

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

Звіт

про виконання лабораторних робіт Блок 2
З курсу “ Веб програмування на стороні клієнта”
“JavaScript”

Виконала
Студентка
групи ФЕІ-22
Тищенко Анна
Перевірів
доц. Анохін В.Є.

Львів 2024

Мета: вивчення html та html5 форм, растрова і векторна графіка на frontend, розробка простої гри.

Хід роботи:

1. Створюю форму опитування для свого сайту, для кожного питання використовую інший тип форми:

```
<!-- work section starts -->
<section class="work" id="work">
  <h1 class="heading">Зацікавлені у <span>Роботі</span> з нами <span></span></h1>
  <div class="test">
    <form id="form">
      <label for="fullName">Прізвище та ім'я:</label><br>
      <input type="text" id="fullName" name="fullName" required><br><br>
      <label for="city">Місто:</label><br>
      <input type="text" id="city" name="city" required><br><br>
      <label for="position">Посада, на яку претендуєте:</label><br>
      <select id="position" name="position" required>
        <option value="developer">Розробник програмного забезпечення</option>
        <option value="designer">Дизайнер</option>
        <option value="manager">Менеджер проєктів</option>
      </select><br><br>
      <label>Ваш рівень освіти:</label><br>
      <input type="radio" id="educationBachelor" name="education" value="bachelor">
      <label for="educationBachelor">Бакалавр</label><br>
      <input type="radio" id="educationMaster" name="education" value="master">
      <label for="educationMaster">Магістр</label><br>
      <input type="radio" id="educationPhD" name="education" value="phd">
      <label for="educationPhD">Доктор наук</label><br><br>
      <label for="experience">Кількість років досвіду роботи:</label><br>
      <input type="number" id="experience" name="experience" min="0" required><br><br>
      <label for="skills">Вкажіть ваші професійні навички та навички комунікації:</label><br>
      <textarea id="skills" name="skills" rows="4" cols="50" required></textarea><br><br>
      <label for="lastJob">Останнє місце роботи:</label><br>
      <input type="text" id="lastJob" name="lastJob" required><br><br>
      <label for="phoneNumber">Контактний номер телефону:</label><br>
      <input type="tel" id="phoneNumber" name="phoneNumber" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" placeholder="XXX-XX-XXX" required><br><br>
      <label>Чи маєте ви досвід роботи зі згаданими технологіями:</label><br>
      <input type="checkbox" id="experienceHTML" name="experienceHTML" value="HTML">
      <label for="experienceHTML">HTML/CSS & React</label><br>
      <input type="checkbox" id="experienceCSS" name="experienceCSS" value="CSS">
      <label for="experienceCSS">Python/PHP</label><br>
      <input type="checkbox" id="experienceJavaScript" name="experienceJavaScript" value="JavaScript">
      <label for="experienceJavaScript">JavaScript/Typescript</label><br><br>
      <label for="expectations">Які ваші очікування від цієї роботи?</label><br>
      <textarea id="expectations" name="expectations" rows="4" cols="50" required></textarea><br><br>
      <input type="submit" value="Відправити" class="btn">
    </form>
  </div>
</section>
```

Результати опитування зберігаю у LocalStorage:

```
document.addEventListener('DOMContentLoaded', function() {
  const form = document.getElementById('form');

  form.addEventListener('submit', function(event) {
    event.preventDefault();

    const formData = {
      fullName: form.fullName.value,
      city: form.city.value,
      position: form.position.value,
      education: document.querySelector('input[name="education"]:checked').value,
      experience: form.experience.value,
      skills: form.skills.value,
      lastJob: form.lastJob.value,
      phoneNumber: form.phoneNumber.value,
      experienceHTML: form.experienceHTML.checked,
      experienceCSS: form.experienceCSS.checked,
      experienceJavaScript: form.experienceJavaScript.checked,
      expectations: form.expectations.value
    };

    localStorage.setItem('formData', JSON.stringify(formData));

    form.reset();

    alert('Форма успішно відправлена!');
```

Прописую запит-фільтри, якщо кандидат має достатній досвід, має знання в усіх мовах, та має освіту бакалавра, користувача посилає на нову сторінку з написаним мною тестом:

```
if (formData.experience >= 2) {
  if (formData.experienceJavaScript && formData.experienceCSS && formData.experienceHTML) {
    if (formData.education !== "") {
      const queryParams = new URLSearchParams(formData);
      const url = "test.html?" + queryParams.toString();

      window.location.href = url;
    }
  }
}
```

CSS:

```
.work .test form {
  display: block;
  flex-wrap: wrap;
  gap: 1rem;
}

.work .test form label {
  font-weight: bold;
  font-size: 1.5rem;
  text-decoration: none;
  display: block;
  margin-bottom: 0.1rem;
}

.work .test form label[for="educationBachelor"],
.work .test form label[for="educationMaster"],
.work .test form label[for="educationPhD"] {
  display: flex;
  font-weight: normal;
}

.work .test form label[for="experienceHTML"],
.work .test form label[for="experienceCSS"],
.work .test form label[for="experienceJavaScript"] {
  display: flex;
  font-weight: normal;
}

.work .test form input[type="text"],
.work .test form input[type="number"],
.work .test form input[type="tel"],
.work .test form textarea,
.work .test form select {
  width: 100%;
  padding: 0.5rem;
  margin-bottom: 1rem;
  border: 1px solid #ccc;
  border-radius: 0.5rem;
}

.work .test form input[type="checkbox"] {
  width: auto;
  margin-right: 0.5rem;
}

.work .test form .btn {
  padding: 0.75rem 1.5rem;
  background-color: #333;
  color: #fff;
  border: none;
  border-radius: 0.5rem;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.work .test form .btn:hover {
  background-color: var(--purple);
}
```

2. Прописую тест:

Створюю json з питаннями та правильними відповідями:

```
const questions = [
  {
    question: "Яка основна різниця між HTML та XHTML?",
    answers: ["XHTML підтримує строгий синтаксис XML, тоді як HTML – це SGML-подібна мова.", "HTML може містити тільки теги, які відповідають стандарту HTML 4.01, тоді як XHTML може містити будь-які теги.", "HTML не потребує закриваючих тегів, тоді як XHTML вимагає, щоб кожен тег був правильно закритий.", "XHTML може містити скриптові файли, correctAnswer: "XHTML підтримує строгий синтаксис XML, тоді як HTML – це SGML-подібна мова."],
  },
  {
    question: "Що таке бандлер в контексті JavaScript розробки?",
    answers: ["Інструмент для стиснення (зжимання) JavaScript файлів.", "Інструмент для об'єднання декількох JavaScript файлів в один.", "Інструмент для автоматизації процесу розгортання веб-додатків.", "Інструмент для управління залежностями модулів JavaScript."],
    correctAnswer: "Інструмент для об'єднання декількох JavaScript файлів в один."
  },
  {
    question: "Що таке props в React?",
    answers: ["Скорочення від properties – властивості компонентів React.", "Спеціальний тип стейту в React компонентах.", "Об'єкт, який передає дані вниз по ієрархії компонентів React.", "Метод, що дозволяє змінювати стан компонента ззовні."],
    correctAnswer: "Об'єкт, який передає дані вниз по ієрархії компонентів React."
  },
  {
    question: "Які з наступних операторів не підтримуються в Python?",
    answers: ["!=", "<<", "++", "//"],
    correctAnswer: "++"
  },
  {
    question: "Яка з наступних опцій є коректним способом оголошення змінної в TypeScript?",
    answers: ["var x: number = 5;", "let x = 5;", "const x: number = 5;", "int x = 5;"],
    correctAnswer: "const x: number = 5;"
  }
];
```

```

<body>
  <h1>Ви є кандидатом на вакантне місце. <span>Пройдіть опитування нижче.</span></h1>
  <div id="quiz-container">
    <h2 id="question"></h2>
    <div id="answers"></div>
    <button id="next-btn">Наступне питання</button>
  </div>

  <div id="result-container" style="display: none;">
    <h2 id="score"></h2>
    <button id="show-answers-btn">Показати правильні відповіді</button>
    <div id="correct-answers" style="display: none;">
      <h3>Правильні відповіді:</h3>
      <ul id="correct-answers-list"></ul>
    </div>
    <h3>Чекайте на повідомлення від нас!</h3>
  </div>
</body>

```

```

const quizContainer = document.getElementById("quiz-container");
const resultContainer = document.getElementById("result-container");
const questionElement = document.getElementById("question");
const answersElement = document.getElementById("answers");
const scoreElement = document.getElementById("score");
const nextButton = document.getElementById("next-btn");
const showAnswersButton = document.getElementById("show-answers-btn");
const correctAnswersList = document.getElementById("correct-answers-list");

let currentQuestionIndex = 0;
let score = 0;
let userAnswers = [];

function showQuestion() {
  const question = questions[currentQuestionIndex];
  questionElement.textContent = question.question;
  answersElement.innerHTML = "";

  question.answers.forEach((answer, index) => {
    const radioBtn = document.createElement("input");
    radioBtn.type = "radio";
    radioBtn.name = "answer";
    radioBtn.value = answer;
    radioBtn.id = `answer${index}`;

    const label = document.createElement("label");
    label.textContent = answer;
    label.setAttribute("for", `answer${index}`);

    const answerDiv = document.createElement("div");
    answerDiv.appendChild(radioBtn);
    answerDiv.appendChild(label);

    answersElement.appendChild(answerDiv);
  });
}

```

```

function checkAnswer() {
  const selectedAnswer = document.querySelector('input[name="answer"]:checked');
  if (selectedAnswer) {
    const answerValue = selectedAnswer.value;
    userAnswers.push(answerValue);
    if (answerValue === questions[currentQuestionIndex].correctAnswer) {
      score++;
    }
    currentQuestionIndex++;
    if (currentQuestionIndex < questions.length) {
      showQuestion();
    } else {
      showResult();
    }
  } else {
    alert("Будь ласка, оберіть відповідь!");
  }
}

function showResult() {
  quizContainer.style.display = "none";
  resultContainer.style.display = "block";
  scoreElement.textContent = `Ваш результат: ${score} з ${questions.length}`;
}

nextButton.addEventListener("click", checkAnswer);

showQuestion();

```

```

showAnswersButton.addEventListener("click", () => {
  const answersList = questions.map((question, index) => {
    return `<li><strong>${question.question}</strong><br>
    Ваша відповідь: ${userAnswers[index]}<br>
    Правильна відповідь: ${question.correctAnswer}</li>`;
  });
  correctAnswersList.innerHTML = answersList.join("");
  document.getElementById("correct-answers").style.display = "block";
});

```

Отримуються посилання на різні елементи DOM, такі як контейнери для питань та результатів, кнопки "Наступне" та "Показати відповіді", і т. д. Далі, встановлюються змінні `currentQuestionIndex`, `score` та `userAnswers`, які використовуються для відстеження поточного питання, рахунку балів та збереження відповідей користувача відповідно. Функція `showQuestion()` викликається для відображення поточного питання з варіантами відповідей. Функція `checkAnswer()` визначає правильну відповідь користувача, оновлює рахунок балів, перемикається на наступне питання і, якщо це останнє питання, викликає функцію `showResult()` для відображення результатів. Функція `showResult()` приховує контейнер тесту та показує контейнер результатів, виводячи загальний рахунок користувача. Обробники подій додаються для кнопок "Наступне" та "Показати відповіді", щоб вони викликали функції `checkAnswer()` та `showAnswers()`, відповідно. Функція `showAnswers()` викликається при натисканні кнопки "Показати відповіді", щоб показати всі правильні відповіді користувача та відповіді, які очікувалися.

CSS:

```

body {
  transition: .25 linear;
  font-size: 62.5%;
  scroll-behavior: smooth;
  scroll-padding-top: 6rem;
  overflow-x: hidden;
  margin: 0;
  padding: 0;
  align-items: center;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
}

h2 {
  margin-top: 0;
}

h1 {
  text-align: center;
  margin-bottom: 3rem;
  margin-top: 3rem;
  font-size: 1.5rem;
}

span {
  color: #003b5d;
}

#quiz-container, #result-container {
  border: 6px solid #003b5d;
  border-style: dotted;
  padding: 20px;
  margin-bottom: 20px;
  text-align: center;
  width: 450px;
  height: 450px;
  max-height: 80vh;
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
}

#answers div {
  margin-bottom: 10px;
  font-size: 1.5rem;
}

#answers label {
  margin-left: 5px;
  font-size: 1rem;
}

button {
  padding: 10px 20px;
  background-color: #333;
  color: #fff;
}

#correct-answers {
  margin-top: 20px;
}

#correct-answers h3 {
  margin-top: 0;
}

#correct-answers ul {
  list-style-type: none;
  padding: 0;
}

#correct-answers li {
  margin-bottom: 10px;
}

@media only screen and (max-width: 600px) {
  h1 {
    margin-top: 1rem;
    margin-bottom: 1rem;
  }
}

```

3. Малюю невеликий фрагмент емблеми факультету в svg графіці:

```
<svg width="50" height="50">
  <rect x="5" y="5" width="40" height="40" rx="10" ry="2.5" fill="none" stroke="#000080" stroke-width="2.5"/>
  <line x1="10" y1="5" x2="10" y2="0" stroke="#000080" stroke-width="2"/>
  <line x1="20" y1="5" x2="20" y2="0" stroke="#000080" stroke-width="2"/>
  <line x1="30" y1="5" x2="30" y2="0" stroke="#000080" stroke-width="2"/>
  <line x1="40" y1="5" x2="40" y2="0" stroke="#000080" stroke-width="2"/>

  <line x1="45" y1="10" x2="50" y2="10" stroke="#000080" stroke-width="2"/>
  <line x1="45" y1="20" x2="50" y2="20" stroke="#000080" stroke-width="2"/>
  <line x1="45" y1="30" x2="50" y2="30" stroke="#000080" stroke-width="2"/>
  <line x1="45" y1="40" x2="50" y2="40" stroke="#000080" stroke-width="2"/>

  <line x1="10" y1="45" x2="10" y2="50" stroke="#000080" stroke-width="2"/>
  <line x1="20" y1="45" x2="20" y2="50" stroke="#000080" stroke-width="2"/>
  <line x1="30" y1="45" x2="30" y2="50" stroke="#000080" stroke-width="2"/>
  <line x1="40" y1="45" x2="40" y2="50" stroke="#000080" stroke-width="2"/>

  <line x1="5" y1="10" x2="0" y2="10" stroke="#000080" stroke-width="2"/>
  <line x1="5" y1="20" x2="0" y2="20" stroke="#000080" stroke-width="2"/>
  <line x1="5" y1="30" x2="0" y2="30" stroke="#000080" stroke-width="2"/>
  <line x1="5" y1="40" x2="0" y2="40" stroke="#000080" stroke-width="2"/>

  <text x="12.5" y="22.5" font-family="Arial" font-size="15" fill="#000080">1</text>
  <text x="27.5" y="22.5" font-family="Arial" font-size="15" fill="#000080">0</text>
  <text x="12.5" y="37.5" font-family="Arial" font-size="15" fill="#000080">0</text>
  <text x="27.5" y="37.5" font-family="Arial" font-size="15" fill="#000080">1</text>
</svg>
```

4. Створюю на сторінці сайту невелику гру:

```
<!DOCTYPE html>
<html Lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Гпа</title>
    <link rel="stylesheet" href="stylegame.css">
  </head>

  <body>
    <script src="script.js"></script>
    <!-- <h1>Кошення <span>ма</span> лана</h1> -->
    <div id="startScreen" class="start-screen">
      <button id="startButton" class="start-button">Start</button>
    </div>
    <div class="game">
      <div id="score" class="score">Score: <span id="scoreValue">0</span></div>
      <div id="cat"></div>
      <div id="paw"></div>

      <div class="sparkle"></div>
    </div>
    <div class="container">
      <div id="bestResults"></div>
      <div id="worstResults"></div>
    </div>
  </body>
</html>
```

Прописую стилі та анімації до всіх компонент гри:

```
.start-screen {
  width: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  margin: auto;
  background-color: rgba(255, 255, 255, 0.5);
  position: relative;
}

.start-button {
  padding: 20px 40px;
  font-size: 20px;
  background-color: rgba(180, 70, 116, 0.5);
  color: #ffff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.start-button:hover {
  background-color: rgba(180, 70, 116, 0.5);
}

.game {
  display: none;
}

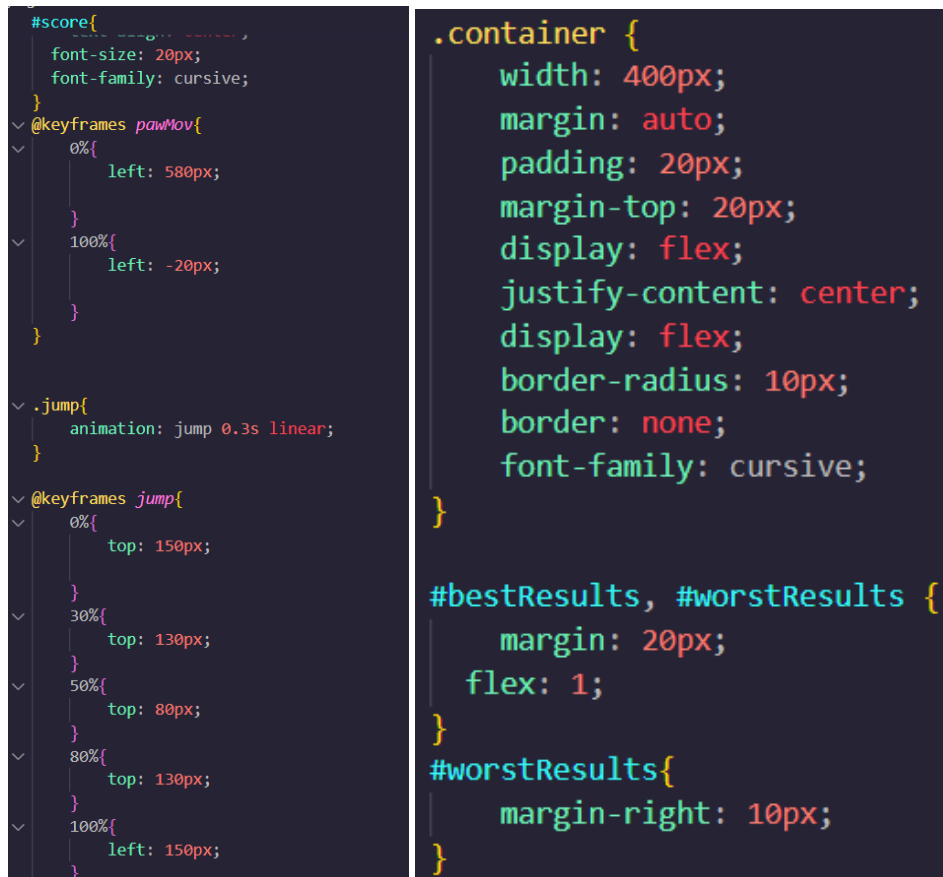
.game {
  width: 600px;
  height: 240px;
  background-image: url(img/888.jpg);
  background-repeat: repeat-x;
  border-radius: 10px;
  margin: auto;
  background-size: 600px 240px;
}

#cat {
  width: 60px;
  height: 60px;
  background-image: url(img/cat.png);
  background-size: 60px 60px;
  position: relative;
  top: 150px;
}

#paw {
  width: 50px;
  height: 40px;
  background-image: url(img/paw.png);
  background-size: 50px 40px;
  position: relative;
  top: 110px;
  left: 580px;
  animation: pawMov 1s infinite linear;
}

@keyframes sparkleMove {
  0% {
    left: 580px;
    opacity: 0;
  }
  90% {
    left: 50px;
    opacity: 1;
  }
  100% {
    left: -20px;
    opacity: 0;
  }
}

.sparkle {
  width: 40px;
  height: 40px;
  background-image: url(img/magic.png);
  background-size: 40px 40px;
  position: relative;
  top: 20px;
  left: 580px;
  opacity: 0;
  animation: sparkleMove 1s linear forwards;
}
```



В грі є нерухома компонента raw при зіткненні з якою гра закінчується, та компонента sparkle при зіткненні з якою до рахунку гри додається +10 до балів.

Прописую в JS:

```

document.addEventListener("DOMContentLoaded", function() {
  const startScreen = document.getElementById("startScreen");
  const startButton = document.getElementById("startButton");

  const game = document.querySelector(".game");
  const scoreDisplay = document.getElementById("scoreValue");
  const bestResultsDiv = document.getElementById("bestResults");
  const worstResultsDiv = document.getElementById("worstResults");
  const container = document.querySelector(".container");
  startButton.addEventListener("click", function() {
    startScreen.style.display = "none";
    game.style.display = "block";
    score = 0;
    scoreDisplay.textContent = score;
  });
  const sparkle=document.getElementById("sparkle");
  const cat = document.getElementById("cat");
  const paw = document.getElementById("paw");
  let scores = JSON.parse(localStorage.getItem("scores")) || [];
  let score = 0;

  document.addEventListener("keydown", function(event) {
    jump();
  });
});

```

```

function jump() {
  if (cat.classList != "jump") {
    cat.classList.add("jump");
  }
  setTimeout(function() {
    cat.classList.remove("jump");
  }, 300);
  score++;
  scoreDisplay.textContent = score;
  document.getElementById("scoreValue").textContent = score;
}

function createSparkle() {
  const sparkle = document.createElement("div");
  sparkle.classList.add("sparkle");

  document.querySelector(".game").appendChild(sparkle);

  const catTop = parseInt(window.getComputedStyle(cat).getPropertyValue("top"));
  const catLeft = parseInt(window.getComputedStyle(cat).getPropertyValue("left"));

  setTimeout(() => {
    const sparkleTop = parseInt(window.getComputedStyle(sparkle).getPropertyValue("top"));
    const sparkleLeft = parseInt(window.getComputedStyle(sparkle).getPropertyValue("left"));

    const sparkleLeftC = parseInt(window.getComputedStyle(sparkle).getPropertyValue("left"));

    if (sparkleLeftC <= catLeft + 60 && sparkleLeft >= catLeft - 10 && catTop >= 140) {
      score += 10;
      scoreDisplay.textContent = score;
      document.getElementById("scoreValue").textContent = score;
    }

    sparkle.remove();
  }, 1000);
}

setInterval(createSparkle, 5000);

let isAlive = setInterval(function(){
  let catTop = parseInt(window.getComputedStyle(cat).getPropertyValue("top"));
  let pawLeft = parseInt(window.getComputedStyle(paw).getPropertyValue("left"));
  let catLeft = parseInt(window.getComputedStyle(cat).getPropertyValue("left"));
  if (pawLeft <= catLeft + 60 && pawLeft >= catLeft - 10 && catTop >= 140) {
    scores.push(score);
    localStorage.setItem("scores", JSON.stringify(scores));
    scores.sort((a, b) => b - a);

    displayScores();
    score = 0;

    game.style.display = "none";
    startScreen.style.display = "block";
  }
}, 10)

function displayScores() {
  bestResultsDiv.innerHTML = "<h3>Top 3 Scores:</h3>";
  worstResultsDiv.innerHTML = "<h3>Bottom 3 Scores:</h3>";

  scores.slice(0, 3).forEach((s, i) => {
    bestResultsDiv.innerHTML += `<p>${i + 1}.Score: ${s}</p>`;
  });

  const nonZeroScores = scores.filter(score => score !== 0);
  const sortedUniqueNonZeroScores = nonZeroScores.filter((score, index, array) => array.indexOf(score) === index);
  const sortedUniqueWorstScores = sortedUniqueNonZeroScores.sort((a, b) => a - b);
  const worstScores = sortedUniqueWorstScores.slice(0, 3);
  worstScores.forEach((s, i) => {
    worstResultsDiv.innerHTML += `<p>${i + 1}.Score: ${s}</p>`;
  });
  container.style.border = "4px solid";
  container.style.borderImage = "linear-gradient(to right, violet, indigo, blue, green, yellow, orange, red)";
  container.style.borderImageSlice = "1";
  container.style.borderRadius = "10px";
}

```

Функція `jump()`:Ця функція відповідає за "стрибок" кота у грі. Вона перевіряє, чи вже кіт не знаходиться у стані стрибка (перевірка відбувається через перевірку класу `jump`). Якщо кіт не стрибає, то додається клас `jump` до елементу `cat`. Після короткої затримки (300 мілісекунд),

клас `jump` видаляється з елемента `cat`, щоб імітувати кінцевий ефект стрибка. Крім того, функція збільшує лічильник `score` та оновлює відображення результату.





Функція `createSparkle()`: Ця функція динамічно створює і "анімує" блискітки на екрані гри. Вона створює новий елемент `<div>` з класом `sparkle` і додає його до елемента `.game`. Після короткої затримки, функція перевіряє, чи сталось зіткнення кота з блискіткою. Якщо таке зіткнення відбулося, лічильник `score` збільшується на 10. Після перевірки блискітка видаляється з екрану гри.

Функція `displayScores()`: Ця функція відповідає за відображення кращих і гірших результатів гри. Вона очищає вміст елементів `bestResultsDiv` та `worstResultsDiv`. Потім вона виводить три найвищих результати у елемент `bestResultsDiv` та три найнижчих ненульових результати у елемент `worstResultsDiv`.

Також в тілі інтервалу перевіряє чи відбулася колізія між котом та лапою.

Результати:

1. Форма опитування:

 [Головна](#) [Про Нас](#) [Товари](#) [Відгуки](#) [Контакти](#) [Місцезнаходження](#) [Робота](#) [Гра](#)   

Зацікавлені У Роботі З Нами ?

Прізвище Та Ім'я:

Місто:

Посада, На Яку Претендуєте:

Розробник Програмного Забезпечення

Ваш Рівень Освіти:

☐

 Бакалавр

☐

 Магістр

☐

 Доктор Наук

Кількість Років Досвіду Роботи:

Прізвище Та Ім'я:

Тищенко Анна

Місто:

Львів

Посада, На Яку Претендуєте:

Дизайнер

Ваш Рівень Освіти:

☒

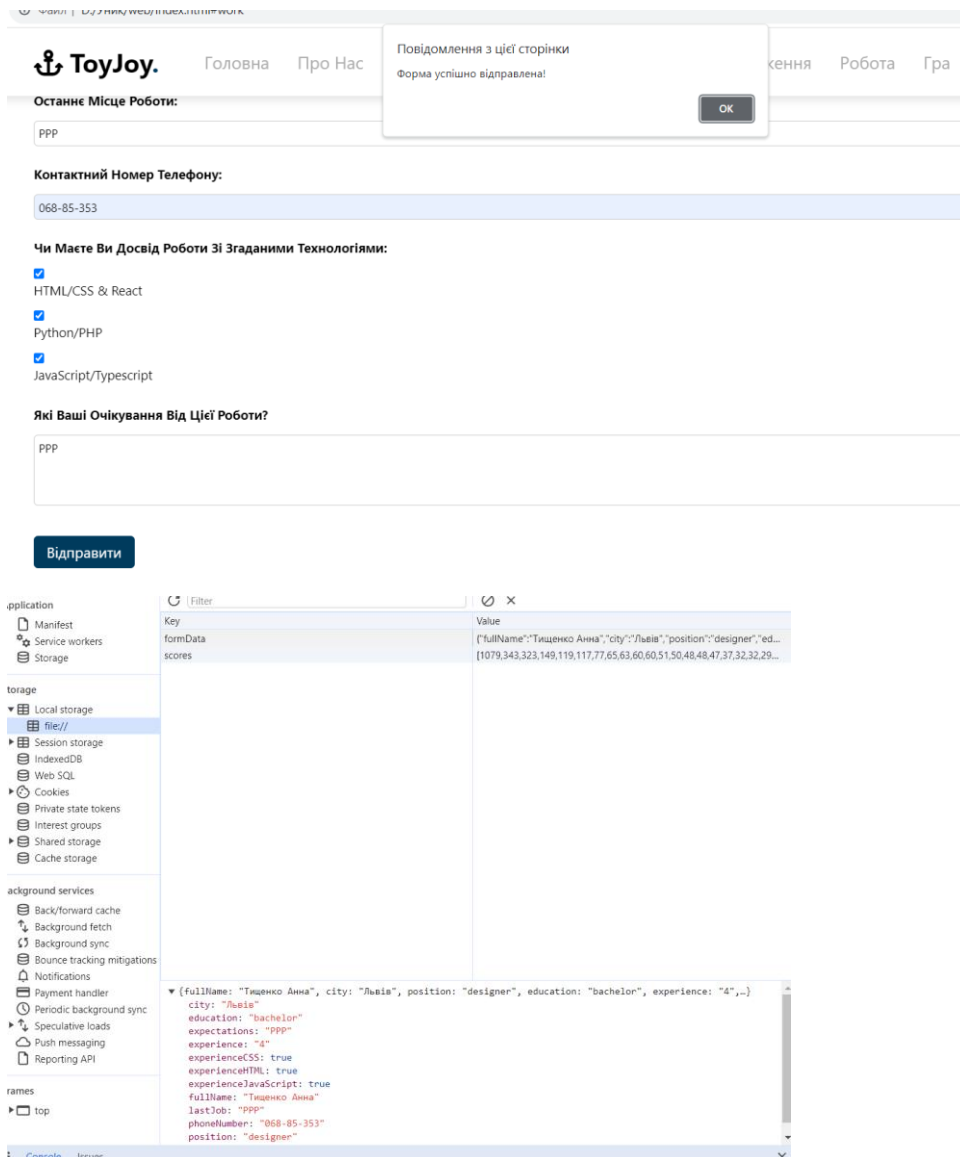
 Бакалавр

☐

 Магістр

☐

 Доктор Наук



Зразу відкривається нова сторінка з тестом, оскільки я написала 4 роки досвіду, вибрала всі мови, та обрала рівень освіти бакалавр.

2. Тест:

Ви є кандидатом на вакантне місце. Пройдіть опитування нижче.

Яка основна різниця між HTML і XHTML?

☐ XHTML підтримує строгий синтаксис XML, тоді як HTML — це SGML-подібна мова.

☐ HTML може містити тільки теги, які відповідають стандарту HTML 4.01, тоді як XHTML може містити будь-які теги.

☐ HTML не потребує закриваючих тегів, тоді як XHTML вимагає, щоб кожен тег був правильно закритий.

☐ XHTML може містити скриптові файли, в той час як HTML — ні

Наступне питання

Ваш результат: 5 з 5

Показати правильні відповіді

Чекайте на повідомлення від нас!

Ваш результат: 5 з 5

Показати правильні відповіді

Правильні відповіді:

Яка основна різниця між HTML і XHTML?
 Ваша відповідь: XHTML підтримує строгий синтаксис XML, тоді як HTML — це SGML-подібна мова.
 Правильна відповідь: XHTML підтримує строгий синтаксис XML, тоді як HTML — це SGML-подібна мова.

Що таке бандлер у контексті JavaScript розробки?
 Ваша відповідь: Інструмент для об'єднання декількох JavaScript файлів в один.
 Правильна відповідь: Інструмент для об'єднання декількох JavaScript файлів в один.

Що таке props в React?
 Ваша відповідь: Об'єкт, який передає дані вниз по ієрархії компонентів у React.
 Правильна відповідь: Об'єкт, який передає дані вниз по ієрархії компонентів у React.

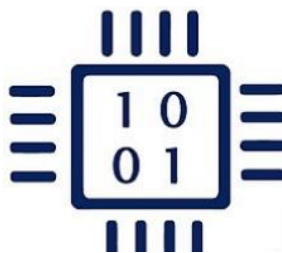
Які з наступних операторів не підтримуються у Python?
 Ваша відповідь: ++
 Правильна відповідь: ++

Яка з наступних опцій є коректним способом оголошення змінної в TypeScript?
 Ваша відповідь: const x: number = 5;
 Правильна відповідь: const x: number = 5;

Чекайте на повідомлення від нас!

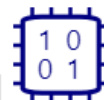
3. SVG-графіка:

Частина емблеми:



Повтори на сайті:

Created By Tishchenko Anna | All Rights Reserved



4. Гра:

 ToyJoy.

[Головна](#)

[Про Нас](#)

[Товари](#)

[Відгуки](#)

[Контакти](#)

[Місцезнаходження](#)

[Робота](#)

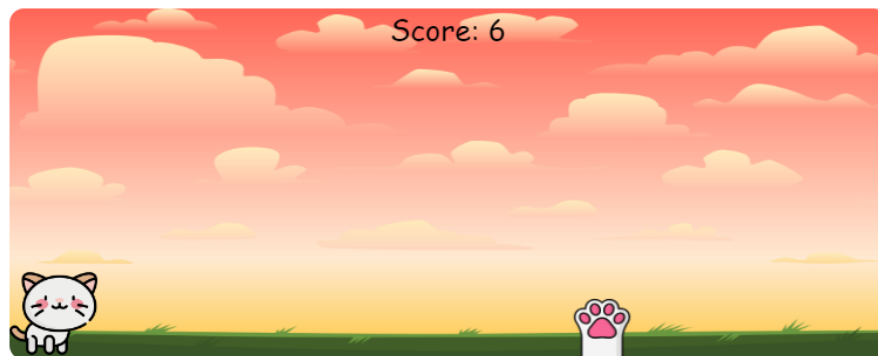
[Гра](#)



Кошення Та Лапа

Start

Кошення Та Лапа



Top 3 Scores:

1.Score: 1079

2.Score: 343

3.Score: 323

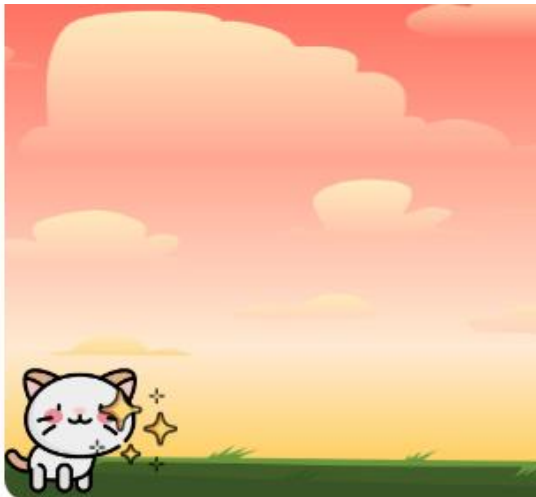
Bottom 3 Scores:

1.Score: 1

2.Score: 2

3.Score: 3

Суть гри перепригувати через лапи. Коли на екрані з'являється зірка, при контакті з нею додається +10 до балів.



Висновок: у ході лабораторної роботи були вивчені і практично застосовані основні принципи верстки веб-сторінок з використанням HTML та HTML5 форм, а також робота з растровою та векторною графікою на frontend. Процес розробки простої гри сприяв кращому розумінню і вмінню застосовувати CSS для створення стильового оформлення та анімаційних ефектів. Крім того, було ознайомлено з технологіями зберігання даних локально за допомогою LocalStorage, а також вивчено використання JavaScript для взаємодії з DOM та обробки подій. Завдяки практичному виконанню кожного кроку розробки, була отримана важлива практична навичка веб-розробки, яка може бути успішно використана в майбутніх проектах.