PROGRAMLAMA DİLLERİ PROJESİ

Öğrenci Adı: Asım Burak Öztürk
Öğrenci Numarası: 22360859063
Ders: Programlama Dilleri
Proje Konusu: JavaScript Tabanlı Gerçek Zamanlı Sözdizimi Vurgulama (Syntax Highlighting) Sistemi

---

1. GİRİŞ

Bu projede, JavaScript dilinde yazılmış bir kod editörü ile gerçek zamanlı sözdizimi vurgulama ve temel hata kontrolü yapan bir sistem geliştirilmiştir. Amaç, kullanıcının yazdığı JavaScript kodunun anlık olarak analiz edilmesi, renklendirilmesi ve olası yazım hatalarının tespit edilmesidir.

2. PROJEYE GENEL BAKIŞ

Kod editörü, HTML/CSS/JavaScript teknolojileri kullanılarak oluşturulmuştur. Geliştirilen uygulama içinde, kullanıcının yazdığı JavaScript kodu tokenizer yardımıyla parçalanmakta, parser ile dilbilgisel kontrol sağlanmakta ve HTML içinde etiketlenerek renklendirme yapılmaktadır. Ayrıca, parantez uyumsuzlukları, eksik noktalı virgül gibi temel hatalar da tespit edilmekte ve ekranda kullanıcıya bildirilmektedir.

3. KULLANILAN YÖNTEMLER

3.1 Lexical Analysis (Sözcüksel Analiz)

Kod içinde sözcükler önceden tanımlı regex desenleriyle ayrıştırılmıştır. Aşağıdaki token türleri desteklenmektedir:

- **Anahtar kelimeler:** if, else, function, while, return, vb.
- **Değişkenler (identifier)**
- **Sayılar**
- **String ifadeler**
- **Boolean ve null değerler**
- **Yorumlar (//)**
- **Operatörler (+, -, ==, &&, vb.)**
- **Noktalama işaretleri ({, }, ;, ,, vb.)**

Tokenizer, metni karakter karakter gezerek şait eşleşen regex üzerinden token oluşturur ve bilinmeyen karakterler unknown olarak işaretlenir.

3.2 Syntax Analysis (Sözdizimsel Analiz)

Top-Down Recursive Descent Parsing yöntemiyle yazılan bir Parser sınıfı, belirli grammar kurallarını kontrol ederek kodun dilbilgisel olarak doğruluğunu kontrol eder. Desteklenen yapılar:

- Değişken tanımları (let/const/var)
- Atama ifadeleri
- If-Else blokları
- While ve For döngüleri
- Fonksiyon tanımları
- return, break, continue ifadeleri

4. PARSER'IN GENEL YAPISI

Parser, token listesini sırayla okur ve tanımlı gramer kurallarına göre sözdizimsel yapıları tanımaya çalışır. Her yapı için ayrı ayrı parseX() fonksiyonları tanımlanmıştır. Örneğin: parseIfStatement, parseFunctionDeclaration, parseWhileStatement. Bu yapı sayesinde modüler, okunabilir ve kolay hata ayıklanabilir bir sistem elde edilmiştir. Kod, parse edilirken Abstract Syntax Tree (AST) yerine doğrudan hata tespiti yapılmakta ve kullanıcıya sunulmaktadır.

5. DİL BİLGİSİ (GRAMMAR)

Parser, aşağıdaki bağlamdan bağımsız gramer (CFG) kurallarına dayanarak çalışmaktadır:

- **Program → Statement***
- **Statement → VariableDeclaration | Assignment | IfStatement | WhileStatement | ForStatement | ReturnStatement | Break | Continue | FunctionDeclaration | Block**
- **Expression → Term (Operator Term)***
- **Term → IDENTIFIER | NUMBER | STRING | BOOLEAN | (Expression)**

Bu yapılar, JavaScript dilinin temelini oluşturan sözdizimsel kuralların küçük bir alt kümesini kapsamaktadır. Fonksiyon parametreleri, blok yapıları ve ifadeler doğru sırada ve içerikte kontrol edilmektedir.

6. HATA YÖNETİMİ

Sistem iki seviyede hata denetimi yapmaktadır:

Lexical Hatalar:

- Tanınmayan karakterler
- Kapanmamış string ifadeler (tek/çift tırnak)
- Tanımsız değişken kullanımı

Syntax Hataları:

- Eksik parantezler ((, {) veya fazladan kapama

- Eksik noktalı virgül
- Yanlış sıralama veya eksik ifade yapıları (örneğin, if sonrasında parantez eksikliği)
- CFG kurallarına uymayan yapıların tespiti

Tespit edilen tüm hatalar DOM içinde alt bölümde listelenerek kullanıcıya gösterilmektedir. Hatalar hem görsel renklendirme hem de detaylı açıklamalarla sunulmaktadır.

7. VURGULAMA RENK ŞEMASI

Aşağıdaki tabloda, her token türü için kullanılan vurgulama renkleri ve stilleri özetlenmiştir:

Token Türü	CSS Sınıfı	Renk/Stil Açıklaması
Anahtar Kelime	.token.keyword	Koyu mavi (`#3b5ea7`), kalın
Sayı	.token.number	Zeytin yeşili (`#6b8e23`)
String	.token.string	Turuncu/kahverengi (`#b86b2a`)
Yorum	.token.comment	Gri (`#7d7d7d`), italik
Değişken (ID)	.token.identifier	Koyu yeşil (`#2d6a4f`)
Operatör	.token.operator	Koyu kırmızı (`#b80000`)
Noktalama	.token.punctuation	Açık kahve (`#7c6f57`)
Boolean	.token.boolean	Altın sarısı (`#b8860b`), kalın
Null	.token.null	Gri (`#808080`), italik

8. GUI UYGULAMASI

Arayüz HTML ve CSS ile inşa edilmiştir. contenteditable bir div editör alanı olarak kullanılır. Kullanıcı kod yazarken anlık olarak renklendirme ve hata bildirimi ekran altında görülmektedir.

9. SONUÇ

Bu projede JavaScript dili kullanılarak, yalın ama etkili bir gerçek zamanlı sözdizim denetleyici ve vurgulama sistemi geliştirilmiştir. Projenin başarılı yönleri:

- Token bazlı renklendirme sisteminin doğrulukla çalışması
- Gerçek zamanlı çalışma ve kullanıcı dostu GUI
- Temel hataların (parantez, string, syntax) anlık tespiti