Yazılım Geliştirme Labaratuvarı

2022-2023 Bahar Dönemi II Dersi Proje I

Kare Puzzle Oyunu

Can Ata Tekirdağlı   
20130732   
201307032@kocaeli.edu.tr  
Tuba Pınar Alataş  
201307064   
201307064@kocaeli.edu.trKartal Yağız Deveci  
171307020  
kartal1eagle@hotmail.com

*Özet*— Projemizde, Visual Studio Code (HTML – CSS – JAVASCRIPT ) ve Nodejs ile yaptığımız bu puzzle oyunu kapsamında, kullanıcılar seçtikleri herhangi bir resmi yükleyerek oyuna başlamaktadır. Seçilen reim JavaScript koduyla eşit 16 parçaya bölünmekte ve yerleştirilen butonlara konulmaktadır. Doğru yere geldiği zaman butonlar hareket ettirilemez. Oyun tamamlandığında kullanıcıdan kullanıcı adı istenip yaptığı skor ve hamle sayısı .txt uzantılı dosyaya kaydedilmektedir. En yüksek skor tutulup skor sayfasında gösterilmektedir.

*Anahtar Kelimeler— HTML, CSS, JavaScript ,NodeJS, GitHUB, puzzle game*

*Abstract*— Visual Studio Code (HTML - CSS - JAVASCRIPT) is used in our project. As part of this puzzle game we made with Nodejs, users start playing by uploading any picture they use. The selected picture is divided into 16 equal parts with the JavaScript code and placed on the placed buttons. When it is in the right place, the buttons cannot be moved. The game impulse is requested from the user for a username, and the number of points and moves made are recorded in a .txt file. The highest score is kept and shows the leaderboard.

*Keywords— HTML, CSS, JavaScript, NodeJS, GitHUB, puzzle game*

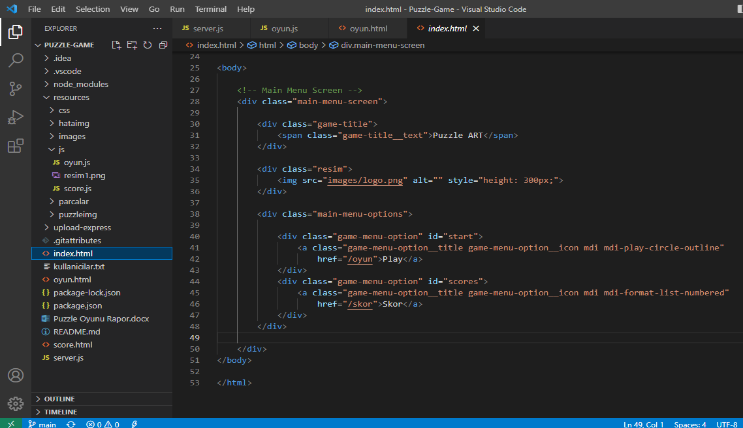
# GİRİŞ

Projemizin konusu olarak yaptığımız bu sistemin amacı kullanıcınn sisteme yüklediği görseli 4x4 boyutunda on altı eş parçaya bölen ardından karıştıran. Sonrasında kullanıcıdan button’lara tıklaması ile seçilen parçaların yer değiştirmesi ve orijinal haline getirilmesi koşuluyla tamamlanan basit görünümlü bir puzzle oyunu yaparak kendimizi CSS, HTML NodeJs ve Javascript alanında geliştirirken ortaya eğlenceli ve estetik bir yazılım çıkarmak.

# CSS-HTML-JAVASCRIPT KODLARI

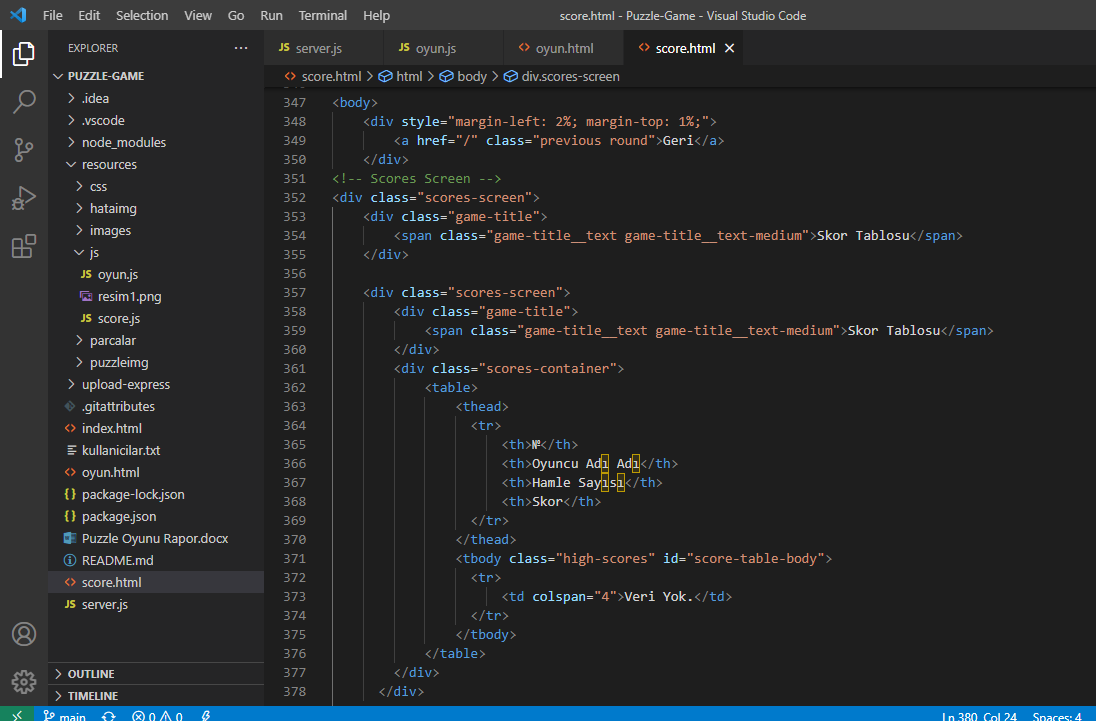
## HTML Kodları

Projemiz Visual Studio üzerinden gerçekleştirilmiştir. Aşağıdaki Görsel 1’de ana sayfa HTML kodumuz gözükmektedir. Ana sayfa kodları “index.html” içindedir.



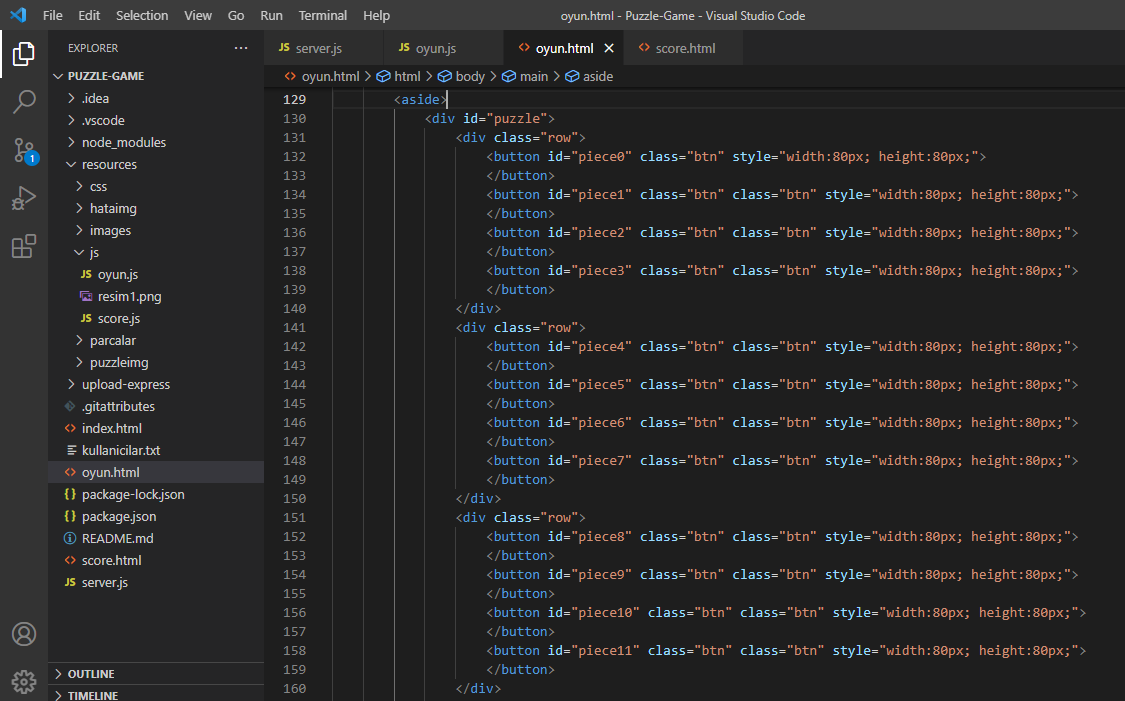
*Görsel 1*

Skor bilgilerinin bulunduğu “score.html” sayfasının kodları Görsel 2’de verilmiştir.



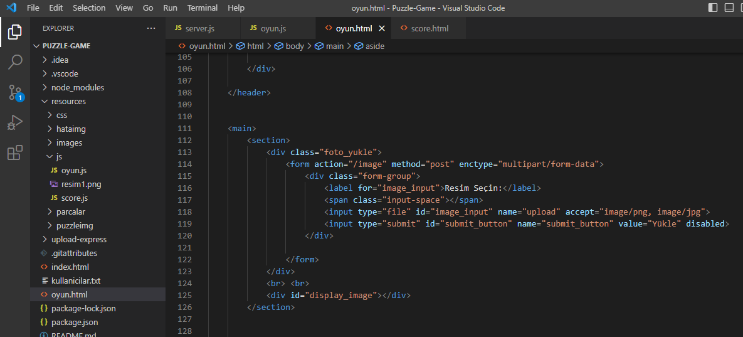
*Görsel 2*

Yüklenen fotoğrafın parçalarının yerleştirildiği butonlar için aşağıdaki HTML kodu yazılmıştır (Görsel 3). Bunun için 16 adet buton oluşturulmuştur ve id’si 0-15 arası olacak şekilde tanımlama yapılmıştır. Bu işlem daha sonra butonların içine bölünmüş fotoğraf parçalarını yerleştirmek için ayarlanmıştır.



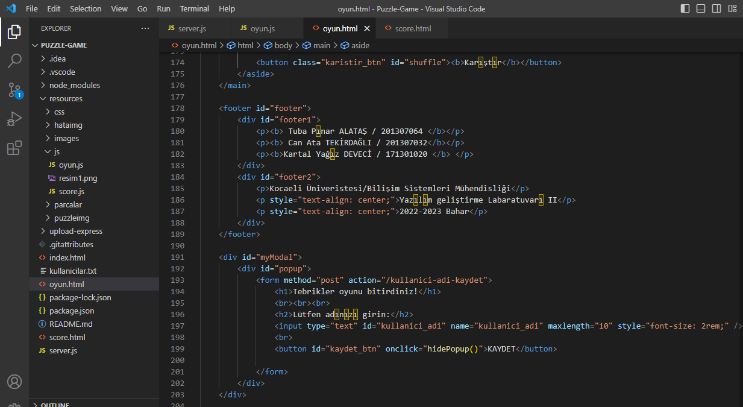
*Görsel 3*

Sayfa tasarımını yaparken sayfamızı “Header-Seciton-Aside-Footer” olarak kısımlara böldük. Aşağıdaki görselde “Section” kısmında bulunan fotoğrafı kullanıcıya yükleten HTML kodu bulunmaktadır(*Görsel 4*).



*Görsel 4*

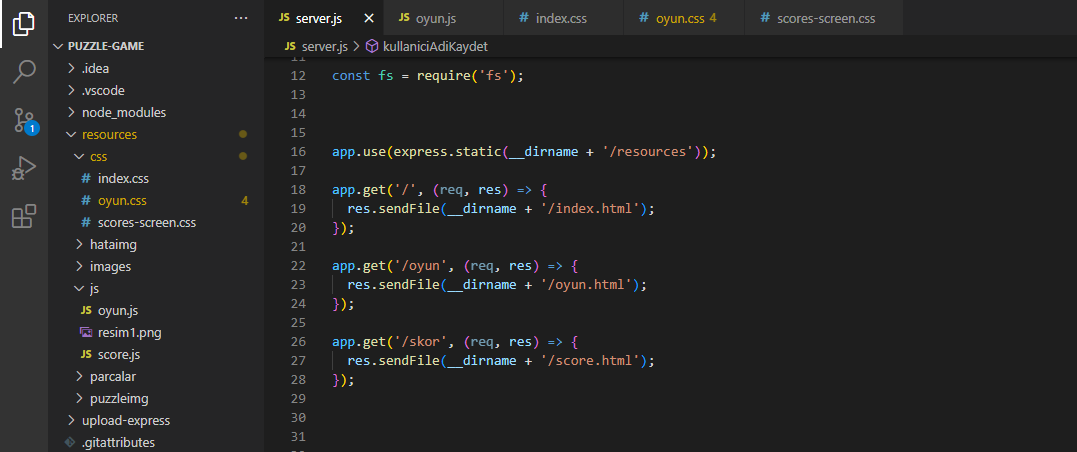
Footer kısmı ve oyun bittiğinde Pop-up olarak çıkarak kullanıcıya kullanıcı adı girmesini isteyen bölümlerin HTML kodu aşağıda verilmiştir [1] (*Görsel 5*).



*Görsel 5*

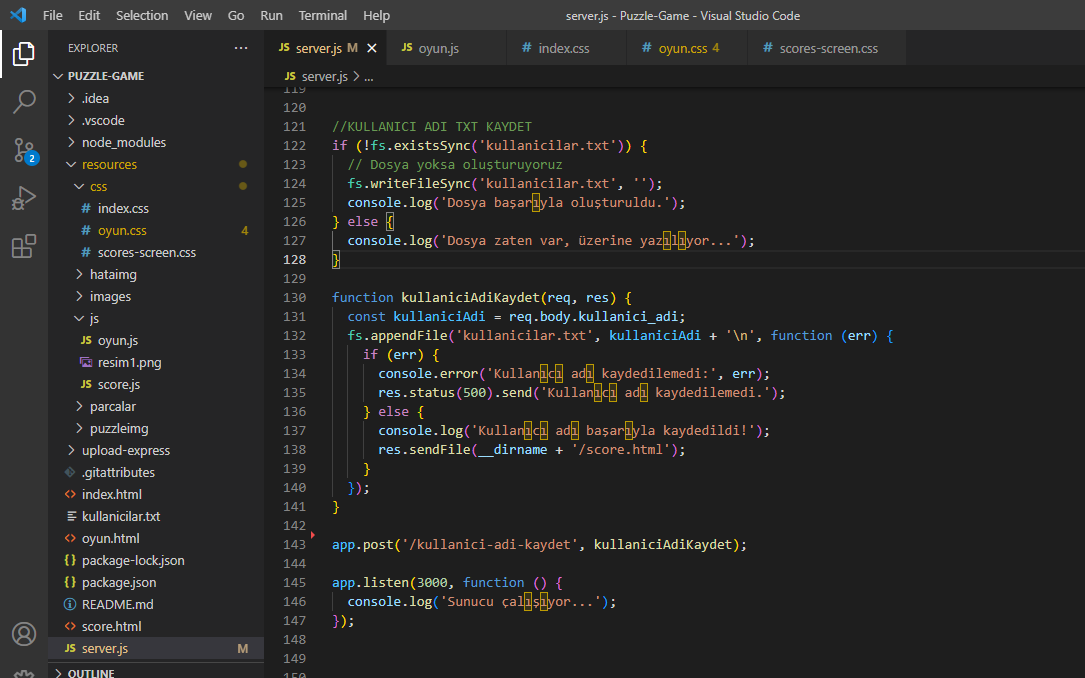
## JavaScript Kodları

Oyun sayfasının JavsaScript kodları “oyun.js” ve “server.js” içinde bulunmaktadır. Terminalden projemizi başlatmak için “ node server.js” kodunu yazıyoruz. Aşağıdaki görselde başka bir sayfaya geçerken kullanacağımız dosya yollarını düzenledik (*Görsel 6*).



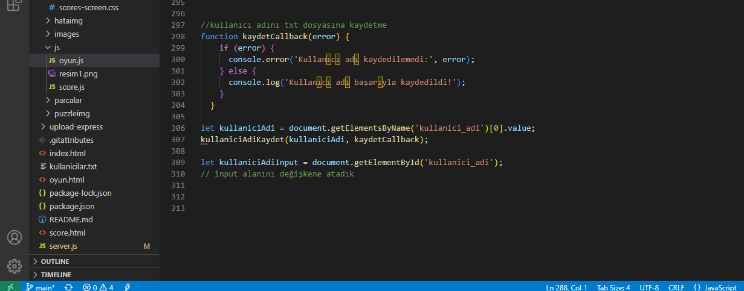
*Görsel 6*

Kullanıcıdan girilen adı “enyuksekskor.txt” dosyasına kaydeden JS kodunun “server.js” dosyasındaki kısmı aşağıdaki görselde verilmiştir(*Görsel 7*).



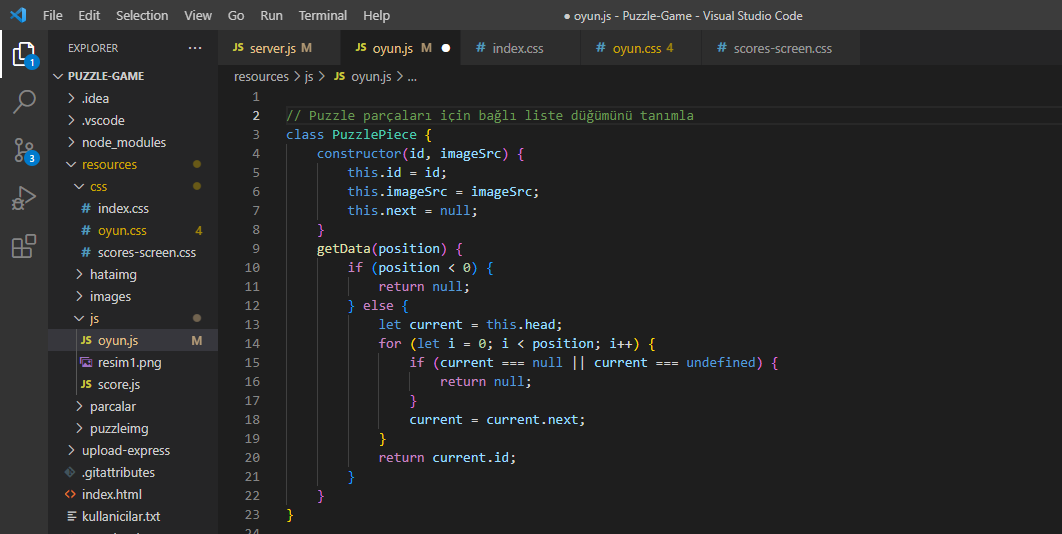
*Görsel 7*

Kullanıcı adını .txt uzantılı dosyaya kaydeden JS kodunun oyun.js dosyasındaki kısmı da bu şekildedir (*Görsel 8*).



*Görsel 8*

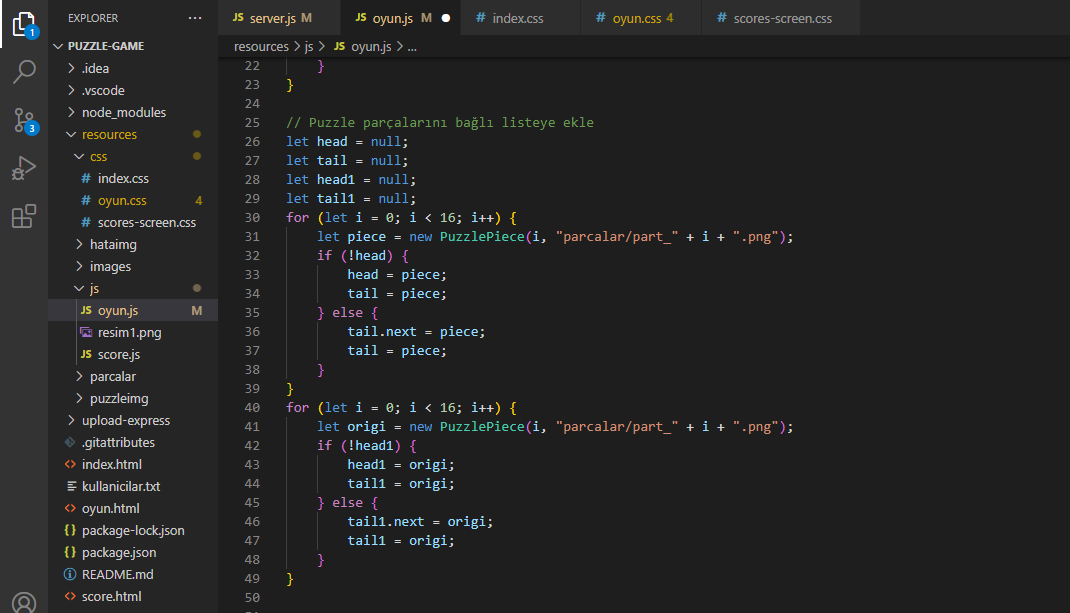
Bu JavaScript kodu bir “PuzzlePiece” sınıfı tanımlar. Bu sınıf, bir puzzle parçasını temsil eder ve her bir puzzle parçası, bir ID ve bir görüntü kaynağı URL'si içerir. Ayrıca, her puzzle parçası, başka bir parçayı gösteren bir next özelliğine de sahiptir(*Görsel 9*). Bu sınıfın “getData” metodu, verilen bir pozisyonda bulunan puzzle parçasının ID'sini döndürür. Eğer pozisyon geçersiz ise (negatif veya listede yoksa) null döndürür [2].



*Görsel 9*

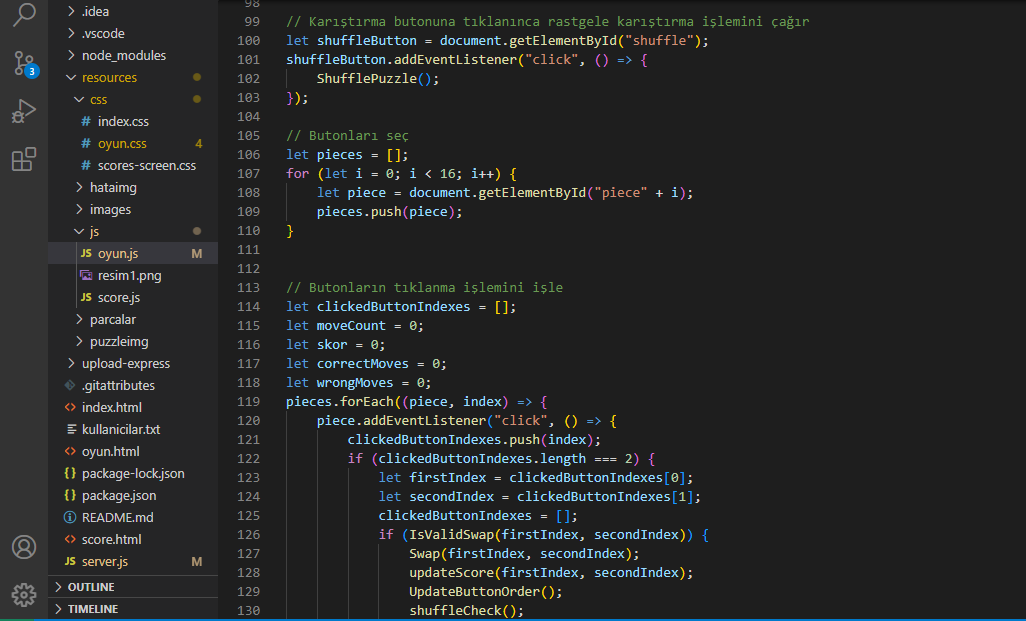
Bu JavaScript kodu, 16 parçalık bir puzzle'ı oluşturmak için iki bağlı liste oluşturur. Her bir puzzle parçası, bir ID ve bir görüntü kaynağı URL'si içerir ve “PuzzlePiece” sınıfını kullanır (*Görsel 10*).

İlk döngü, head ve tail değişkenlerini kullanarak bir bağlı liste oluşturur. Her bir puzzle parçası, “new PuzzlePiece” kullanılarak oluşturulur ve sonra bağlı listeye eklenir. İkinci döngü de aynı şekilde çalışır, ancak ikinci bir bağlı liste (head1 ve tail1 değişkenleri kullanılarak) oluşturur ve her bir puzzle parçası için ayrı bir nesne oluşturur [3].



*Görsel 10*

Karıştır butonuna tıklayınca rastgele karıştırma işlemini çağıran, butonları seçip tıklama işlemini işleyen kod *Görsel 11*’de verilmiştir.



*Görsel 11*

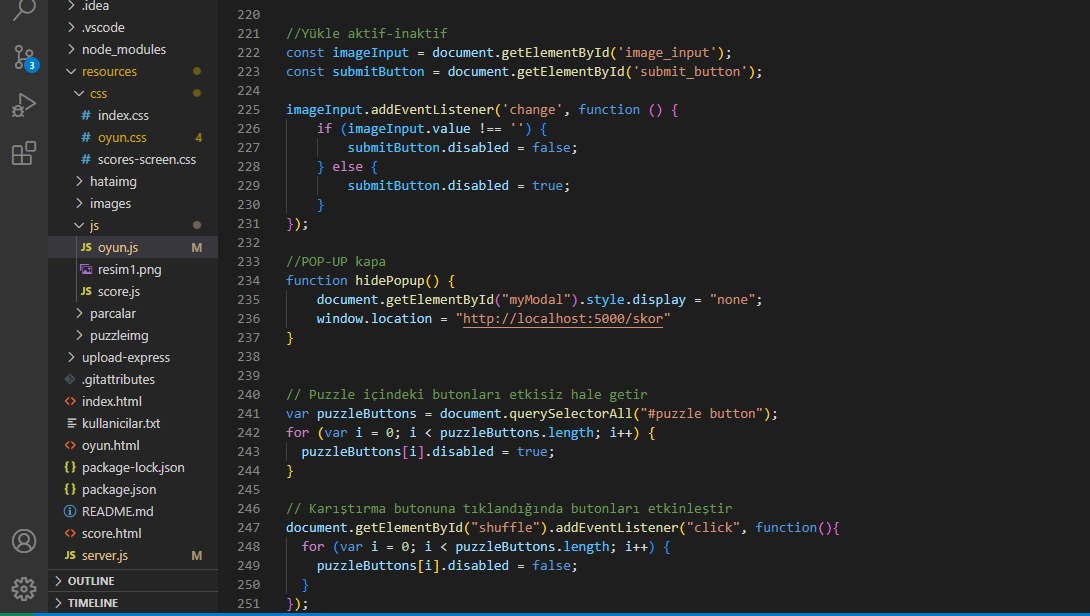
*Görsel 12*’deki JavaScript kodu, “CheckPuzzleSolved” adlı bir fonksiyon tanımlar. Bu fonksiyon, iki bağlı listeyi karşılaştırarak puzzle'ın tamamlandığını kontrol eder. Eğer puzzle tamamlanmışsa, "myModal" adlı bir modül açar. Fonksiyonun işleyişi şöyledir: önce “current” ve “ata” değişkenleri, sırasıyla “head” ve “head1” olarak atanır. Daha sonra, for döngüsü, tüm parçaları karşılaştırmak için kullanılır. Eğer parçaların ID'leri eşleşirse, karşılaştırılan buton devre dışı bırakılır ve bitis değişkeni 1 artırılır. bitis değişkeni 15'e ulaştığında, tüm puzzle parçaları eşleştiği için modül açılır [4].

UpdateMoveCount fonksiyonu, “moveCountDisplay” adlı bir HTML öğesini günceller ve hamle sayısını görüntüler.

**

*Görsel 12*

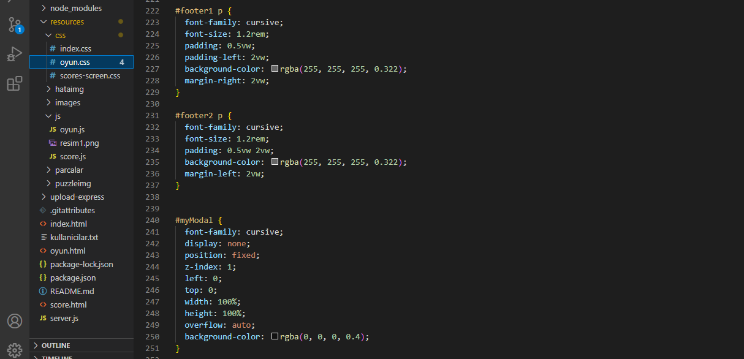
Aşağıdaki *Görsel 13*’de yükle butonunu sadece fotoğraf seçildiği zaman aktif yapar [4]. Ayrıca Pop-up açma kapama ve karıştır butonuna bastıktan sonra puzzle parçalarını aktif hale getiren JS kodu yer almaktadır.



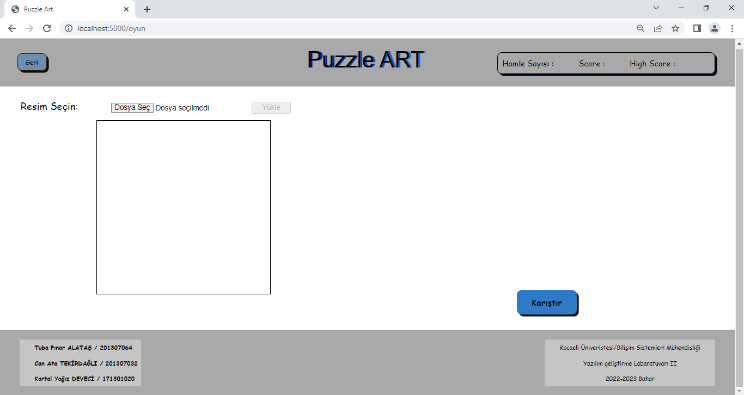
*Görsel 13*

## CSS Kodları

Oyun sayfamıza ait bazı css kodları ekran görüntüleri aşağıda verilmişitir :

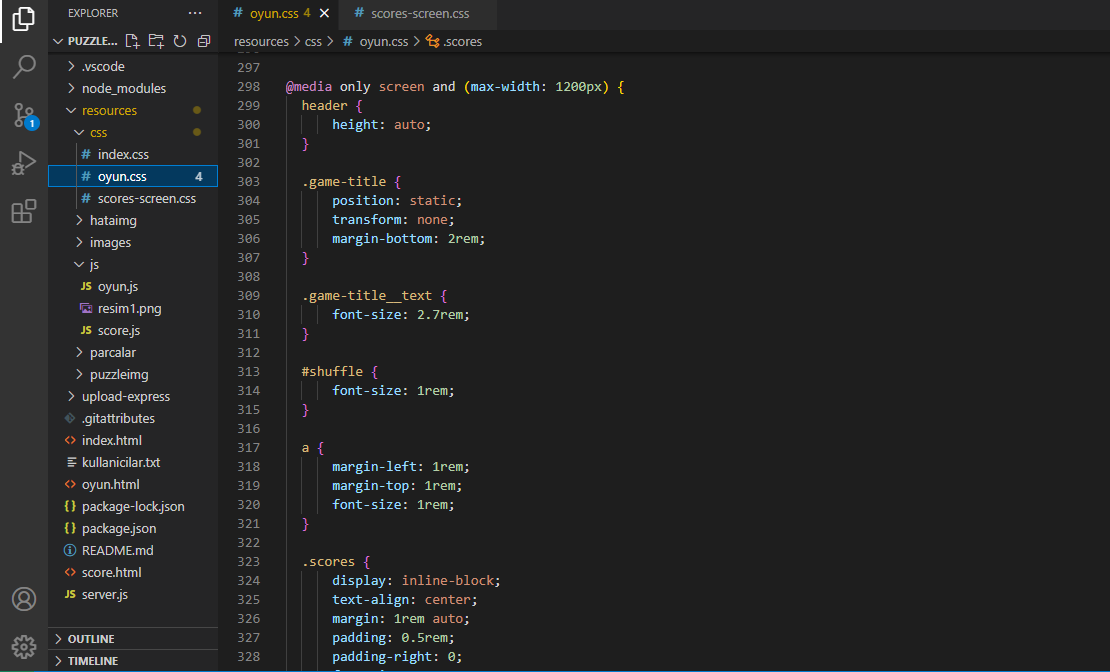


*Görsel 14*



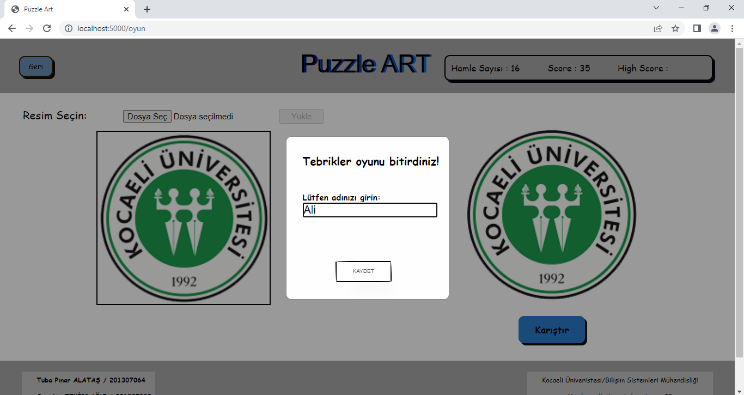
*Görsel 15*

Oyunun repsonsive olması için yazdığımız CSS kodlarından bazıları verilmiştir (*Görsel 16*) [5].

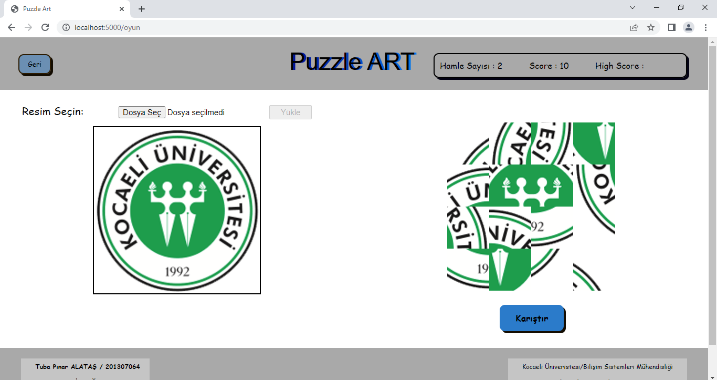


*Görsel 16*

Oyunumuzun localhost’ta çalıştığı ve bu sırada alınan bazı ekran fotoğraflarına aşağıda yer verilmiştir :



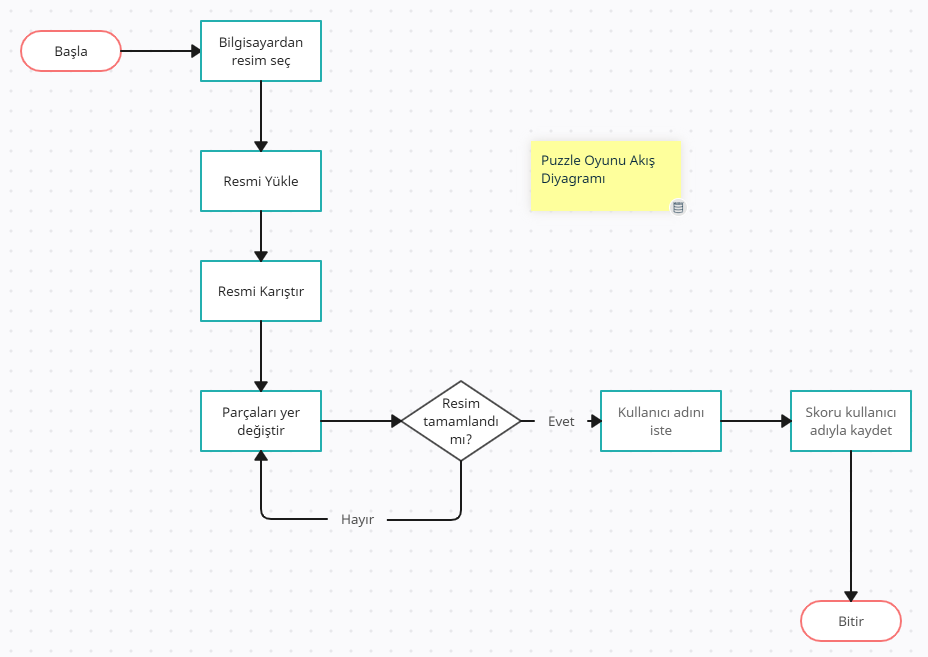
*Görsel 17*



*Görsel 18*

# AKIŞ ŞEMASI

Projenin basit anlamda çalışma prensibini görsel olarak tanıtmak için çizdiğimiz akış diyagramı:



# Kaynakça

|  |  |
| --- | --- |
| [1] | «markdownguide,» 2018. [Çevrimiçi]. Available: https://www.markdownguide.org/basic-syntax/#reference-style-links. |
| [2] | «stackoverflow,» 2021. [Çevrimiçi]. Available: https://stackoverflow.com/questions/48852567/how-to-swap-the-position-of-two-buttons-with-constraint-layout. |
| [3] | «code.tutsplus,» 2017. [Çevrimiçi]. Available: https://code.tutsplus.com/tutorials/create-an-html5-canvas-tile-swapping-puzzle--active-10747. |
| [4] | «stackoverflow,» [Çevrimiçi]. Available: https://stackoverflow.com/questions/35742199/how-to-add-image-to-button-via-javascript. |
| [5] | «stackoverflow,» 2020. [Çevrimiçi]. Available: https://stackoverflow.com/questions/42821107/express-cannot-get-images. |
| [6] | «w3schools,» [Çevrimiçi]. Available: https://www.w3schools.com/css/css3\_buttons.asp. |