно використовувати JSON. Все працює коректно.

Завдання 3

Мета: намалювати SVG, розробити гру та покращити її модифікаціями.

Практичне завдання

Хід роботи

1. Малюємо фрагмент рисунка емблеми факультету у векторній графіці у тегах svg:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.0" xmlns="http://www.w3.org/2000/svg"
width="100" height="100" viewBox="0 0 1621.000000 1848.000000"
preserveAspectRatio="xMidYMid meet">

  <g transform="translate(0.000000,1848.000000) scale(0.100000,-0.100000)"
fill="#000000" stroke="none">
    <path d="M8060 18193 c-383 -17 -748 -93 -1014 -213 -59 -26 -66 -32 -57 -48
17 -32 117 -94 166 -103 54 -10 176 11 235 41 73 37 217 81 349 106 186 36
548 45 752 19 164 -21 328 -62 437 -110 202 -89 307 -79 461 43 123 19 -93 40
c-232 101 -494 163 -809 193 -114 11 -344 17 -450 13z"/>
    <path d="M6760 17867 c-50 -33 -72 -78 -78 -156 -6 -83 8 -127 52 -170 82 -80
226 -29 266 95 25 75 -16 179 -88 225 -56 36 -106 38 -152 6z"/>
    <path d="M9459 17867 c-107 -71 -106 -291 1 -348 81 -44 187 0 231 96 23 49
24 78 4 136 -40 114 -154 171 -236 116z"/>
    <path d="M6495 17600 c-288 -81 -682 -334 -1059 -680 -173 -159 -354 -358
-454 -500 1-44 -61 39 19 c104 53 355 113 601 143 56 7 102 15 102 18 0 4 -25
23 -55 44 -64 44 -162 83 -234 93 -28 4 -51 10 -51 15 0 10 65 28 145 40 33 5
151 12 263 16 111 3 202 9 202 12 -2 30 -137 135 -200 156 -28 9 -48 21 -45
25 19 30 194 46 329 31 54 -6 101 -10 102 -8 2 2 -10 31 -26 65 -31 61 -98
132 -125 132 -37 0 -10 18 42 29 78 15 231 14 338 -4 50 -8 94 -15 98 -15 13
0 -27 51 -104 134 1-71 76 53 24 c30 13 102 38 162 56 59 18 107 37 107 42 0
4 -4 8 -10 8 -13 0 -32 38 -39 76 1-6 31 -60 -17z"/>
  </g>
</svg>
```

Рис.1

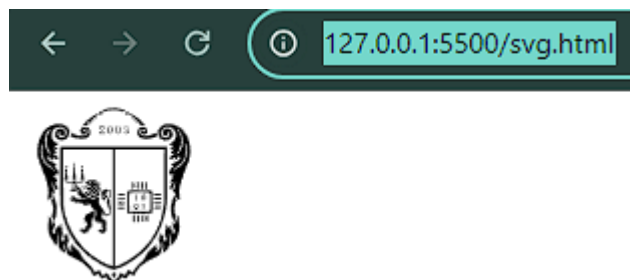


Рис.2 Результат

2. Розробка простої гри з використанням тегу <canvas>:

```
<title>Vertical Moving Cube Game with Obstacles</title>
<style>
  canvas {
    border: 1px solid black;
    background-color: #f0f0f0;
  }
</style>

<script>
  const canvas = document.getElementById('gameCanvas');
  const ctx = canvas.getContext('2d');

  let cubeY = canvas.height / 2;
  let cubeWidth = 20;
  let cubeHeight = 20;
  let dy = -2;
  let score = 0;
  let lives = 3;
  let obstacles = [];
  let objects = [];
  let gameLoop;
```

Рис.3 Створення потрібних об'єктів

```
function drawCube() {
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.beginPath();
  ctx.rect(canvas.width / 2 - cubeWidth / 2, cubeY, cubeWidth, cubeHeight);
  ctx.fillStyle = "#0095DD";
  ctx.fill();
  ctx.closePath();
}
```

Рис.4 Функція створення кубу

```
function drawObstacles() {
  obstacles.forEach(function(obstacle, index) {
    ctx.beginPath();
    ctx.rect(obstacle.x, obstacle.y, obstacle.width, obstacle.height);
    ctx.fillStyle = "#FF0000";
    ctx.fill();
    ctx.closePath();
  });
}
```

Рис.5 Функція створення перешкод

```
function generateObject() {
  let objectRadius = 10;
  let object = {
    x: canvas.width,
    y: Math.random() * canvas.height,
    radius: objectRadius
  };
  objects.push(object);
}
```

Рис.6 Функція створення шарів (для очків)

```
function collisionDetection() {
  obstacles.forEach(function(obstacle, index) {
    if(cubeY < obstacle.y + obstacle.height &&
       cubeY + cubeHeight > obstacle.y &&
       canvas.width / 2 > obstacle.x &&
       canvas.width / 2 < obstacle.x + obstacle.width) {
      lives--;
      obstacles.splice(index, 1);
    }
  });
}
```

Рис.7 Функція руху кубу

```
function generateObstacle() {
  let obstacleHeight = Math.floor(Math.random() * 20) + 10;
  let obstacle = {
    x: canvas.width,
    y: Math.random() * (canvas.height - obstacleHeight),
    width: 20,
    height: obstacleHeight
  };
  obstacles.push(obstacle);
}
```

Рис.8 Функція появи нових перешкод

```

function draw() {
  drawCube();
  drawObstacles();
  drawObjects();
  drawScore();
  drawLives();

  if(cubeY + dy < 0 || cubeY + dy > canvas.height - cubeHeight) {
    dy = -dy;
  }

  cubeY += dy;

  collisionDetection();

  obstacles.forEach(function(obstacle, index) {
    obstacle.x -= 2;
    if(obstacle.x + obstacle.width < 0) {
      obstacles.splice(index, 1);
    }
  });

  objects.forEach(function(object, index) {
    object.x -= 2;
    if(object.x + object.radius < 0) {
      objects.splice(index, 1);
    }
  });

  if(obstacles.length < 5 && Math.random() < 0.05) {
    generateObstacle();
  }

  if(objects.length < 5 && Math.random() < 0.03) {
    generateObject();
  }
}

```

Рис.9 Функція скидання гри до початкового стану при зіткненні зі перешкодами

```

function startGame() {
  score = 0;
  lives = 3;
  obstacles = [];
  objects = [];
  gameLoop = setInterval(draw, 10);
  document.getElementById('startAgainButton').style.display = "none";
}

let startButton = document.getElementById('startAgainButton');
startButton.addEventListener('click', startGame);

document.addEventListener("keydown", moveCube);

```

Рис.10 Функція перезапуску гри та відновлення гри на екрані

```

let startButton = document.getElementById('startAgainButton');
startButton.addEventListener('click', startGame);

document.addEventListener("keydown", moveCube);

function moveCube(event) {
  if(event.key === "ArrowUp") {
    dy = -2;
  } else if(event.key === "ArrowDown") {
    dy = 2;
  }
}

```

Рис.11 Обробник подій натискання на клавіші

3. Додаємо модифікації гри:

```

function drawLives() {
  ctx.font = "16px Arial";
  ctx.fillStyle = "#0095DD";
  ctx.fillText("Lives: " + lives, canvas.width - 65, 20);
}

```

```

function startGame() {
  score = 0;
  lives = 3;
  obstacles = [];
  objects = [];
  gameLoop = setInterval(draw, 10);
  document.getElementById('startAgainButton').style.display = "none";
}

```

Рис.12 Тепер у гравця є 3 життя (при зіткненні з перешкодами вони віднімаються)

```

function generateObstacle() {
  let obstacleHeight = Math.floor(Math.random() * 20) + 10;
  let obstacle = {
    x: canvas.width,
    y: Math.random() * (canvas.height - obstacleHeight),
    width: 20,
    height: obstacleHeight
  };
  obstacles.push(obstacle);
}

```

```

function drawScore() {
  ctx.font = "16px Arial";
  ctx.fillStyle = "#0095DD";
  ctx.fillText("Score: " + score, 8, 20);
}

```

Рис.13 Додаємо розрахунок очок гравця (за зібрані шари)

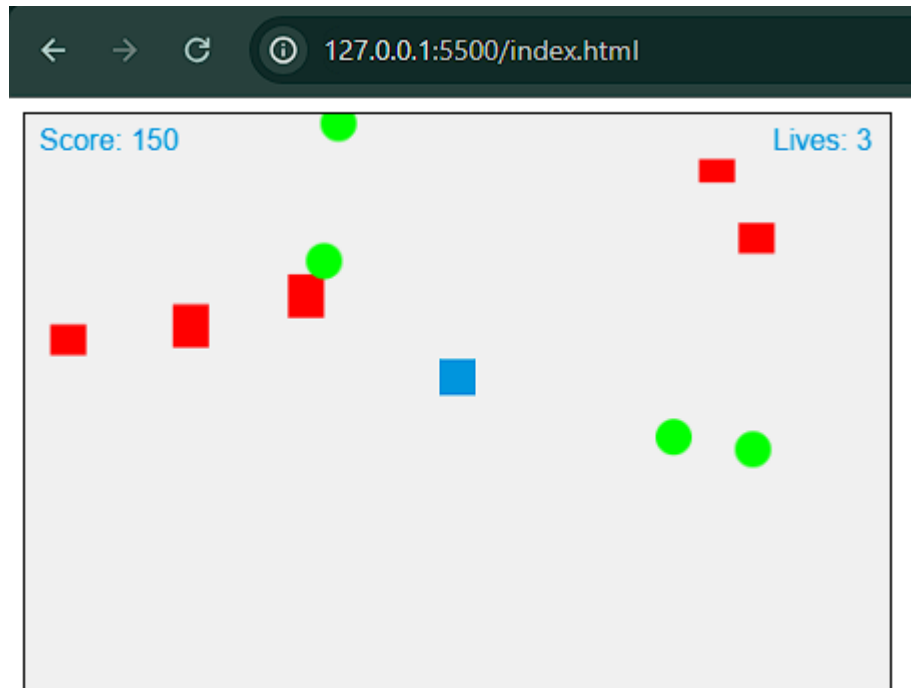


Рис.14 Результат

Висновок: намальовано svg. Створено гру та модифіковано її.
Получено досвід роботи з тегом <canvas>. Все працює коректно.