

Создание интерактивной рефлексии – батарейка для подведения итогов урока с помощью нейросети Qween

Голубева Евгения Павловна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Цель данной статьи – создание интерактивной рефлексии - батарейка с помощью нейросети Qween. Для создания интерактивного сайта рефлексия - батарейка использовалась нейросеть Qween. В результате работы разработана интерактивная рефлексия – батарейка.

Ключевые слова: рефлексия, интерактив, нейросеть Qween.

Creating an interactive reflection - a battery using the Qween neural network

Golubeva Evgeniya Pavlovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create an interactive reflection - a battery using the Qween neural network. The Qween neural network was used to create an interactive reflection - a battery. As a result, an interactive reflection - a battery was developed.

Keywords: reflection, interactivity, Qween neural network.

1 Введение

1.1 Актуальность

В условиях реализации требований ФГОС и перехода к личностно-ориентированному обучению особое значение приобретает не только процесс передачи знаний, но и этап рефлексии, позволяющий ученику осознать собственный прогресс и затруднения. В любом классе, на фоне активного становления учебной самостоятельности, традиционные формы подведения итогов урока «смайлики», «лесенка успеха» или устное обсуждение зачастую носят формальный характер: школьники механически отмечают уровень «всё понял» без глубинного анализа, а учитель тратит до 10–15 минут на сбор обратной связи, не получая при этом структурированной диагностики по каждому ученику.

Методика интерактивной рефлексии «Батарейка», где каждый обучающийся визуально оценивает собственные результаты по шкале «уровень заряда» от 0 до 100%, позволяет превратить итоговый этап урока в личностно-значимую процедуру. Однако создание такого дидактического инструмента вручную сопряжено с трудностями: необходимо разработать

критерии самооценки, адаптированные под конкретные учебные задачи: например, понимание алгоритма, скорость счета, умение объяснить товарищу, обеспечить интерактивность и мгновенную визуализацию для всего класса. Это требует от учителя либо глубоких навыков программирования, либо значительных временных затрат на создание презентаций с макросами.

Нейросеть Qween, специализирующаяся на генерации интерактивных учебных сценариев с элементами визуализации и адаптивной логикой, представляет собой оптимальное решение. С помощью Qween педагог может за 5–7 минут сгенерировать полностью готовую цифровую рефлексию «Батарейка»: от формулировки вопросов для самоанализа по таксономии Блума до динамической шкалы заряда, которая меняется при клике ученика.

Таким образом, разработка методики создания интерактивной рефлексии «Батарейка» для подведения итогов урока в классе с помощью нейросети Qween является актуальной и практически значимой задачей. Проект направлен на сокращение временных затрат учителя при организации обратной связи, повышение осознанности учебной деятельности у школьников и формирование навыков адекватной самооценки, что в конечном счёте способствует росту учебной мотивации и качеству усвоения материала.

1.2 Обзор исследований

Ю.А. Макковеева посвятила статью анализу потенциала использования нейросетей в процессе обучения аудированию [1]. Продемонстрировала использование современных инструментов на основе искусственного интеллекта для создания дидактических материалов Г.Р. Водяненко [2]. М.Н.Малинова описала варианты использования нейросетей повышающие интенсивность обучения, создающие более эффективный, персонализированный и увлекательный процесс обучения, который подготовит учеников к успеху в будущем. [3]. В статье рассматривала современные подходы к использованию нейросетевых технологий в образовательном процессе, с акцентом на предмет математики. Также проанализировала возможности, которые предоставляют нейросети для учителей, включая автоматизацию рутинных задач, создание индивидуализированных учебных планов и поддержку в оценивании знаний учащихся И.В. Вакатова [4]. Н.Е. Иванова подготовила таблицу с сервисами для создания образовательного контента средствами нейросетей российских разработчиков [5].

1.3 Цель исследования

Цель исследования – создание интерактивной рефлексии - батарейка с помощью нейросети Qween.

2 Материалы и методы

Для разработки интерактивной рефлексии в формате «батарейка», используемой для подведения итогов урока, обращение к нейросети Qween стало оптимальным решением. Выбор в пользу Qween обусловлен её

уникальной способностью быстро генерировать визуально понятные и легко настраиваемые интерактивные элементы, что особенно ценно в образовательной среде, где время педагога ограничено. Благодаря интуитивно понятному интерфейсу и готовым шаблонам, учитель без глубоких технических знаний может за считанные минуты создать яркий метафорический инструмент: уровень «заряда» батарейки наглядно демонстрирует, насколько усвоен материал, а анимированные элементы позволяют учащимся в игровой форме оценить свою работу на занятии. Qween автоматизирует рутинную часть разработки (раскладку кнопок, масштабирование, цветовую индикацию), освобождая педагога для творческого наполнения: достаточно текстового запроса, чтобы нейросеть предложила несколько вариантов дизайна и логики взаимодействия. Такой подход не только экономит время, но и снижает порог вхождения в цифровые инструменты, делая современную рефлексию доступной, персонализированной и эффективной — без необходимости изучать программирование или сложные конструкторы.

3 Результаты

Для того, чтобы приступить к созданию интерактивной рефлексии батарейка необходимо зарегистрироваться на платформе Qween. Войти в нейросеть можно через Google или Github аккаунт или создать аккаунт в нейросети (см. рис. 1).

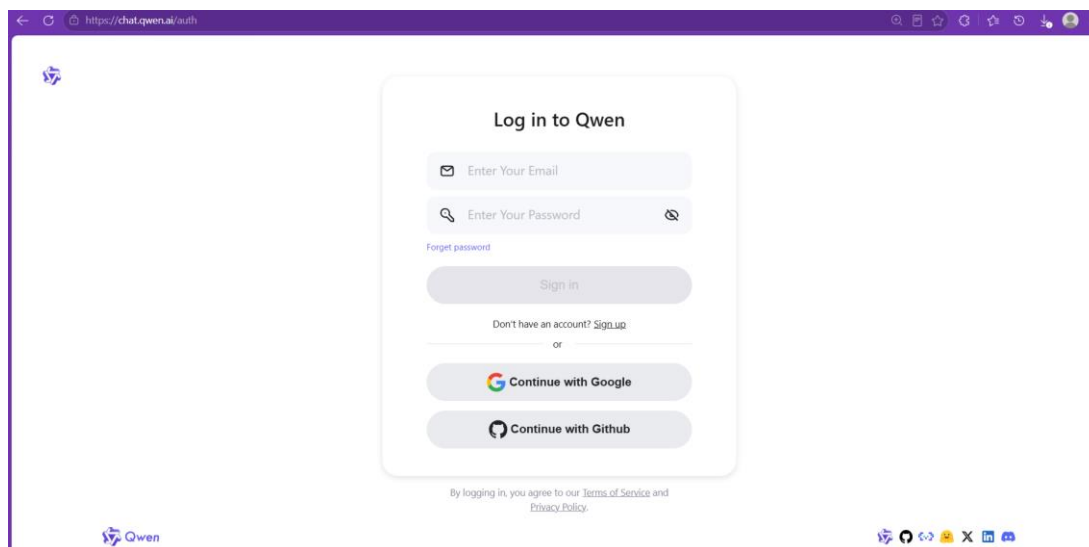


Рисунок 1 – Вход в нейросеть Qween

После регистрации, открывается главная страница нейросети. На главной странице нейросети Qween находится строка для ввода промта, также есть и голосовой ввод запроса (см. рис. 2).

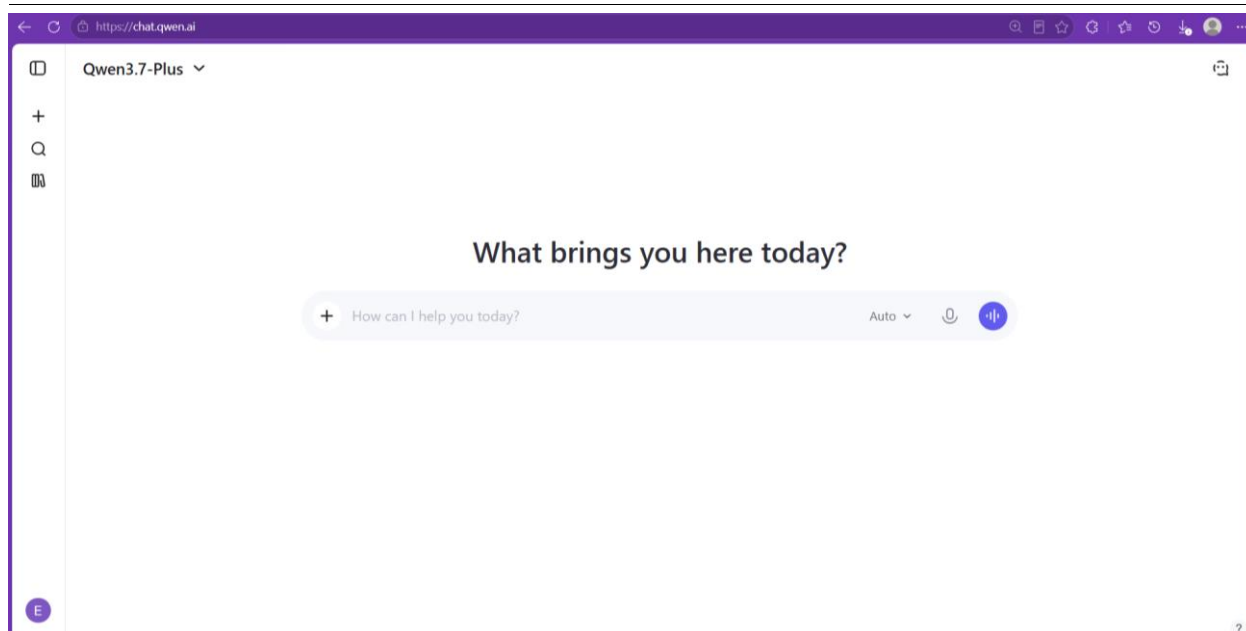


Рисунок 2 – Главная страница нейросети Qween

Для создания интерактивной рефлексии – батарейка для подведения итогов урока необходимо написать промт. Промт должен начинаться с чёткого указания роли (опытный Front-end разработчик) и формата результата (единый HTML-файл). Затем следует детально описать визуальную архитектуру: справа – крупная стилизованная иконка батарейки с внутренней сеткой для цветных элементов, слева – три большие кнопки с конкретными подписями и цветовыми темами (зелёная, жёлто-оранжевая, красная). Механика интерактива формулируется пошагово: при нажатии на любую из кнопок должен добавляться цветной «элемент» (зелёный / оранжевый / красный) в следующую свободную ячейку батарейки, визуальное имитируя падение и закрепление; общее количество ячеек задаёт шкалу 0–100%. Одновременно ведётся подсчёт нажатий на каждую кнопку, отображаемый рядом с ней. Батарейка заполняется пропорционально – по мере добавления элементов пользователь видит соотношение цветов, отражающее уровень понимания. Дополнительно нужно указать требования к анимациям (плавное появление элементов, переходы, эффект «энергии» с помощью градиентов), адаптивности (крупные элементы для сенсорных экранов) и чистоте кода (без внешних библиотек, семантическая разметка, CSS Grid / Flexbox). Завершить промт следует просьбой реализовать всю логику на JavaScript с обработкой предельного заполнения батарейки и возможностью сброса.

Готовый промт для интерактивной рефлексии – батарейка для подведения итогов урока: Ты опытный Front-End разработчик. Напиши код для интерактивной HTML-игры рефлексии для урока. Все должно быть в одном файле (HTML + CSS + JS). Визуальная часть: С правой стороны экрана должна быть большая, стилизованная иконка батарейки (Battery Icon) с анимацией заполнения. С левой стороны от батарейки расположены три большие кнопки с подписями: Кнопка 1: "Всё понял / Всё получилось" (Зеленая тема). Кнопка 2: "Есть вопросы / Не всё понятно" (Желтая/ Оранжевая

тема). Кнопка 3: "Ничего не понял / Было сложно" (Красная тема). Логика работы: Батарейка имеет шкалу от 0% до 100%. При нажатии на Кнопку 1: выпадают зеленые элементы и остаются в батареечке. Если средний - оранжевым. При нажатии на Кнопку 2: Выпадают желтые элементы и также остаются в батареечке. При нажатии на Кнопку 3: Выпадают красные маленькие элементы и также занимают свое место. При нажатии кнопок постепенно батареечка заполняется разными цветами и визуально видно, сколько человек усвоило хорошо, сколько средне, а сколько плохо. также на экране показывается сколько человек нажало на кнопку 1, 2 и 3. Требования: Используй современные градиенты для эффекта энергии внутри батареечки. Добавь плавную анимацию (transition) при изменении цвета и уровня заряда. Сделай дизайн адаптивным и крупным, чтобы было удобно нажимать на интерактивной доске или планшете. Код должен быть чистым и готовым к запуску в браузере (см. рис. 3).

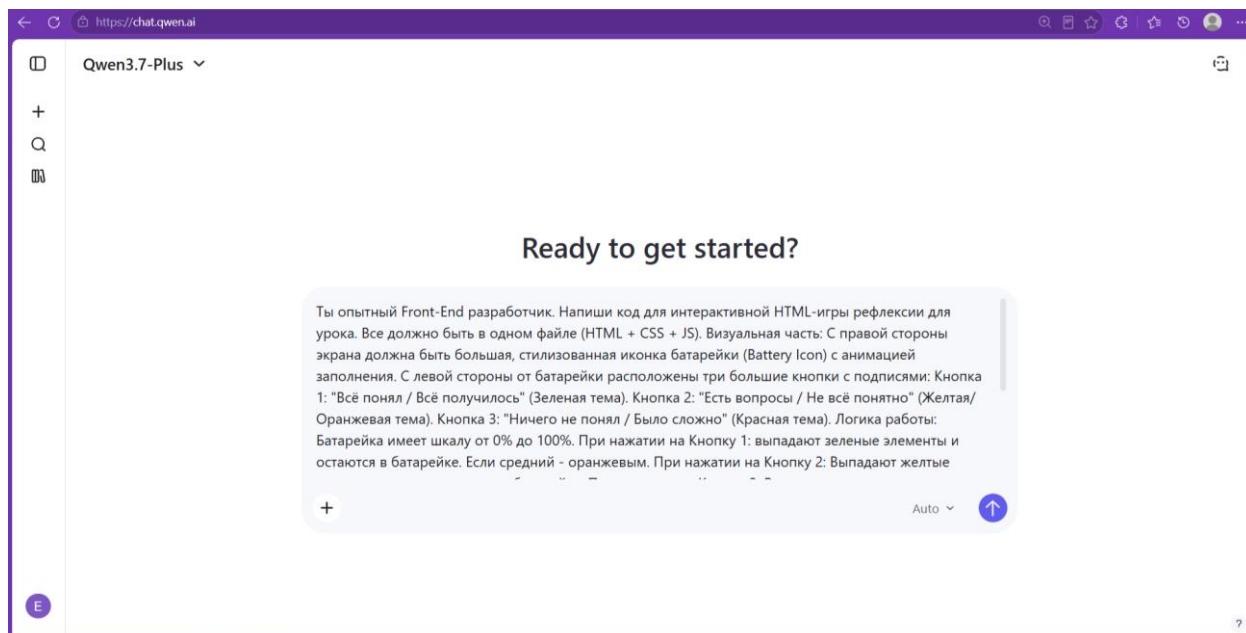


Рисунок 3 – Новый чат

После отправления запроса, нейросеть предоставляет готовый код (см. рис. 4).

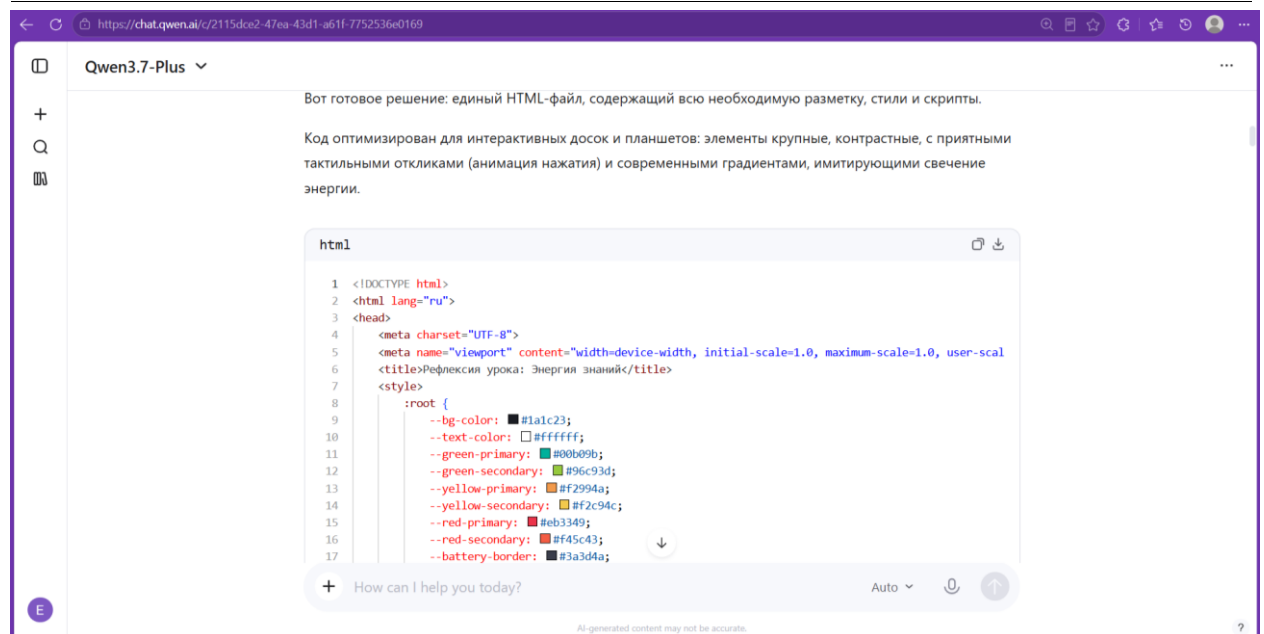


Рисунок 4 – Процесс создания интерактивной рефлексии – батарейка для подведения итогов урока

Готовый код, предоставленный нейросетью Qween:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=1.0, user-scalable=no">
  <title>Рефлексия урока: Энергия знаний</title>
  <style>
    :root {
      --bg-color: #1a1c23;
      --text-color: #ffffff;
      --green-primary: #00b09b;
      --green-secondary: #96c93d;
      --yellow-primary: #f2994a;
      --yellow-secondary: #f2c94c;
      --red-primary: #eb3349;
      --red-secondary: #f45c43;
      --battery-border: #3a3d4a;
      --battery-bg: #252833;
    }

    * {
      box-sizing: border-box;
      margin: 0;
      padding: 0;
      font-family: 'Segoe UI', Roboto, Helvetica, Arial, sans-serif;
      -webkit-tap-highlight-color: transparent;
    }

    body {
      background-color: var(--bg-color);
      color: var(--text-color);
      height: 100vh;
      display: flex;
      flex-direction: column;
      overflow: hidden;
    }

    /* Основной контейнер */
    .main-container {
```

```
        display: flex;
        flex: 1;
        width: 100%;
        max-width: 1400px;
        margin: 0 auto;
        padding: 20px;
        gap: 40px;
        align-items: center;
        justify-content: center;
    }

    /* Левая панель с кнопками */
    .buttons-panel {
        flex: 1;
        display: flex;
        flex-direction: column;
        gap: 20px;
        width: 100%;
        max-width: 500px;
    }

    .reflex-btn {
        border: none;
        border-radius: 20px;
        padding: 30px 20px;
        font-size: 1.5rem;
        font-weight: 700;
        color: white;
        cursor: pointer;
        transition: all 0.2s cubic-bezier(0.25, 0.46, 0.45, 0.94);
        display: flex;
        align-items: center;
        justify-content: space-between;
        box-shadow: 0 8px 20px rgba(0,0,0,0.3);
        position: relative;
        overflow: hidden;
    }

    .reflex-btn:active {
        transform: scale(0.96);
    }

    .reflex-btn::after {
        content: '';
        position: absolute;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
        background: linear-gradient(135deg, rgba(255,255,255,0.2) 0%,
        rgba(255,255,255,0) 100%);
        pointer-events: none;
    }

    .btn-green {
        background: linear-gradient(135deg, var(--green-primary), var(--green-secondary));
        box-shadow: 0 8px 25px rgba(0, 176, 155, 0.4);
    }
    .btn-green:hover { box-shadow: 0 12px 30px rgba(0, 176, 155, 0.6); transform: translateY(-2px); }

    .btn-yellow {
        background: linear-gradient(135deg, var(--yellow-primary), var(--yellow-secondary));
        box-shadow: 0 8px 25px rgba(242, 153, 74, 0.4);
    }
    .btn-yellow:hover { box-shadow: 0 12px 30px rgba(242, 153, 74, 0.6); transform: translateY(-2px); }

    .btn-red {
```

```
background: linear-gradient(135deg, var(--red-primary), var(--red-secondary));
box-shadow: 0 8px 25px rgba(235, 51, 73, 0.4);
}
.btn-red:hover { box-shadow: 0 12px 30px rgba(235, 51, 73, 0.6); transform:
translateY(-2px); }

.btn-icon {
  font-size: 2.5rem;
}

/* Правая панель с батареей */
.battery-panel {
  flex: 1;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  width: 100%;
  max-width: 400px;
}

.battery-outer {
  width: 100%;
  max-width: 220px;
  height: 400px;
  border: 8px solid var(--battery-border);
  border-radius: 30px;
  background-color: var(--battery-bg);
  position: relative;
  padding: 10px;
  box-shadow:
    inset 0 0 30px rgba(0,0,0,0.5),
    0 0 40px rgba(0,0,0,0.3);
  display: flex;
  flex-direction: column-reverse; /* Заполнение снизу вверх */
}

/* Контакт батарейки сверху */
.battery-outer::before {
  content: '';
  position: absolute;
  top: -20px;
  left: 50%;
  transform: translateX(-50%);
  width: 80px;
  height: 20px;
  background-color: var(--battery-border);
  border-radius: 10px 10px 0 0;
}

.battery-inner {
  width: 100%;
  height: 100%;
  border-radius: 18px;
  overflow: hidden;
  display: flex;
  flex-direction: column-reverse;
  position: relative;
}

/* Сегменты заполнения */
.fill-segment {
  width: 100%;
  height: 0%; /* Будет изменяться через JS */
  transition: height 0.6s cubic-bezier(0.34, 1.56, 0.64, 1);
  position: relative;
}

.fill-green {
```

```

background: linear-gradient(to top, var(--green-primary), var(--green-
secondary));
box-shadow: 0 0 20px var(--green-primary);
}
.fill-yellow {
background: linear-gradient(to top, var(--yellow-primary), var(--yellow-
secondary));
box-shadow: 0 0 20px var(--yellow-primary);
}
.fill-red {
background: linear-gradient(to top, var(--red-primary), var(--red-
secondary));
box-shadow: 0 0 20px var(--red-primary);
}

/* Эффект пузырьков/энергии внутри */
.fill-segment::after {
content: '';
position: absolute;
top: 0;
left: 0;
right: 0;
bottom: 0;
background-image: radial-gradient(circle, rgba(255,255,255,0.3) 2px,
transparent 2px);
background-size: 15px 15px;
opacity: 0.5;
animation: energyPulse 2s infinite alternate;
}

@keyframes energyPulse {
0% { opacity: 0.3; transform: translateY(0); }
100% { opacity: 0.6; transform: translateY(-5px); }
}

/* Статистика */
.stats-container {
margin-top: 30px;
width: 100%;
background: rgba(255, 255, 255, 0.05);
border-radius: 15px;
padding: 20px;
backdrop-filter: blur(10px);
}

.stat-row {
display: flex;
justify-content: space-between;
align-items: center;
padding: 10px 0;
border-bottom: 1px solid rgba(255,255,255,0.1);
font-size: 1.2rem;
}

.stat-row:last-child {
border-bottom: none;
}

.stat-label {
display: flex;
align-items: center;
gap: 10px;
}

.stat-dot {
width: 16px;
height: 16px;
border-radius: 50%;
}

.stat-value {

```

```
        font-weight: bold;
        font-size: 1.4rem;
    }

    .total-count {
        text-align: center;
        margin-top: 15px;
        font-size: 1.1rem;
        color: #8b8e99;
        text-transform: uppercase;
        letter-spacing: 1px;
    }

    /* Анимация падающих частиц */
    .particle {
        position: fixed;
        width: 20px;
        height: 20px;
        border-radius: 50%;
        pointer-events: none;
        z-index: 9999;
        transition: all 0.8s cubic-bezier(0.25, 0.46, 0.45, 0.94);
        box-shadow: 0 0 10px currentColor;
    }

    /* Адаптивность для мобильных устройств и вертикальных досок */
    @media (max-width: 900px) {
        .main-container {
            flex-direction: column;
            gap: 20px;
            padding: 15px;
        }

        .buttons-panel {
            max-width: 100%;
            flex-direction: row;
            flex-wrap: wrap;
        }

        .reflex-btn {
            flex: 1;
            min-width: 140px;
            padding: 20px 10px;
            font-size: 1.1rem;
            flex-direction: column;
            text-align: center;
            gap: 10px;
        }

        .btn-icon {
            font-size: 2rem;
        }

        .battery-outer {
            height: 250px;
            max-width: 140px;
        }

        .stats-container {
            padding: 15px;
        }

        .stat-row {
            font-size: 1rem;
        }
    }
</style>
</head>
<body>

    <div class="main-container">
```

```

    <!-- Левая часть: Кнопки -->
    <div class="buttons-panel">
        <button class="reflex-btn btn-green" onclick="handleVote(1, this)" id="btn-
1">
            <span>Всё понял / Всё получилось</span>
            <span class="btn-icon">😊</span>
        </button>

        <button class="reflex-btn btn-yellow" onclick="handleVote(2, this)"
id="btn-2">
            <span>Есть вопросы / Не всё понятно</span>
            <span class="btn-icon">🤔</span>
        </button>

        <button class="reflex-btn btn-red" onclick="handleVote(3, this)" id="btn-
3">
            <span>Ничего не понял / Было сложно</span>
            <span class="btn-icon">😞</span>
        </button>
    </div>

    <!-- Правая часть: Батарейка и статистика -->
    <div class="battery-panel">
        <div class="battery-outer" id="battery">
            <div class="battery-inner">
                <div class="fill-segment fill-green" id="fill-green"></div>
                <div class="fill-segment fill-yellow" id="fill-yellow"></div>
                <div class="fill-segment fill-red" id="fill-red"></div>
            </div>
        </div>

        <div class="stats-container">
            <div class="stat-row">
                <div class="stat-label">
                    <div class="stat-dot" style="background: var(--green-
secondary);"></div>
                    <span>Всё понял</span>
                </div>
                <div class="stat-value" id="count-1">0</div>
            </div>
            <div class="stat-row">
                <div class="stat-label">
                    <div class="stat-dot" style="background: var(--yellow-
secondary);"></div>
                    <span>Есть вопросы</span>
                </div>
                <div class="stat-value" id="count-2">0</div>
            </div>
            <div class="stat-row">
                <div class="stat-label">
                    <div class="stat-dot" style="background: var(--red-
secondary);"></div>
                    <span>Было сложно</span>
                </div>
                <div class="stat-value" id="count-3">0</div>
            </div>
            <div class="total-count">
                Всего проголосовало: <span id="total-count">0</span>
            </div>
        </div>
    </div>

    <script>
        // Состояние приложения
        const state = {
            count1: 0,
            count2: 0,
            count3: 0
        };
    </script>

```

```

// Цвета для частиц
const colors = {
  1: '#96c93d', // Зеленый
  2: '#f2c94c', // Желтый
  3: '#f45c43'  // Красный
};

/**
 * Обработчик нажатия на кнопку
 * @param {number} type - 1, 2 или 3
 * @param {HTMLElement} btnElement - элемент кнопки для анимации
 */
function handleVote(type, btnElement) {
  // 1. Обновляем счетчики
  if (type === 1) state.count1++;
  if (type === 2) state.count2++;
  if (type === 3) state.count3++;

  const total = state.count1 + state.count2 + state.count3;

  // 2. Обновляем текст статистики
  document.getElementById('count-1').textContent = state.count1;
  document.getElementById('count-2').textContent = state.count2;
  document.getElementById('count-3').textContent = state.count3;
  document.getElementById('total-count').textContent = total;

  // 3. Рассчитываем проценты для батарейки
  const p1 = total > 0 ? (state.count1 / total) * 100 : 0;
  const p2 = total > 0 ? (state.count2 / total) * 100 : 0;
  const p3 = total > 0 ? (state.count3 / total) * 100 : 0;

  // 4. Применяем высоту к сегментам батарейки
  document.getElementById('fill-green').style.height = `${p1}%`;
  document.getElementById('fill-yellow').style.height = `${p2}%`;
  document.getElementById('fill-red').style.height = `${p3}%`;

  // 5. Запускаем анимацию падающих элементов
  spawnParticles(type, btnElement);
}

/**
 * Создает анимацию падающих частиц от кнопки к батарейке
 */
function spawnParticles(type, btnElement) {
  const battery = document.getElementById('battery');
  const btnRect = btnElement.getBoundingClientRect();
  const batRect = battery.getBoundingClientRect();

  const color = colors[type];
  const particleCount = 8; // Количество частиц за одно нажатие

  for (let i = 0; i < particleCount; i++) {
    const particle = document.createElement('div');
    particle.className = 'particle';
    particle.style.backgroundColor = color;
    particle.style.color = color; // Для box-shadow: currentColor

    // Начальная позиция (центр кнопки с небольшим разбросом)
    const startX = btnRect.left + btnRect.width / 2 + (Math.random() * 60 -
30);

    const startY = btnRect.top + btnRect.height / 2;

    particle.style.left = `${startX}px`;
    particle.style.top = `${startY}px`;
    particle.style.transform = 'scale(1)';
    particle.style.opacity = '1';

    document.body.appendChild(particle);

    // Форсируем перерисовку для запуска transition

```

```

particle.offsetHeight;

// Конечная позиция (случайная точка внутри батарейки)
// Ограничиваем область, чтобы частицы летели примерно в центр батарейки
const endX = batRect.left + batRect.width / 2 + (Math.random() * 80 -
40);

const endY = batRect.top + batRect.height / 2 + (Math.random() * 100 -
50);

// Запускаем анимацию
requestAnimationFrame(() => {
  particle.style.transform = `translate(${endX - startX}px, ${endY -
startY}px) scale(0.2)`;
  particle.style.opacity = '0';
});

// Удаляем элемент из DOM после завершения анимации
setTimeout(() => {
  particle.remove();
}, 800);
}

// Добавляем эффект "нажатия" для тач-устройств (убирает задержку 300мс)
document.querySelectorAll('.reflex-btn').forEach(btn => {
  btn.addEventListener('touchstart', function() {}, {passive: true});
});
</script>
</body>
</html>

```

Готовый код необходимо скачать. Для этого необходимо нажать на кнопку скачивания, и код скачается в формате HTML (см.рис.5).

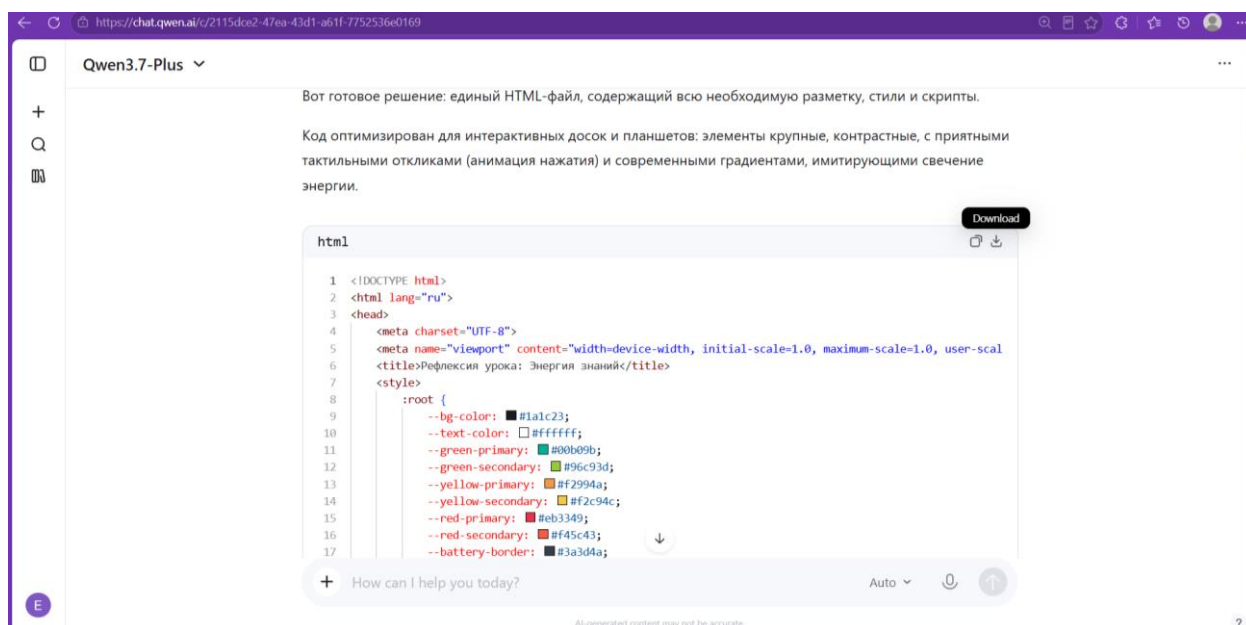


Рисунок 5 – Код

После того, как файл скачался, необходимо открыть в браузере. Готовая интерактивная рефлексия – батарейка для подведения итогов урока (см.рис.6).

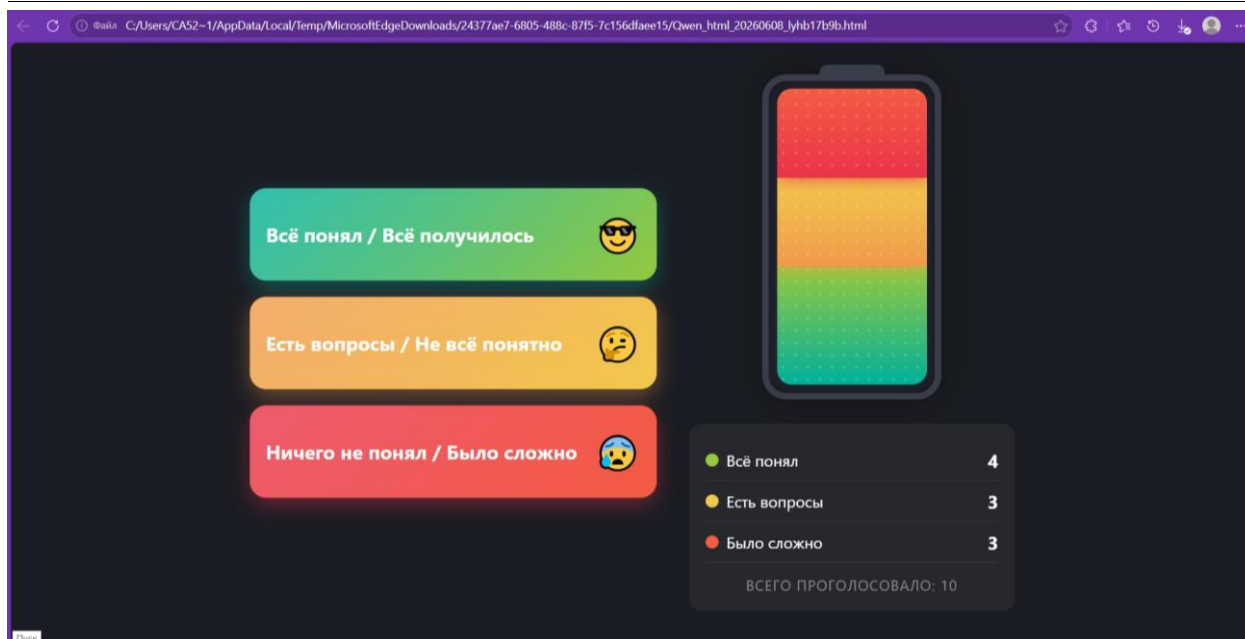


Рисунок 6 – Готовый результат

4. Выводы

В данной работе показан процесс создания интерактивной рефлексии – батарейка для подведения итогов урока с помощью нейросети Qween. Научная статья может служить методическим пособием в учебной деятельности.

Библиографический список

1. Ширяева Ю. Н. Анализ нейросетей, возможных для использования учителями при обучении аудированию на уроках иностранного языка // Научное наставничество в лингвистике, лингводидактике и межкультурной коммуникации: Материалы Всероссийской научно-практической конференции, Магнитогорск, 18 апреля 2025 года. Магнитогорск: Магнитогорский государственный технический университет им. Г.И. Носова, 2025. С. 159-163.
2. Водяненко Г. Р. Инструменты с искусственным интеллектом в работе педагога // Интерактивная наука. 2023. №. 8 (84). С. 21-24.
3. Малинова М. Н. Нейросети: революционный инструмент для учителя технологии в ххi веке // Формирование и развитие новой парадигмы науки в эпоху. 2025. С. 79.
4. Вакатова И. В. Нейросети в помощь учителю математики: инновационные инструменты для современного образования // Цифровые технологии и инновации в развитии науки и образования: сборник статей по материалам Всероссийской научно-практической конференции с международным участием, Чебоксары, 11 апреля 2025 года. Чебоксары: Чувашский государственный педагогический университет им. И.Я. Яковлева, 2025. С. 46-49.
5. Иванова Н. Е. Разработка цифрового образовательного контента учителями начальных классов с помощью нейросетей // Образование в цифровую

эпоху: опыт, проблемы и перспективы: сборник статей по материалам Международной научно-практической конференции преподавателей, студентов, аспирантов, докторантов и заинтересованных лиц. В 2-х томах, Нижний Новгород, 17–18 декабря 2025 года. Том I. Нижний Новгород: ИП Гладкова О.В., 2025. С. 55-59.