A logo with a flower design

Description automatically generated

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU: PROGRAMLAMA LAB 2 3.PROJE**

**ÖĞRENCİ ADI: UTKU EMRE ERMİŞ**

**ÖĞRENCİ NUMARASI: 220501034**

**DERS SORUMLUSU:**

**DR. ÖĞR. ÜYESİ ERAY DURSUN**

**TARİH: 2/06/2024**

1. **GİRİŞ**
   1. **Projenin amacı**

**Bu projede bizden basit mantık devrelerini tasarlamak için bir platformu geliştirmemiz beklenmektedir. Program**

**ile ilgili kısıtlar aşağıdaki gibidir:**

**o Geliştirilecek olan platformun araçlar kısmında aşağıdaki özellikler bulunmalıdır:**

**o Mantık kapıları (Her bir mantık kapısını eklemek için kullanılacak araçlar):**

**O NOT Gate**

**O Buffer**

**O AND Gate**

**O OR Gate**

**O NAND Gate**

**O NOR Gate**

**O XOR Gate**

**O XNOR Gate**

**O Giriş çıkış elemanları:**

**O INPUT**

**O OUTPUT**

**O LED ışıkları**

**O Bağlantı elemanları:**

**O Çizgi çizme**

**O Bağlantı düğümü**

**O Kontrol tuşları:**

**O Çalıştır**

**O Reset**

**O Durdur**

1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

**O Mantık kapıları olcak**

**O Giriş çıkış elemanları olcak**

**O Bağlantı elemanları olcak**

**O Kontrol tuşları olcak**

1. **TASARIM**
   1. **Mimari tasarım**

**O Menü**

**O Gates**

**O Wire**

**O Input**

**O Output**

* 1. **Kullanılacak teknolojiler**

**O Java ile Node.js kullanılarak yapılacak**

**O Node.js, sunucu tarafında JavaScript çalıştırmayı sağlayan, olay güdümlü ve asenkron yapıda çalışan açık kaynaklı bir çalışma ortamıdır. Google'ın V8 JavaScript motoru üzerine kurulmuş olan Node.js, yüksek performanslı ve ölçeklenebilir ağ uygulamaları geliştirmek için idealdir.**

**İşte Node.js'in bazı önemli özellikleri ve kullanım alanları:**

**Önemli Özellikler**

**Olay Güdümlü ve Asenkron Yapı:**

**Node.js, olay güdümlü bir mimariye sahiptir ve asenkron I/O işlemlerini destekler. Bu sayede, geleneksel bloklama yapan yöntemlerden kaçınarak yüksek performanslı uygulamalar geliştirilmesine olanak tanır.**

**Tek İş Parçacığı:**

**Node.js, tek bir iş parçacığı kullanarak çalışır ancak olay döngüsü ve arka plan iş parçacıkları sayesinde yüksek verimlilik sağlar. Bu, özellikle I/O yoğunluklu uygulamalar için avantajlıdır.**

**Hızlı ve Verimli:**

**Node.js, Google'ın V8 JavaScript motoru üzerinde çalıştığı için çok hızlıdır. V8, JavaScript kodunu yerel makine koduna çevirerek çalıştırır, bu da performansı artırır.**

**NPM (Node Package Manager):**

**Node.js ekosistemi, dünyanın en büyük paket yöneticilerinden biri olan npm'e sahiptir. npm, yüz binlerce açık kaynaklı paket ve kütüphane içerir, bu da geliştiricilerin projelerinde tekrar kullanılabilir kodlar bulmalarını kolaylaştırır.**

**Kullanım Alanları**

**Web Sunucuları ve RESTful API'ler:**

**Node.js, yüksek performanslı ve ölçeklenebilir web sunucuları ve API'ler oluşturmak için yaygın olarak kullanılır. Express.js gibi framework'ler, bu tür uygulamaların geliştirilmesini daha da kolaylaştırır.**

**Gerçek Zamanlı Uygulamalar:**

**Chat uygulamaları, oyun sunucuları ve gerçek zamanlı analiz araçları gibi uygulamalar, Node.js'in asenkron yapısı ve WebSocket desteği sayesinde etkin bir şekilde geliştirilir.**

**Mikroservisler:**

**Node.js, mikroservis mimarisi ile dağıtık sistemler ve uygulamalar oluşturmak için uygun bir ortam sağlar. Küçük ve bağımsız servisler, Node.js kullanılarak kolayca geliştirilebilir ve yönetilebilir.**

**Komut Satırı Araçları:**

**Node.js, komut satırı araçları ve script'ler oluşturmak için de kullanılır. CLI (Command Line Interface) araçları, npm aracılığıyla kolayca dağıtılabilir ve kullanılabilir.**

**Sunucu Tarafı Uygulamalar:**

**Geleneksel sunucu tarafı uygulamalar, Node.js'in sunduğu hızlı I/O işlemleri ve olay tabanlı model sayesinde verimli bir şekilde geliştirilebilir.**

* 1. **Kullanıcı arayüzü tasarımı**

**O Sağ taraftan gateler ve input/outputları sürükleyerek ekrana getirilir.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**O Düğümlere tıklayarak kablolar bağlanır.**

**A screenshot of a computer

Description automatically generated**

**O Inputlara çift tıklayarak “0” veya “1” yapılabilir.**

**A screenshot of a computer

Description automatically generated**

**O Ona göre de devre çalıştırılır.**

**A screenshot of a computer

Description automatically generated**

**O Alttaki çöp kutusu simgesiyle de devre kaldırılır.**

**A screenshot of a computer

Description automatically generated**

1. **UYGULAMA**
   1. **Kodlanan bileşenlerin açıklamaları**

**O index.html**

**“head” BÖLÜMÜ:**

**O Meta Etiketleri:**

**charset="UTF-8": Bu etiket, sayfanın karakter kodlamasını UTF-8 olarak belirler. UTF-8, geniş bir karakter yelpazesini destekleyen evrensel bir kodlama standardıdır ve çeşitli dillerdeki karakterleri doğru bir şekilde görüntülemek için kullanılır.**

**name="viewport" content="width=device-width, initial-scale=1.0": Bu meta etiket, sayfanın mobil cihazlarda duyarlı (responsive) olmasını sağlar. width=device-width değeri, sayfanın genişliğini cihazın ekran genişliğiyle aynı yapar. initial-scale=1.0 değeri ise başlangıç ölçeğini 1 olarak ayarlar, yani sayfa başlangıçta %100 oranında görüntülenir.**

**O Favicon:**

**rel="icon": Bu etiket, tarayıcı sekmesinde görünen küçük simgeyi (favicon) belirler.**

**type="image/svg+xml": Favicon'un SVG formatında olduğunu belirtir.**

**href="/vite.svg": Favicon dosyasının yolu.**

**O CSS Bağlantısı:**

**rel="stylesheet": Bu etiket, harici bir stil sayfasına bağlantı sağlar.**

**href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css": Bootstrap CSS dosyasının URL'si. Bootstrap, hızlı ve kolay bir şekilde duyarlı web siteleri oluşturmak için kullanılan popüler bir CSS çerçevesidir.**

**integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN": Dosyanın bütünlüğünü sağlamak için bir SHA-384 hash değeridir. Bu, dosyanın değiştirilmediğinden emin olmak için kullanılır.**

**crossorigin="anonymous": Kaynağın başka bir origin'den (kaynak) yüklenebileceğini belirtir. Bu, CORS (Cross-Origin Resource Sharing) politikalarıyla ilgilidir.**

**O Sayfa Başlığı:**

**<title>Proje 3</title>: Bu etiket, tarayıcı sekmesinde görünen sayfa başlığını belirler. Kullanıcılar ve arama motorları için sayfanın konusunu özetler. Bu örnekte, sayfanın başlığı "Proje 3" olarak ayarlanmıştır.**

**“body” BÖLÜMÜ:**

**O “body” etiketi:**

**class="vh-100": Bu sınıf, body'nin yüksekliğini %100 viewport yüksekliği (tam ekran yüksekliği) olarak ayarlar.**

**O Araç Konteyneri:**

**class="tools container-fluid pt-3 d-flex":**

**tools: Bu div'in bir araç konteyneri olduğunu belirtir.**

**container-fluid: Bootstrap sınıfı ile tam genişlikte bir konteyner oluşturur.**

**pt-3: Bootstrap sınıfı ile üstten 3 birim boşluk ekler.**

**d-flex: Flexbox düzeni kullanarak içindeki öğelerin yan yana yerleştirilmesini sağlar.**

**O Araç Düğmeleri:**

**class="d-flex flex-column justify-content-between" style="min-width: 50px; max-width: 80px":**

**d-flex: Flexbox düzeni kullanarak öğeleri düzenler.**

**flex-column: Öğelerin dikey olarak yerleştirilmesini sağlar.**

**justify-content-between: Öğeler arasında eşit boşluk bırakır.**

**min-width: 50px; max-width: 80px: Div'in minimum ve maksimum genişliğini belirler.**

**İçindeki düğmeler, kullanıcının çeşitli mantık bileşenlerini (INPUT, OUTPUT, LED, BUFFER, NOT, AND, NAND, OR, NOR, XOR, XNOR) seçmesine olanak tanır. Her düğme, tıklanıldığında createTool işlevini çağırır ve ilgili bileşeni tuvale ekler.**

**type="button": Düğmenin türünü belirtir.**

**tool="INPUT": Düğmenin temsil ettiği aracı belirtir.**

**title="Logic Input": Düğmenin üzerine gelindiğinde gösterilecek açıklama.**

**onclick="createTool(this)": Düğmeye tıklandığında createTool işlevini çağırır.**

**class="d-flex justify-content-center btn-light btn": Düğmenin stilini belirler.**

**O Simülatör Tuvali:**

**id="canvas": Kullanıcıların mantık bileşenlerini yerleştirebileceği çalışma alanı. Bu div, JavaScript tarafından işlenecek ve kullanıcı etkileşimini sağlayacak.**

**O Sıfırlama Düğmesi:**

**class="position-absolute d-flex w-100 justify-content-center align-content-center mb-2 bottom-0 end-0":**

**position-absolute: Düğmenin konumunu sabitler.**

**d-flex: Flexbox düzeni kullanarak öğeleri düzenler.**

**w-100: Düğmenin genişliğini tam genişlik yapar.**

**justify-content-center: Düğmeyi yatayda ortalar.**

**align-content-center: Düğmeyi dikeyde ortalar.**

**mb-2: Alt kenardan 2 birim boşluk ekler.**

**bottom-0 end-0: Düğmeyi alt sağ köşeye yerleştirir.**

**\*\*onclick="reset**

**onclick="reset()": Düğmeye tıklandığında reset() işlevini çağırır. Bu işlev, simülasyon tuvalini temizler.**

**class="d-flex justify-content-center btn-primary btn": Düğmenin stili ve düzenini belirler. Bootstrap sınıfları kullanılarak düğme mavi arka plan rengine ve ortalanmış bir düzene sahiptir.**

**O Harici Betikler:**

**src="/src/main.js": Ana JavaScript dosyasının yolu. Bu dosya, simülasyonun temel işlevlerini içerir.**

**src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.9.4/p5.min.js": P5.js kütüphanesinin URL'si. Bu kütüphane, simülasyon tuvalini oluşturmak için kullanılır.**

**src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js": Bootstrap JavaScript dosyasının URL'si. Bu dosya, Bootstrap bileşenlerinin etkileşimini sağlar.**

**Gates.js:**

**Constructor (Yapıcı Fonksiyon): Sınıf, kapı tipini temsil eden bir dize alarak başlatılır. Dizeye göre, kapı genişliği ve yüksekliği gibi özellikler belirlenir. Ayrıca, fare konumunu kullanarak kapının konumu ayarlanır ve giriş ve çıkış düğmeleri oluşturulur.**

**destroy(): Bu yöntem, kapıyı ve bağlı düğmeleri yok eder.**

**draw(): Bu yöntem, kapının ve giriş/çıkış düğmelerinin görünümünü çizer. Ayrıca, fare kapının üzerine gelirse veya kapı işaretlenmişse, bir çerçeve çizer. Kapının adını da kapının altına yazar.**

**refreshNodes(): Bu yöntem, kapıdaki düğmelerin kimliklerini yeniler.**

**generateOutput(): Bu yöntem, kapının çıkış değerini hesaplar. Tüm girişlerin bağlı olup olmadığını kontrol eder ve ardından kapı türüne bağlı olarak çıkış değerini hesaplar.**

**allInputsConnected(): Bu yöntem, tüm girişlerin bağlı olup olmadığını kontrol eder.**

**calculateValue(): Bu yöntem, kapının çıkış değerini hesaplar. Kapı türüne bağlı olarak girişlerin değerlerini kullanarak çıkışı hesaplar.**

**convertToType(): Bu yöntem, kapı tipini bir dizeden sınıfa dönüştürür.**

**isMouseOver(): Bu yöntem, fare pozisyonunun kapının üzerinde olup olmadığını kontrol eder.**

**mousePressed(): Bu yöntem, fare düğmelerinden herhangi birine basıldığında çalışır. Kapının yerini ayarlamak için kullanılır veya taşınabilir modunu etkinleştirir.**

**mouseReleased(): Bu yöntem, fare düğmesinden herhangi birinin bırakıldığında çalışır. Taşınabilir modu devre dışı bırakır.**

**mouseClicked(): Bu yöntem, fare tıklaması algılandığında çalışır. Eğer fare kapının üzerindeyse, kapı işaretlenir. Ayrıca, giriş/çıkış düğmeleri için de tıklama işlemlerini yönetir.**

**Input.js:**

**constructor: Giriş nesnesini başlatır ve çıkış düğümünü oluşturur.**

**destroy: Girişi ve çıkış düğümünü yok eder.**

**draw: Girişi ve çıkış düğümünü çizer.**

**refreshNodes: Node ID'sini yeniler.**

**isMouseOver: Fare girişin üzerinde mi diye kontrol eder.**

**Fare olaylarını dinler: mousePressed, mouseReleased, doubleClicked, mouseClicked.**

**toggle: Giriş değerini değiştirir.**

**main.js:**

**preload: Giriş kapılarının SVG görüntülerini önyükler.**

**setup: p5.js ve global yöneticilerin kurulumunu yapar. Bu fonksiyon, canvas'ı oluşturur ve fare ve klavye olaylarını dinler.**

**windowResized: Pencere boyutu değiştiğinde canvas'ı yeniden boyutlandırır.**

**draw: Tüm öğeleri çizer, bu öğeler arasında kapılar, mantıksal girişler, mantıksal çıkışlar ve teller bulunur.**

**mousePressed, mouseReleased, doubleClicked, mouseClicked: Fare olaylarını işler, özellikle kapıların, mantıksal girişlerin, mantıksal çıkışların ve tellerin fare etkileşimlerini yönetir.**

**deleteElement: İşaretlenmiş öğeyi siler, bu öğeler arasında kapılar, mantıksal girişler, mantıksal çıkışlar ve teller bulunur.**

**reset: Tüm panoyu sıfırlar, tüm öğeleri yok eder ve dizileri temizler.**

**window nesnesine belirli fonksiyonları ve değişkenleri atar, bu da bunlara global erişim sağlar.**

**menu.js:**

**createTool: Bu fonksiyon, kullanıcı arayüzündeki bir araç düğmesine tıklandığında çağrılır ve belirli bir aracı oluşturur. İlgili HTML düğmesinin özniteliklerine (attribute) dayanarak oluşturulacak aracın türünü belirler.**

**elTool: Fonksiyona, oluşturulacak aracın HTML düğmesi (element) geçirilir.**

**tool: HTML düğmesinin "tool" özniteliği, oluşturulacak aracın türünü belirler.**

**isGate: HTML düğmesinin "isGate" özniteliği, oluşturulacak aracın bir kapı mı yoksa mantıksal giriş/çıkış mı olduğunu belirler.**

**gate.push(new Gate(tool)): Eğer oluşturulacak araç bir kapı ise, Gate sınıfından bir nesne oluşturulur ve gate dizisine eklenir.**

**logicInput.push(new Input()): Eğer oluşturulacak araç bir mantıksal giriş ise, Input sınıfından bir nesne oluşturulur ve logicInput dizisine eklenir.**

**logicOutput.push(new Output()): Eğer oluşturulacak araç bir mantıksal çıkış ise, Output sınıfından bir nesne oluşturulur ve logicOutput dizisine eklenir.**

**logicOutput.push(new Output(true)): Eğer oluşturulacak araç bir LED (Light Emitting Diode - Işık Yayan Diyot) ise, Output sınıfından bir nesne oluşturulur ve logicOutput dizisine eklenir.**

**node.js:**

**Node sınıfı: Mantıksal devredeki bir düğümü temsil eder. Her bir düğümün bir ID'si, bir konumu, bir değeri ve giriş durumu gibi özellikleri vardır.**

**constructor: Düğüm nesnesi oluşturulurken başlangıç değerleri atanır. Konum, değer ve giriş durumu gibi bilgiler alır. Ayrıca düğümün benzersiz bir kimliği (ID) atanır ve düğüm nodeList dizisine eklenir.**

**destroy: Düğümü yok eder. Bağlı telleri ve karşı düğümleri günceller ve bu düğümü nodeList dizisinden kaldırır.**

**draw: Düğümün görüntüsünü çizer. Değere bağlı olarak doldurma rengini belirler.**

**setID: Düğümün kimliğini değiştirir.**

**setInputState: Düğümün giriş durumunu ayarlar.**

**setSibiling: Kardeş düğümü ayarlar.**

**getSibiling: Kardeş düğümü alır.**

**getValue: Düğümün değerini alır.**

**setValue: Düğümün değerini ayarlar.**

**updatePosition: Düğümün konumunu günceller.**

**isMouseOver: Fare düğümün üzerinde mi diye kontrol eder.**

**mouseClicked: Fare tıklama olayını işler ve bir düğüm eklemeye karar verir.**

**fillValue: Düğümün değerine bağlı olarak doldurma rengini belirler.**

**GİTHUB HESABI: https://github.com/UtkuEmreErmis**