function timeoutPromiseResolve(interval) {

  return new Promise((resolve, reject) => {

    setTimeout(function () {

      resolve("successful");

    }, interval);

  });

}

function timeoutPromiseReject(interval) {

  return new Promise((resolve, reject) => {

    setTimeout(function () {

      reject("error");

    }, interval);

  });

}

async function timeTest() {

await timeoutPromiseResolve(5000);

await timeoutPromiseReject(2000);

await timeoutPromiseResolve(3000);

}

let startTime = Date.now();

timeTest()

  .then(() => {})

  .catch((e) => {

    console.log(e);

    let finishTime = Date.now();

    let timeTaken = finishTime - startTime;

    alert("Time taken in milliseconds: " + timeTaken);

  });

timeoutPromiseResolve : این تابع خروجی پرامیس دارد یعنی بعد از فراخوانی این تابع اگر .then بنویسیم مقدار success با زمان interval به تابع بعدی داده میشود .

timeoutPromiseReject : این تابع خروجی پرامیس دارد و پس از زمان interval اگر پرامیسی خطا رخ دهد catch خطا را چاپ میکند و چون reject فقط دارد نمیتوان روی این تابع .then زد و فقط میتوان .catch استفاده نمود .

timeTes : این تابعی است که خروجی پرامیس ندارد و داخل آن چندین تابع پرامیس فراخوانی میشود خود این تابع هم باید تبدیل به پرامیس شود که بتوان هنگام فراخوانی روی آن .then بزنیم و باید صبر کند تا کار اولی تمام شود تا به تابع بعدی برود . به کمک async await این مشکل رفع میشود .

زمان شروع را میگیرد و پس از 7 ثانیه alert را نمایش میدهد و error هم چاپ میکند .

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous/Async_await>

سرچ کردم و این لینک رو مطالعه کردم .