Таймер

```
import flatpickr from "flatpickr";
import "flatpickr/dist/flatpickr.min.css";
import iziToast from "izitoast";
import "izitoast/dist/css/iziToast.min.css";
import { convertMs, addLeadingZero } from "./utils.js";
import "./timer.css";

const refs = {
    dateTimePicker: document.querySelector("#datetime-picker"),
    startButton: document.querySelector("[data-start]"),
    days: document.querySelector("[data-days]"),
    hours: document.querySelector("[data-hours]"),
    minutes: document.querySelector("[data-minutes]"),
    seconds: document.querySelector("[data-seconds]"),
};

let userSelectedDate = null;
let timerId = null;
refs.startButton.setAttribute("disabled", true);

const options = {
    enableTime: true,
    time_24hr: true,
    defaultDate: new Date(),
    minuteIncrement: 1,
    onClose(selectedDates) {
        const selectedDate = selectedDates[0];
        if (selectedDate <= new Date()) {
            iziToast.error({
                title: 'Error',
                message: 'Illegal operation',
                backgroundColor: '#F44336',
                titleColor: '#ffffff',
                messageColor: '#ffffff',
                close: true,
                progressBar: true,
                progressBarColor: '#B51B1B',
                position: 'topRight',
                timeout: 50000,
                class: 'custom-error-toast',
            });
            refs.startButton.setAttribute("disabled", true);
            refs.dateTimePicker.removeAttribute("disabled");
        } else {
            userSelectedDate = selectedDate;
            refs.startButton.removeAttribute("disabled");
        }
    },
};

flatpickr(refs.dateTimePicker, options);
```

```
function startTimer() {
    if (!userSelectedDate) return;

    refs.startButton.setAttribute("disabled", true);
    refs.dateTimePicker.setAttribute("disabled", true);

    timerId = setInterval(() => {
        const timeLeft = userSelectedDate - new Date();
        if (timeLeft <= 0) {
            clearInterval(timerId);
            iziToast.success({
                title: "Finished",
                message: "Countdown reached zero!",
            });
            refs.startButton.setAttribute("disabled", true);
            refs.dateTimePicker.removeAttribute("disabled");
            updateTimer(0);
        } else {
            updateTimer(timeLeft);
        }
    }, 1000);
}

function updateTimer(ms) {
    const { days, hours, minutes, seconds } = convertMs(ms);
    refs.days.textContent = addLeadingZero(days);
    refs.hours.textContent = addLeadingZero(hours);
    refs.minutes.textContent = addLeadingZero(minutes);
    refs.seconds.textContent = addLeadingZero(seconds);
}

document.querySelector("[data-start]").addEventListener("click", startTimer);
```

Генератор промісів

```
import iziToast from "izitoast";
import "izitoast/dist/css/iziToast.min.css";
import "./snackbar.css";

const refs = {
    form: document.querySelector('.form'),
    delay: document.querySelector('[name="delay"]'),
    state: document.querySelectorAll('[name="state"]'),
    submit: document.querySelector(".submit-btn"),
}

function createPromise(delay, state) {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            if (state === 'fulfilled') {
                resolve({ delay });
```

```javascript
            } else {
                reject({ delay });
            }
        }, delay);
    });
}

function submitForm() {
    refs.form.addEventListener("submit", (event) => {
        event.preventDefault();
        const delay = Number(refs.delay.value);
        const state = Array.from(refs.state).find(input => input.checked)?.value;

        createPromise(delay, state)
        .then(({ delay }) => {
            iziToast.success({
                title: 'Success',
                message: `Fulfilled promise in ${delay}ms`,
                position: 'topRight',
                backgroundColor: '#59A10D',
                titleColor: '#ffffff',
                messageColor: '#ffffff',
                close: true,
                progressBar: true,
                progressBarColor: '#326101',
                position: 'topRight',
                timeout: delay,
                class: 'custom-success-toast',
            });
        })
        .catch(({ delay }) => {
            iziToast.error({
                title: 'Error',
                message: `Rejected promise in ${delay}ms`,
                backgroundColor: '#F44336',
                titleColor: '#ffffff',
                messageColor: '#ffffff',
                close: true,
                progressBar: true,
                progressBarColor: '#9C1C1C',
                position: 'topRight',
                timeout: delay,
                class: 'custom-error-toast',
            });
        });

        refs.form.reset();
    });
}

submitForm();
```

Пошук зображень

```
import iziToast from "izitoast";
import "izitoast/dist/css/iziToast.min.css";
import SimpleLightbox from "simplelightbox";
import "simplelightbox/dist/simple-lightbox.min.css";
import { fetch } from "./api.js";
import '../style.css';
import './search.css';

const refs = {
    form:  document.querySelector('.search-form'),
    gallery: document.querySelector('.gallery'),
    loader: document.querySelector('.loader'),
    loadMore: document.querySelector('.load-more'),
};

let lightbox;
let currentQuery = '';
let currentPage = 1;

refs.form.addEventListener('submit', onSearch);
refs.loadMore.addEventListener('click', onLoadMore);

async function onSearch(event) {
    event.preventDefault();

    const query = refs.form.querySelector('input').value.trim();
    currentPage = 1;
    currentQuery = query;

    if (query === '') {
        iziToast.error({
            title: 'Error',
            message: 'Please enter a search term!',
            backgroundColor: '#F44336',
            titleColor: '#ffffff',
            messageColor: '#ffffff',
            close: true,
            progressBar: true,
            progressBarColor: '#9C1C1C',
            position: 'topRight',
            timeout: 50000,
            class: 'custom-error-toast',
        });
        return;
    }

    refs.gallery.innerHTML = '';
    refs.loader.classList.remove('hidden');
    refs.loadMore.classList.add('hidden');

    try {
        const data = await fetch(currentQuery, currentPage);
        renderGallery(data.hits, true);

        if (data.totalHits > 0) {
            iziToast.success({
                title: 'Success',
                message: `Found ${data.totalHits} images for "${query}"`,
                position: 'topRight',
                backgroundColor: '#59A10D',
                titleColor: '#ffffff',
                messageColor: '#ffffff',
                close: true,
                progressBar: true,
                progressBarColor: '#326101',
                position: 'topRight',
                timeout: 50000,
                class: 'custom-success-toast',
            });

            if (data.totalHits > data.hits.length) {
                refs.loadMore.classList.remove('hidden');
            }
        } else {
            iziToast.error({
                title: 'Error',
                message: `No images found for "${query}"`,
                position: 'topRight',
                backgroundColor: '#F44336',
                titleColor: '#ffffff',
                messageColor: '#ffffff',
                close: true,
                progressBar: true,
                progressBarColor: '#9C1C1C',
                position: 'topRight',
                timeout: 50000,
                class: 'custom-error-toast',
            });
        }
    } catch (error) {
        iziToast.error({
            title: 'Error',
            message: 'Something went wrong. Please try again!',
            backgroundColor: '#F44336',
            titleColor: '#ffffff',
            messageColor: '#ffffff',
            close: true,
            progressBar: true,
            progressBarColor: '#9C1C1C',
            position: 'topRight',
            timeout: 50000,
            class: 'custom-error-toast',
        });
    } finally {
        refs.loader.classList.add('hidden');
    }

    refs.form.reset();
}

async function onLoadMore() {
    currentPage += 1;
    refs.loader.classList.remove('hidden');
    refs.loadMore.classList.add('hidden');

    try {
        const data = await fetch(currentQuery, currentPage);
        renderGallery(data.hits, false);

        if (data.hits.length > 0) {
            refs.loadMore.classList.remove('hidden');
        }
    } catch (error) {
        iziToast.error({
            title: 'Error',
            message: 'Something went wrong. Please try again!',
            backgroundColor: '#F44336',
            titleColor: '#ffffff',
            messageColor: '#ffffff',
            close: true,
            progressBar: true,
            progressBarColor: '#9C1C1C',
            position: 'topRight',
            timeout: 50000,
            class: 'custom-error-toast',
        });
    } finally {
        refs.loader.classList.add('hidden');
    }
}

function renderGallery(images, replace = false) {
    const markup = images
        .map(image => {
            return `
            <li class="gallery-item">
                <a href="${image.largeImageURL}">
                    <img class="gallery-img img" src="${image.webformatURL}"
alt="${image.tags}" loading="lazy" />
                </a>
                <div class="info">
                    <div class="info-details">
                        <p class="info-title">Likes</p>
                        <p>${image.likes}</p>
                    </div>
                    <div class="info-details">
                        <p class="info-title">Views</p>
                        <p>${image.views}</p>
                    </div>
                    <div class="info-details">
                        <p class="info-title">Comments</p>
                        <p>${image.comments}</p>
                    </div>
                    <div class="info-details">
                        <p class="info-title">Downloads</p>
                        <p>${image.downloads}</p>
                    </div>
                </div>
            </li>
            `;
        })
        .join('');

    if (replace) {
        refs.gallery.innerHTML = markup;
    } else {
        refs.gallery.insertAdjacentHTML('beforeend', markup);
    }

    if (lightbox) {
        lightbox.refresh();
    } else {
        lightbox = new SimpleLightbox('.gallery a', {
            captionsData: 'alt',
            captionDelay: 250,
        });
    }
}
```

```
import axios from 'axios';

const fetch = async (query, page) => {
    const accessKey = "31440578-d40e7eed5873a4f1028e16656";
    const response = await axios.get(
        `https://pixabay.com/api/`, {
        params: {
            key: accessKey,
            q: query,
            page,
            per_page: 9,
            orientation: 'horizontal',
        },
    });
    return response.data;
}

export { fetch };
```