Assignment 7 JQuery
Kuanysh Beibarys SE-2413

Task 1 — Real-time Search Filter:
Implemented a live search function that filters video cards as the user types in the search box.

```
// Task 1 – Real-time search filter
$("#search").on("keyup", function () {
  let value = $(this).val().toLowerCase();
  $(".video-card").filter(function () {
    $(this).toggle($(this).text().toLowerCase().includes(value));
  });
});
```

Task 2 — Autocomplete Search Suggestions:
Added an autocomplete dropdown that displays up to 5 matching video titles while typing.

```
// Task 2 – Autocomplete search suggestions
const titles = $(".video-title")
  .map(function () {
    return $(this).text();
  })
  .get();

$("#search").on("input", function () {
  let input = $(this).val().toLowerCase();
  $("#suggestions").remove(); // очищаем старые подсказки
  if (input.length > 0) {
    let matches = titles.filter((t) => t.toLowerCase().includes(input));
    let ul = $("<ul id='suggestions' class='suggestions'></ul>");
    matches.slice(0, 5).forEach((m) => ul.append(`<li>${m}</li>`));
    $(this).after(ul);
  }
});

$(document).on("click", "#suggestions li", function () {
  $("#search").val($(this).text());
  $("#suggestions").remove();
});

$(document).click(function (e) {
  if (!$(e.target).closest("#search, #suggestions").length) {
    $("#suggestions").remove();
  }
});
```

Task 3 — Highlight Matching Words:
Implemented keyword highlighting so that the searched text appears in yellow within video titles.

```
// Task 3 – Highlight matching words
$("#search").on("input", function () {
  let keyword = $(this).val();
  $(".video-title").each(function () {
    let text = $(this).text();
    if (keyword.length > 0) {
      let regex = new RegExp(`(${keyword})`, "gi");
      $(this).html(text.replace(regex, "<span class='highlight'>$1</span>"));
    } else {
      $(this).text(text);
    }
  });
});

```

Task 4 — Scroll Progress Bar:
Created a progress bar at the top of the page that visually indicates how far the user has scrolled.

```
// Task 4 – Scroll progress bar
$("body").prepend('<div id="progressBar"></div>');
$(window).on("scroll", function () {
  let scroll = $(window).scrollTop();
  let height = $(document).height() - $(window).height();
  let scrolled = (scroll / height) * 100;
  $("#progressBar").css("width", scrolled + "%");
});

```

Task 5 — Animated Counter:
Added a smooth counting animation for numeric elements (useful for statistics or analytics).

```
// Task 5 — Animated counter (пример для будущих секций)
$(".counter").each(function () {
  let $this = $(this),
    countTo = $this.attr("data-target");
  $({ countNum: $this.text() }).animate(
    { countNum: countTo },
    {
      duration: 2000,
      easing: "linear",
      step: function () {
        $this.text(Math.floor(this.countNum));
      },
      complete: function () {
        $this.text(this.countNum + "+");
      },
    }
  );
});

```

Task 6 — Loading Spinner on Submit:
The form submit button now shows “Please wait…” and becomes disabled temporarily to simulate loading.

```
// Task 6 — Loading spinner on form submit
$("form").on("submit", function (e) {
  e.preventDefault();
  let btn = $(this).find("button[type='submit']");
  btn.prop("disabled", true).html("⏳ Please wait...");
  setTimeout(() => {
    btn.prop("disabled", false).html("Submit");
    showToast("✅ Form submitted successfully!");
  }, 2000);
});

```

Task 7 — Notification System (Toast):
Developed a toast notification system that displays messages for actions like liking, unliking, or form submission.

```
// Task 7 – Notification system (Toast)
$("body").append('<div id="toast"></div>');

function showToast(message, type = "info") {
  const toast = $("#toast");
  toast.stop(true, true);
  toast.text(message);

  if (type === "success") toast.css("background-color", "#28a745");
  else if (type === "error") toast.css("background-color", "#dc3545");
  else if (type === "info") toast.css("background-color", "#007bff");
  else toast.css("background-color", "#333");

  toast.addClass("show");

  setTimeout(() => toast.removeClass("show"), 2500);
}
```

Task 8 — Like Button with Counter:
Added a ❤️ like button for each video with a random initial count and pulse animation on click.

```
// Task 8 – Like button with random counter
$(".video-card").each(function () {

  let randomLikes = Math.floor(Math.random() * 10000);
  $(this).append(`
    <div class="like-container">
      <button class="like-btn">♡</button>
      <span class="like-count">${randomLikes}</span>
    </div>
  `);
});
                                    (local function)(this: any): void
$(document).on("click", ".like-btn", function () {
  let $btn = $(this);
  let $count = $btn.siblings(".like-count");
  let likes = parseInt($count.text());
  let videoTitle = $btn.closest(".video-card").find(".video-title").text();

  if ($btn.hasClass("liked")) {
    $btn.removeClass("liked").text("♡");
    $count.text(likes - 1);
    showToast(`💔 You unliked "${videoTitle}"`, "error");
  } else {
    $btn.addClass("liked").text("❤️");
    $count.text(likes + 1);
    $btn.animate({ fontSize: "26px" }, 150).animate({ fontSize: "20px" }, 150);
    showToast(`❤️ You liked "${videoTitle}"`, "success");
  }
});
```

Task 9 — Lazy Image Loading:
Images are loaded only when visible in the viewport, improving page performance and loading speed.

```
// Task 9 – Lazy image loading
$("img").each(function () {
  let src = $(this).attr("src");
  $(this).attr("data-src", src).removeAttr("src");
});

$(window).on("scroll", function () {
  $("img[data-src]").each(function () {
    if ($(this).offset().top < $(window).scrollTop() + $(window).height()) {
      $(this).attr("src", $(this).attr("data-src"));
      $(this).removeAttr("data-src");
    }
  });
});

// Initial check (load visible images)
$(window).trigger("scroll");
});
```