



**T.C.**

**Fatih Sultan Mehmet Vakif Üniversitesi**

**MÜHENDSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜH. bölümü**

**LİSANS Bitirme Projesi**

**Kuran'da Arama Motoru**

**Abdurrahman RAJAB**

**Danışman: Prof. Dr. Burhanettin Can**

**İkinci Danışman: Prof. Dr. Mehmet Akif Eyler**

**AĞUSTOS 2020**

**İSTANBUL**



**T.C.**

**Fatih Sultan Mehmet Vakif Üniversitesi**

**MÜHENDSLİK FAKÜLTESİ**

**Kuran'da Arama Motoru**

**Abdurrahman RAJAB**

**BİLGİSAYAR MÜH. bölümü**

**LİSANS bitirme projesi**

Bu tez .../.../2020 tarihinde jüri tarafından Oybirliği/Oyçokluğu ile kabul edilmiştir.

Juri

imza

Prof. Dr. Burhanettin Can

...

Prof. Dr. Mehmet Akif Eyler

...

Ör. Gör. Dr. Shaaban A.I. SAHMOUD

...

Dr. Ör. Üyesi Ayla Gülcü

...



## ÖZET

### Kuran'da Arama Motoru

Abdurrahman RAJAB

Lisans Tezi, Bilgisayar MÜh. Anabilim Dalı

Danışman: Prof. Dr. Burhanettin Can

İkinci Danışman: Prof. Dr. Mehmet Akif Eyler

Ağustos 2020, 40 sayfa

Özel amaçlı bir arama motoru olarak sadece Kuran içindeki metinlere aramaktır. Özel amaçlı arama motorları genel arama motorlarından daha farklı ve özel bilgiler göstermektedir. Programın şu andaki versiyonda üç farklı bölümden oluşmaktadır bu bölümler ise Arama motoru, Oku ve Simi.

Arama motoru ise kuranda arapça yanında hem üç farklı dil ile arama yapmaktadı, Kullanıcı hem kök ile arayabilir hem de metin ile arayabilir. Kullanıcının deneyimi kolaylaştırılması amacıyla Ses ile arama özelliği bulunmaktadır.

Programın ilk açıldığında Kuran metni okuyup ekrana hazırlıyor kullanıcının isteği kök ile aramak ise kökleri hazırladıktan sonra yazdırma ve arama işlemi yapılmaktadır. Arama metni yazınca kullanıcının aratabileceği ihtimaller ve otomatik tamamlama özelliği görebilirsiniz bu özellik ise kuran metnindeki bulunan kelimelere göre çalışmaktadır. Uzun ve ekranda sığmayan sonuçlar için sayfalık kısmı bulunmaktadır sayfalara göre kullanıcı gidebilmektedir. Bundan ziyade eğer ayetin metni uzun ise ve kullanıcı bu metin tek satırda görmeye istiyorsa gösterilebilir. Kullanıcıya odaklı bir program geliştirdiğimiz için kullanıcının seçenekleri bulunmaktadır bu seçeneklerden kelime boyutu programın rengi gibi özellikler seçilebilir.

Simi ise ayetler arasında cosine benzerlik metodu ile benzerlik oranı hesaplayan bölüm, bu algoritma geliştirmesi için terimlerin frekansı ile ters doküman frekansı eklendi hem de sonuçları simetrik bir şekilde çıkması için ekstradan sadece ortak ve benzeyen kökleri ile aramaktadır. (detaylı bir şekilde daha sonra anlatıldı)

Simi ilk açılışında ise kurandaki tüm ayetlere geçip bu ayetleri kök vektörüne çeviriyor bu çevirme yaptıktan sonra herhangi bir ayet seçerek bu ayeter göre kuran-daki benzer ayetleri aramaktadır. Bu program şu anda hem cosine benzerlik algoritması kullanıyor bu algoritma iki vektör arasındaki açı bularak bu vektörler birbirine ne kadar benzediğini buluyor, aynı zamanda bu algoritma da terimleri frekansı yani bir ayet içinde ne kadar bir etki sahibi olduğunu ve ne kadar tekrarlandığını hesaplama metot bulunmaktadır, en son ters doküman frekansı ise bir kelime çok fazla geçiyorsa o zaman o kelime oranı düşük olur.

Oku Bunun yanında kuran okuması sağlayan ve küçük cihazlara odaklayan bir

okuma programı bulunmaktadır. Bu program kullanıcıyı istediği ve seçtiği bir tefsirden dataları ve bilgileri sunabilir hem de kullanıcıyı okuduğu ayet üzerine basarak o ayetin tefsiri gösterebilir. Programın aşırı yüklenmesi olmasın diye aynı zamanda üç süre yüklemektedir ihtiyacı oldukça diğer sürelerle yükleniyor.

Daha sonra ise benzerlik algoritmaları geliştirip bilgi çıkartma algoritmaları olarak kullanılabilir bu bilgiler ile bir bot oluşturup konuya göre bir paylaşım ve fikir paylaşırabiliriz, google kullandığı pageRank algoritması gibi bir şey de geliştirilebilir.

Kuranın dataları ise ve onun ilgili metinler Tanzil projesinden aldım. Bu kaynak ise güvenli ve bir çok farklı yazılış verdiği için kullanıldı. (Reference)

**ANAHTAR KELİMELER:** Arama Motoru, Arapça, Web, PWA, interactive, text manipulation, Similarity, cosine similarity, information retrieval, information extraction, JS, PWA, Web development, Search Engine, multilingual

**JÜRİ:** Prof. Dr. Burhanettin Can ...  
Prof. Dr. Mehmet Akif Eyler ...  
Ör. Gör. Dr. Shaaban A.I. SAHMOUD ...  
Dr. Ör. Üyesi Ayla Gülcü ...

## **ABSTRACT**

### **Kuran Rehberi - Arabic Search Engine for Quran**

**Abdurrahman RAJAB**

**BA. Thesis in Computer engineering**

**Supervisor: Prof. Dr. Burhanettin Can**

**Second Supervisor: Prof. Dr. Mehmet Akif Eyler**

**August 2020, 40 pages**

A special purpose search engine, That only searches for texts in the Qur'an. Special purpose search engines can give better results than the general search engines based on their specialities and can get hidden knowledge in better ways. The current version of the program consists of three different sections, these sections are Search engine, Read and Simi.

The search engine is searching in the Quran with three different languages besides Arabic. The user can both search by root and search by text. Voice search feature is available to facilitate the user's experience.

When the program is first opened, the program reads the text and prepare it to show in the screen. If the user searches with the root, printing and searching are done after preparing the roots. When you type the search text, you can see the possibilities that the user can search by the auto-complete feature. This feature works according to the words in the text of the Quran. There are pagination system for long results, and the user can go according to the specific page.

Since we have developed a user-oriented program, the user has options to make his experience much better like the single line mode and changing the font size and color of the highlights.

Simi, on the other hand, calculates the similarity ratio with the cosine similarity method between the verses, the frequency of the terms and the inverse document frequency are added to develop the algorithm, and it searches with only the common and similar roots to get the results symmetrically. (described later in detail)

In the first opening of Simi, It goes to all the verses in the Qur'an and convert these verses into the root vector. The program is currently using both the cosine similarity algorithm, this algorithm finds the angle between the two vectors and finds how similar these vectors are, at the same time, I added a method to calculate the frequency of the terms, to know how much influence it has in one verse and how many times it is repeated.

one the other hand inverse document frequency rate will be low if the word

occurrence in the whole document is large.

Oku is a reading program that provides Quran reading and focuses on small devices. This program can provide the user with data and information from a tafsir that they chooses, and can also show the tafsir of that verse by pressing on the verse he read. The program loads three chapters at the same time so that the program does not overload.

In the future, we can develop similarity algorithms and use information extraction algorithms, with the results , we can create a bot and share verses based on the subject, even though we can developer an algorithms similar to pageRank algorithm that google uses.

The data of the Qur'an and its related texts is from Tanzil project. This resource was used because it is safe and provides many different spelling.

**KEYWORDS:** Arama Motoru, Arapça, Web, PWA, interactive, text manipulation, Similarity, cosine similarity, information retrieval, information extraction, JS, PWA, Web development, Search Engine, multilingual

**COMMITTEE:** Prof. Dr. Burhanettin Can ...  
Prof. Dr. Mehmet Akif Eyler ...  
Ör. Gör. Dr. Shaaban A.I. SAHMOUD ...  
Dr. Ör. Üyesi Ayla Gülcü ...



## ÖNSÖZ

Annem Najwa Ghaloul'un, şu an yanımda olmayan rahmetli babam Muhammed Rajab'ın ve sevgili kardeşlerimin desteği , sonsuz sevgisi ve hatalarıma karşı gösterdiği sabrından dolayı teşekkür ediyorum ki ne kadar teşekkür etsem onların hakkını ödeyemem.

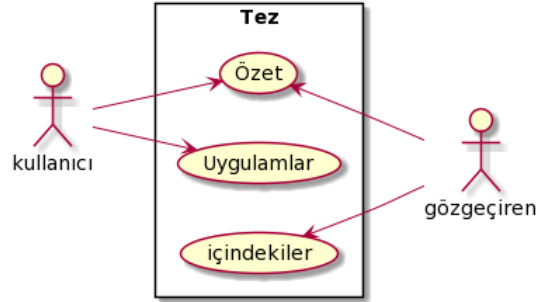
Annemden sonra bu programı sürekli takip eden, her zaman destek veren ve bana inanan Prof. Dr. Mehmet Akif Eylere ayrıca teşekkür ediyorum ve saygılarımı iletiyorum.

Bazı ameller ne kadar basit gözükse de büyük bir katkı ve etkisinin olduğu kolay bi şekilde anlaşılmaz ,bu dönemde yaptığım çalışmalar bunu daha net bi şekilde gösterdi.

İnşallah hayırlı ve verimli bir program olur.

Kitabı nasıl okurum?

Bu kitapta Kuran Rehberi adlı uygulamanın nasıl çalıştığını ve neden bu şekilde olduğu anlatılıyor. Eğer kullanıcıysanız ve programın özelliklerini öğrenmeyi istiyorsanız sadece özet ve uygulamalar bölümünü okumanız yeterli olur, programın algoritmasının nasıl çalıştığını ve hangi teknolojiler kullanıldığını kabaca anlamayı istiyorsanız özet ve içindikiler bölümünü okumanız yeterli olur, programcı iseniz teknolojileri neden ve nasıl kullandığı ilk sayfada anlatılıyor ve ihtiyacınız oldukça okumanızı tavsiye ederim.



Şekil 1: Kitabı nasıl okurum

## İÇİNDEKİLER

ÖZET . . . . .	i
ABSTRACT . . . . .	iii
ÖNSÖZ . . . . .	v
AKADEMİK BEYAN . . . . .	viii
SİMGELER VE KISALTMALAR . . . . .	ix
ŞEKİLLER DİZİNİ . . . . .	x
1. özet: (kabaca Ne Yaptığı) . . . . .	1
2. Giriş . . . . .	3
3. Tasarım ( Ideaş Simple İş More Concept And Mention Few Other Colour Theory And Optimisation Things) . . . . .	4
3.1. Algoritma . . . . .	6
3.2. Arayüz . . . . .	9
4. Kullanılan Teknolojiler . . . . .	12
4.1. Javascript . . . . .	12
4.2. Asenkron Programlama . . . . .	12
4.3. Web Workers . . . . .	12
4.4. Web Sayfaların Tasarımı . . . . .	12
4.5. Pwa . . . . .	13
4.6. Local Storage . . . . .	13
4.7. Regex . . . . .	13
4.8. Modules . . . . .	13
4.9. Git . . . . .	13
4.10. Git Stats . . . . .	14
4.11. Git Hours . . . . .	14
4.12. Git şubmodules . . . . .	14
4.13. Github . . . . .	14
4.14. Github Pages . . . . .	14
4.15. Speech Recognition (chrome) . . . . .	14
4.16. Html Frames . . . . .	15
5. Kütüphaneler . . . . .	16
5.1. Bootstrap . . . . .	16
5.2. Revealjs . . . . .	16
5.3. intro.js . . . . .	16
5.4. Mermaidjs . . . . .	16
5.5. Jsdocs . . . . .	16
6. Zorluklar . . . . .	18
6.1. Arapça Metni . . . . .	18
6.2. Türkçe Dili . . . . .	19
6.3. Html Sayfaların Manipülasyonu . . . . .	19
6.4. Normalizasyon Fonksiyonları . . . . .	20
6.5. Sayfa Linkleri Manipülasyonu . . . . .	22
6.6. Web Context Menü . . . . .	22
6.7. Google Forms . . . . .	22
6.8. çok Dilli Web Sitesi . . . . .	22
6.9. Web Tarayıcı Geçmişi . . . . .	22
6.10. Tek Satır Mod . . . . .	23

6.11. İşlemlerin önceliği . . . . .	23
6.12. Arayüz Dönmesi . . . . .	23
6.13. Sayfalama . . . . .	23
6.14. Bilgi Edinme Yöntemleri Information Retrieval Importance And Usage (cosine Similarity, Bm25, Tf,idf) . . . . .	24
6.15. Sürekli Kaydırma . . . . .	25
6.16. Event Programlama Dilleri . . . . .	26
7. Uygulamalar: . . . . .	27
7.1. Ara . . . . .	27
7.2. Oku . . . . .	32
7.3. Simi . . . . .	33
8. Gelecek Yönelik çalışmalar . . . . .	35
8.1. Konu Bazlı Arama . . . . .	35
8.2. Bilgi Edinme Ve Bilgi çıkartma . . . . .	35
8.3. Test . . . . .	35
8.4. Font . . . . .	35
8.5. Bot . . . . .	38
9. Elhamdülillah . . . . .	39
10. KAYNAKLAR . . . . .	40

## AKADEMİK BEYAN

Lisans Tezi olarak sunduđum “Kuran Rehberi” adlı bu alıřmanın, akademik kurallar ve etik deđerlere uygun olarak bulunduđunu belirtir, bu tez alıřmasında bana ait olmayan tım bilgilerin kaynađını gosterdiđimi beyan ederim.

.../.../2020

Abdurrahman RAJAB

İmza

## SİMGELER VE KISALTMALAR

### Kısaltmalar:

PWA Progressive Web application

## ŞEKİLLER DİZİNİ

1. Kitabı nasıl okurum . . . . .	v
3.1. ara genel algoritması . . . . .	5
3.2. ara bir dalın algoritması . . . . .	5
3.3. simi tablo oluşturan algoritma . . . . .	6
3.4. Data Preprocessing işlemi . . . . .	7
3.5. arama algoritması . . . . .	7
3.6. query complete algoritması . . . . .	8
3.7. benzerlik algoritması . . . . .	9
3.8. oku algoritması . . . . .	9
3.9. programın arayüzü . . . . .	10
3.10. Boyama rengi sarı seçilmiş . . . . .	10
3.11. simi arayüzü . . . . .	11
3.12. oku arayüzü . . . . .	11
6.1. Hareke kontrol algoritması . . . . .	19
6.2. Şedde şekli . . . . .	20
6.3. öğretti kelimesi 1 . . . . .	21
6.4. öğretti kelimesi 2 . . . . .	21
6.5. Algoritma grafi . . . . .	24
6.6. BM25 algoritması . . . . .	25
7.1. Tur girişi . . . . .	27
7.2. Link komutları . . . . .	27
7.3. Metin tamamlama . . . . .	28
7.4. Ses butonu . . . . .	28

7.5. Ses seçenekleri . . . . .	28
7.6. sayfalama sistemi . . . . .	29
7.7. çok kaynak . . . . .	29
7.8. Tefsire seçenekleri . . . . .	30
7.9. tek satır seçenekleri . . . . .	30
7.10. sadece arapça seçeneği . . . . .	30
7.11. Tek satır seçeneği . . . . .	31
7.12. Tefsire gösterme . . . . .	31
7.13. küçük ekranları . . . . .	32
7.14. büyük ekranları . . . . .	32
7.15. Ekran görüntüsü . . . . .	32
7.16. Simi menüsü . . . . .	33
7.17. Simi Linki . . . . .	33
7.18. Fatıha suresi ilk ayetin Sonuçları . . . . .	34
7.19. Küçük ekran menüsü . . . . .	34
8.1. idarattum kelimesi . . . . .	36
8.2. namaz kelimesi . . . . .	37

## 1. özet: (kabaca Ne Yaptığı)

Özel amaçlı bir arama motoru olarak sadece Kuranın içindeki metinleri aramaktadır. Özel amaçlı arama motorları genel arama motorlarından daha farklı ve özel bilgiler göstermektedir. Programın şu anki versiyonu üç farklı bölümden oluşmaktadır; bu bölümler şunlardır: Arama motoru, Oku ve Simi.

Arama motoru, Kuranda Arapça ile beraber üç farklı dil ile arama yapmaktadır. Kullanıcı hem kök hem de metin ile arayabilir. Kullanıcının deneyimini kolaylaştırılması amacıyla ses ile arama özelliği bulunmaktadır.

Program ilk açıldığında Kuran metni okuyup ekrana hazırlamakta ve kullanıcının isteği kök ile arama ise kökleri hazırladıktan sonra yazdırma ve arama işlemi yapmaktadır. Arama metni yazınca kullanıcının aratabileceği ihtimaller ve otomatik tamamlama özelliklerini görebilirsiniz. Bu özellik Kuran metnindeki bulunan kelimelere göre çalışmaktadır. Uzun ve ekrana sığmayan sonuçlar için sayfalık kısmı bulunmaktadır ve sayfalara göre kullanıcı istediği sayfaya gidebilmektedir. Bundan ziyade eğer ayetin metni uzun ise ve kullanıcı bu metni tek satırda görmek istiyorsa gösterilebilir. Kullanıcıya odaklı bir program geliştirdiğimiz için kullanıcının uygulamayı kendine göre özelleştirebileceği seçenekler bulunmaktadır. Bu seçenekleri şu şekilde sıralayabiliriz: kelime boyutu, programın rengi...

Simi ise ayetler arasında kosinüs benzerlik metodu ile benzerlik oranını hesaplayan bölümdür. Bu algoritmanın geliştirilmesi için terimlerin frekansı ile ters doküman frekansı eklenmiştir ve sonuçları simetrik bir şekilde çıkması için ek olarak sadece ortak ve benzeyen kökleri aramaktadır. (Detaylı bir şekilde daha sonra anlatıldı)

Simi ilk açıldığında Kurandaki tüm ayetlere geçip bu ayetleri kök vektörüne çevirmekte ve bu çevirme yapıldıktan sonra herhangi bir ayeti seçerek bu ayete göre Kurandaki benzer ayetleri aramaktadır. Bu programda şu an kosinüs benzerlik algoritması kullanılıyor. Bu algoritma iki vektör arasındaki açıyı bularak bu vektörlerin birbirine ne kadar benzediğini buluyor, aynı zamanda da bu algoritmadaki terimlerin frekansını yani bir ayet içinde ne kadar bir etki sahibi olduğunu ve ne kadar tekrarlandığını hesaplama metodu bulunmaktadır. Son olarak ters doküman frekansı ise bir kelime ne kadar fazla geçiyorsa o kelimenin oranının da o kadar düşük olduğunu gösterir.

Oku: Bunun yanında Kuranın okunmasını sağlayan ve küçük cihazlara odaklayan bir okuma programı bulunmaktadır. Bu program hem kullanıcının seçtiği bir tefsirden verileri ve bilgileri sunabilir hem de kullanıcının okuduğu ayet üzerine basarak o ayetin tefsirini gösterebilir. Programın aşırı yüklenmesi olmasın diye aynı zamanda üç sure yüklemektedir, ihtiyacı oldukça diğer sureler sonradan yüklenmektedir.

Daha sonra ise benzerlik algoritmaları geliştirip bilgi çıkartma algoritmaları olarak kullanılabilir. Bu bilgiler ile bir bot oluşturup konuya göre bir paylaşım oluşturabilir ve fikir çıkarabiliriz, Googleın kullandığı pageRank algoritması gibi bir algoritma da geliştirilebilir.



Kuranın verileri Kuran ile ilgili metinlerin bulunduđu Tanzil projesinden alınmıřtır. Bu kaynak ise gvenli ve birok farklı eřit yazılıř verdiđi iin kullanılmıřtır. (Reference)

## 2. Giriş

Program hem üç farklı ayrı bölümden oluştuğu için hem de son kullanıcıya göre geliştirildiği için genel programları inceleyip ve her bölümde hem farklı hem de ayrı bir literatür araştırması yapılmasının daha uygun olacağını öngörülmüştür. Bu tarz araştırmalar daha sonraki bölümlerde anlatılmaktadır.

İlk araştırma yaptığımda bazı temel soruların cevabı bulmaya çalıştım, bu sorular ise: Piyasada (çevrimiçi) ne tarz programlar vardır? Bu programlar ne gibi özellikler sunmaktadır? Bu özellikler teknoloji gelişiminde nerede kalmıştır? Yeni nesil kullanıcılara ne tip şeyler sunabiliriz? İnsan anlayışı ile bilgisayar algoritmalarını kullanarak yeni bilgi (Hidden knowledge) elde edebilir miyiz?

Bu sorulara tek tek cevap vermeye çalıştığım bu sonuçlara vardığımı gördüm:

Piyasada Kuran ile ilgili bir çok program var; fakat bu programların çoğu arama ya da bilgi benzerliği için kullanılmamaktadır, sadece normal okuma programlarıdır. Bunların yanında bu tarz programlar son teknolojiler ve küçük ekranlara düşünmeden yazılmıştır, bu sebepten dolayı birçok kullanıcı kaybedebilir. Ama bu özelliklere dikkat edilseydi kullanıcılar uygulamadan daha memnun kalabilir ve kullanmaya devam edebilirdi.

Referans için incelediğim programları aşağıda anlattım:

Hasenat: Çok fonksiyonlu olan bir program olmasına rağmen web versiyonu yoktur, kullanmak istediğiniz zamanda uygulamayı indirmeniz gerekir, ancak son zamanlarda insanlar sadece masaüstü programlar kullanmıyorlar, telefon ve tabletler de sık sık kullanılıyor. O yüzden web tabanlı bir uygulamaya döndüğü zaman daha verimli bir uygulama olacağını ve daha fazla kullanıcıya erişebileceğini düşünüyorum.

Quran.com: Web tabanlı olmasına rağmen herhangi bir arama yapmamakta, fakat Kurana sesli okuma gibi faydalı ve verimli özellikler vermektedir.. Örnek olarak alınır!

Kuranmeali: Araması sadece Türkçede yapılır, bunu yanında bir de farklı kaynaklardaki Kuran tefsirlerine erişim sağlar.

Bu programlar ve yazılımların çoğunun aramalarını tek bir şekilde gerçekleştiriyor ve ayetler arasındaki benzerliği bulmak için bilgiler ve köklere göre arama yapılmıyor, bu yüzden de bu işlevleri sağlayacak yeni bir program hazırlama ihtiyacı duyduk.

Son zamanlarda kullanıcılar ve teknolojilerde çoğu internet tabanlı basit ve sadece tarayıcıda çalışabilen programlar çıkmaktadır ve bunu düşünüp Kendimiz için bunun gibi bir şey neden yapmıyoruz? sorusunu yönelttik. Hem yapınca farklı teknolojiler incelemiş oluruz hem de öğrenmiş oluruz diye düşündük. Bir sonraki bölümler de bir nebze de olsa kendi öğrenim deneyimlerimi sizinle paylaşıyorum.

### **3. Tasarım ( Ideas Simple İş More Concept And Mention Few Other Colour Theory And Optimisation Things)**

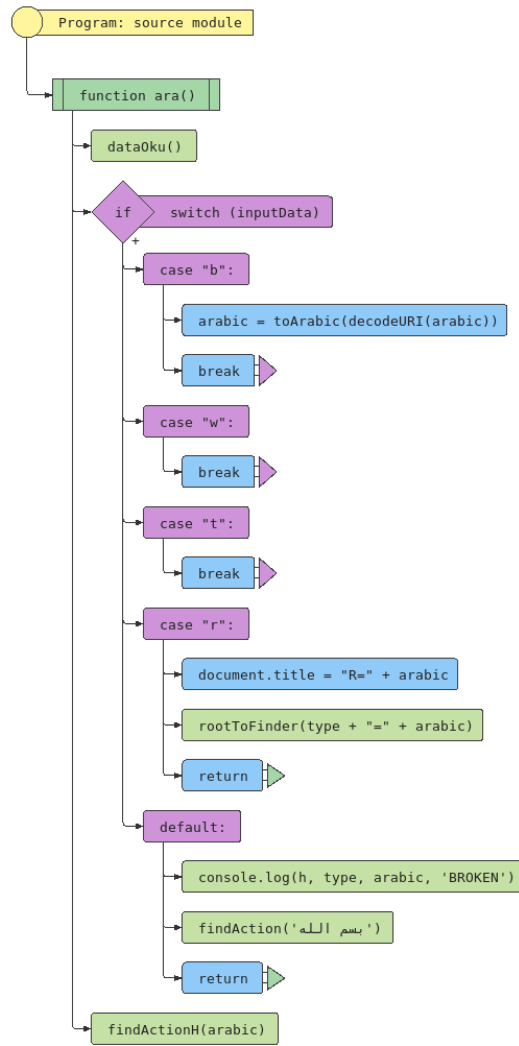
Tasarım açısından kullanıcıya basit ve programı açınca uygulamanın kişiye ne sunduğunu, nasıl kullanılabileceğini anlatan bir tanıtım sayfası koydum. Programın ilk sayfası sadece ve sadece arama yeri ve aranan bilgileri gösteren bir yer olarak karşımıza çıkıyor. Daha fazlası gerekiyorsa üst kısımda kullanıcının seçenekleri bulunuyor.

Bu şekilde sade bir tasarım olunca kimseye ihtiyaç olmadan kolaylıkla herkes kullanabilir.

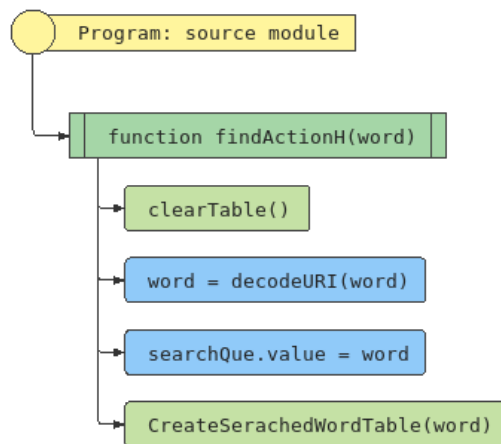
Algoritma açısından ise her program için farklı algoritma kullanılmaktadır, onların nasıl geliştirildiği detaylı bir şekilde her algoritma bölümünde anlatılmaktadır. Kabaca nasıl çalıştığı özet bölümünde anlatıldı, burada hızlıca geçiyorum.

Temelde tüm verileri ortak bir yere toplar, ilgili algoritmaya gönderir ve okutur, O veriyi okuduktan sonra her programın ihtiyacına göre verileri işleyip kullanır.

Ara: Arama motorumuz Kuran verilerini okuyup, kullanıcının girdiği veriye göre arama yapar ve sonucunu gösterir. Aşağıdaki 1.1 graf bu algoritmayı kabaca gösterir.

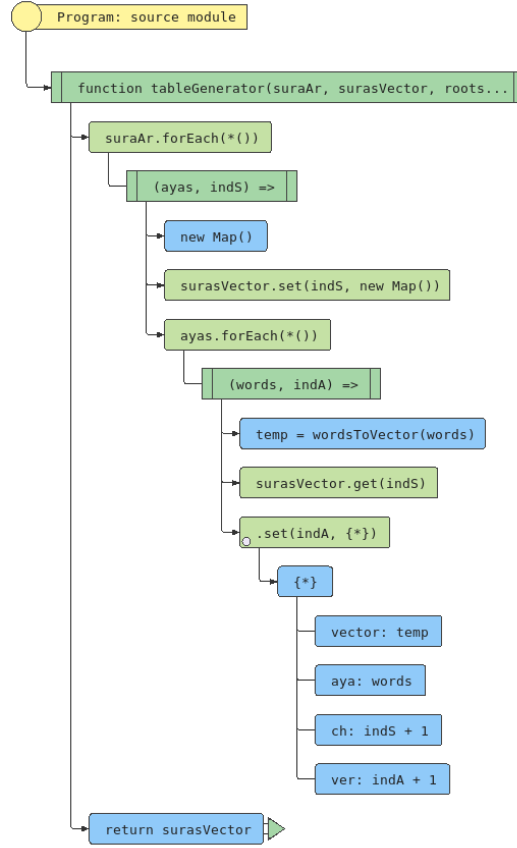


Şekil 3.1: ara genel algoritması



Şekil 3.2: ara bir dalin algoritması

Simi: Bu algoritma ise tüm Kuran ayetlerini kelime vektörüne çevirir. Sonrasında bu vektör üzerindeki benzerlik algoritmaları kullanılır.



Şekil 3.3: simi tablo oluşturan algoritma

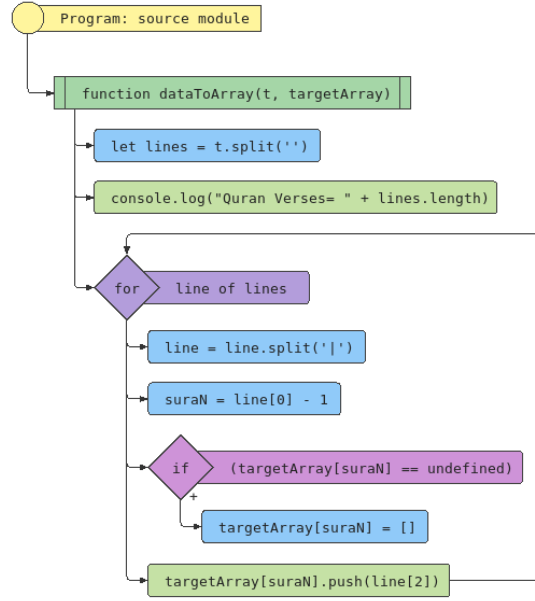
Oku: Bu program ise veriyi gösterir, ancak gösterince sadece üç adet süre üzerine gider. Aksi takdirde program aşırı yüklenmesine sebep olabilir.

### 3.1. Algoritma

Bu bölümde ise algoritmaların son tasarımı ve son durumu anlatılıyor. Bu algoritmaların neden bu şekilde olduğunu ve nasıl geliştirildiğini ise bir sonraki bölümlerde anlatılıyor, burada sadece çalışan ve verimli olan algoritmaları tek tek inceleyeceğiz.

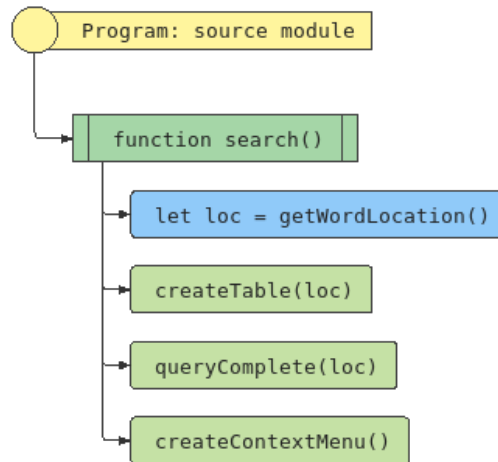
Bu üç programın başında ise veri okuma algoritması geliyor, bu algoritma temel olarak verileri işleyip kullanıyor, işlemeyen diğer programlar çalışmıyor.

Veri işlenmesi ise okunan sureler ve ayetler bir vektör içinde tutulur ve bu vektör kullanılarak diğer programlar çalışır.

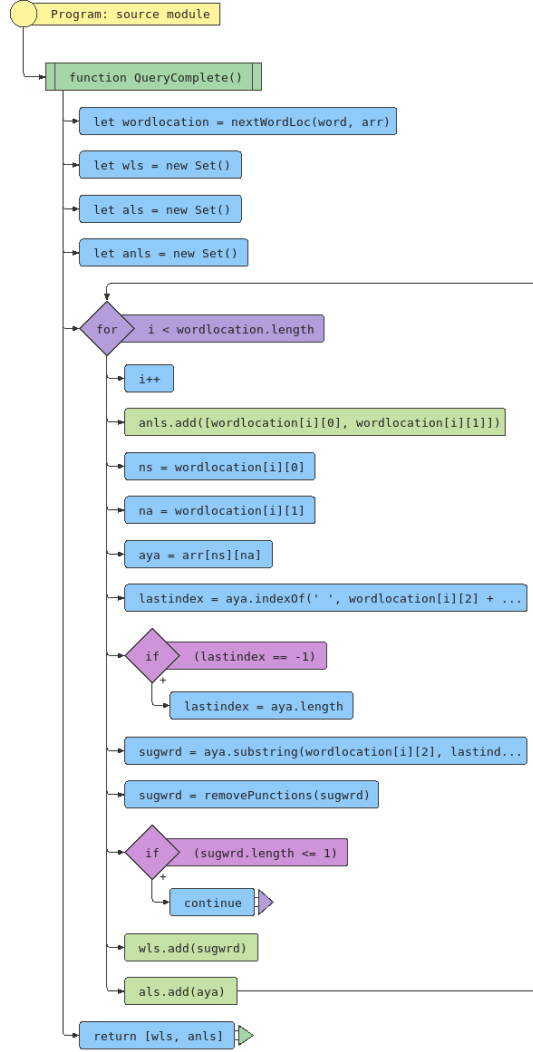


Şekil 3.4: Data Preprocessing işlemi

Ara: Arama motorunun algoritması ilk sitenin linkinden direkt bir bilgi alıyor ve bilgileri işleyip hareket ediyor, bu bilgilerin türü farklı olabilir. Kullanıcı direkt kelime, kök, buckwalter ve tefsirde arayabilir. Gelen bilgiye göre kelimeye çevirip ve bu kelimeleri arama algoritması içinde işlenir. Arama algoritması ise Kuran verileri içinde gezip hangi ayet ve sure içinde bulunduğu ile ilgili bir bilgiyi sunar, bu bilgi ise tablo oluşturma algoritması ve metin tamamlama algoritmasına gönderilir, metin tamamlama algoritması bir sonraki boşluğa kadar olan ilk 10 bilgiyi geri döndürür ve ekrana bir bağlam menüsü olarak eklenir. Tablo oluşturma ise gelen ayet ve sure bilgileri alınıp her sayfa için kullanıcı seçtiği miktarda veri gösterilir, gösterilen veri içinde bir daha aynı kelime aranır bulunduğu bir etiket koyup ve renklendirilir. En son aranan kelime ve metin giriş yerinden yazılmış ise sayfanın linki bu kelimeye göre gösterilir ve sunulur.



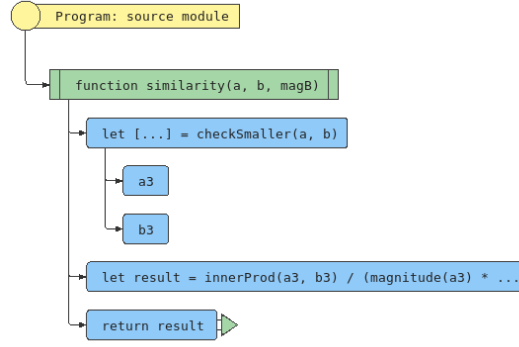
Şekil 3.5: arama algoritması



Şekil 3.6: query complete algoritması

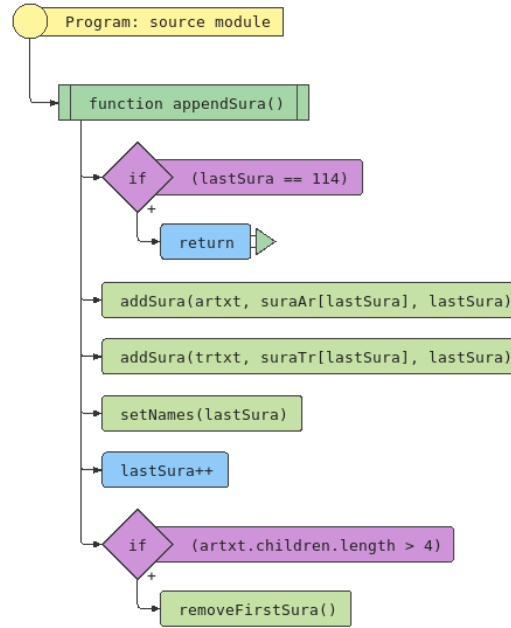
Simi: Simi ise ara gibi verileri bekleyip hazırlayıp her ayeti kök vektörüne çevirir, çevirdikten sonra seçilen ayetlere göre bu kök vektör kullanılarak diğer ayetler arasındaki benzerlik oranı bulunur, bu benzerlik oranı kullanıcının seçtiği oranın üstünde olduğu zamanda ekrana yazdırılır, buradaki geri dönen sonuçlar az olduğu için sayfalama ihtiyacı duyulmadı.

Benzerlik algoritmalarının geliştirmesi için birkaç farklı algoritma kullanıldı, bazıları direkt kosinüs açısına göre hesaplanırken bazıları da terimler ve doküman frekansı göre yapıldı, en son BM25 arama ve bilgi edinme algoritması kullanıldı, bu algoritmalar kullanıcının istediğini seçer ve gösterir.



Şekil 3.7: benzerlik algoritması

Oku: Oku ise temel algoritma kullanarak verileri kullanıcıya sunar, sunulan sureleri sürekli kaldırabilmesi için bir sonraki sure ve önceki sure gösterilir. Bu sayede kullanıcı ne kadar zaman harcadığını ve ne zaman kaybettiğini hissetmeyecektir (reference). Bir sonraki ve önceki sureler göstermesi için farklı fonksiyonları kullanıldı ve bu şekilde devam ediyor. Tefsir ile kuran arasındaki uyumun düzgün bir şekilde işleyebilmesi için bir ayetin üstüne tıklayarak ayetin tefsiri gösterir.



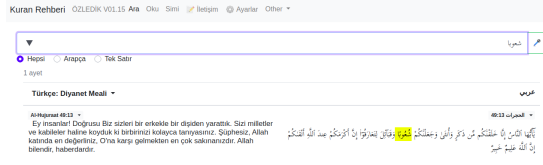
Şekil 3.8: oku algoritması

### 3.2. Arayüz

Arayüz tasarımı da basit ve ortak bir şekilde yapmaya çalıştım, o yüzden programın sayfasına sadece kullanıcının ihtiyacı olan ve basit elemanları konuldu, daha fazla bir şey lazım ise üst menüden ulaşılabilir. Tasarım küçük ve orta boy cihazlara göre yapıldı, her tür cihaza uyumlu olması için ekran genişlikler alınıp ona



göre farklı bir şekilde gösterilir. Aşağıdaki görseller her ekran boyutu için farklılıklar gösterir.



Şekil 3.9: programın arayüzü

Temelde tüm programlara benzer bir tasarım sahibi yapmaya çalıştım bu şekilde kullanıcı farklı programa geçtiğinde programın aynı kişiye bağlı olduğunu gösterir ve bizim görsel kimliğimiz sabitlenir(reference design identity).

Ara: Ara programında gösterilen veriler farklı şekillerde gösterilir, bu farklı kullanıcıya tüm ayetlerin bir sayfada sığması ve bu sayfadaki gereken bilgilere sahip olunması için yapıyor. Tek satır modu ise bu özelliği sağlar, ayet yanında sure numarası ismi ve ayetin numarası gösterilmektedir, hem tefsir hem de meali istenilen dilde gösterir.

Aranan kelime ise kullanıcının kolay bir şekilde anlaması için boyanmıştır, bu boyamada kullanıcı istediği rengi seçer.



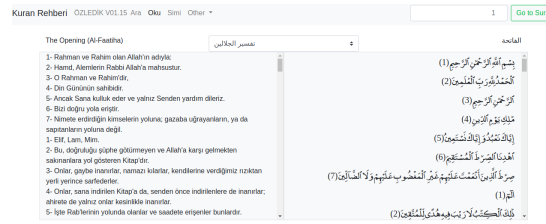
Şekil 3.10: Boyama rengi sarı seçilmiş

Simi de aynı şekilde yapılmıştır, ancak hem benzerlik oranı ve benzemeyen kelimelerin boyanması eklenmiş hem de benzerlik oranı oluşturulmuştur.



Şekil 3.11: simi arayüzü

Oku ise aynı menüye sahibi olup ancak sadece iki metni yan yana gösterir.



Şekil 3.12: oku arayüzü

## 4. Kullanılan Teknolojiler

Herhangi bir açık kaynak bir program diğer insanlardan alıntı alır ve onların üstüne bir program geliştirir, bu bölümde ise hangi teknoloji kullanıldığı ve ne fayda sağladığını anlatıyorum, seçilen teknolojilerin çoğu açık kaynaktır, yazarlar sağ olsun.

### 4.1. Javascript

Web tabanlı bir uygulama ve her yerde çalışması için JavaScript programlama dili üzerinde yazılmıştır, bu programlama dilinin özelliklerinden birisi asenkron olmaktadır. Asenkron olduğu için web sayfaların döndürmesi ve arayüz göstermesi çok hızlı olur, fakat alıştığımız pascal fortran ya da java gibi çalışmamaktadır, kodları satır satır okumuyor fonksiyonun işlenmesi biraz sürecek ise bir sonraki fonksiyona gider ondan sonra da eski fonksiyon işlemleri gerçekleştirir.

### 4.2. Asenkron Programlama

Bu tarz bir programlama (reference) sistemi olunca bir programın eski fonksiyonlarının çalışması ve bitmesi için bazı kontrolleri ve beklentileri koyması gerekir o yüzden await, event, callback ya da promise gibi özellikler kullanılabilir, bu özelliklerin her biri için farklı ve ayrı bir amaç söz konusudur. Örnek olarak await kullanılmaya kalkılırsa kullanılan fonksiyonun sonucu bir promise olarak döndürmesi gerekiyor.Promise döndürdüğünde sadece ve sadece o fonksiyon gerçekleştirdiğinde çözülmüş döndürüyor, detaylar için (reference) kitabına bakıp inceleyebilirsiniz.

### 4.3. Web Workers

Bu tarz bir programlama dili için herhangi bir thread ya da ayrı mekanizma çalışan bir kod yoktur, tüm kodlar aynı thread içinde çalışır, tüm kodları yükleyip ondan sonra çağrılabilir, ondan mütevellit bazı hatalar olabilir. Eğer uzun çalışan ve işlemi çok büyük bir döngü içinde olduğumuzda kod çalışması zor olur ve bazen asenkron olayları bozulabilir veya beklediğimiz gibi olmayabilir, o yüzden farklı amaçla kullanılan web workers diye bir şey çıkmıştır, bu özellik ise java script kodu farklı bir thread çalıştırır ve oradan kullanılabilir, bu thread ana thread arasındaki iletişim eventler ile sürüyor. Bir event geldiğinde ne yaptığını ve nasıl işleneceğini anlatılır ve bu şekilde devam edilir.

### 4.4. Web Sayfaların Tasarımı

Web sayfalarının tasarlanması için ve onlara şekli verilmesi için CSS diye bir gösterim dili kullanıyor, bu dil programın içindeki bulunan elemanların nasıl görüneceği ve gözüktüğünde nasıl bir effect olacağını yazıyor, bu görselleştirme dili kullanılması küçük ve orta boy ekranlarda ekranın farklılığı anlar ve ona göre farklı bir şekil gösterilir.

#### 4.5. Pwa

Progressive web application son zamanlarda yeni çıkan bir kavramdır, bu kavram ise bir programın Tüm cihazlar üzerinde çalışması için ne yapılmalı? sorusundan çıktı, bu teknoloji ise web tarayıcıları kullanır ve programlar web tarayıcısı üzerinden çalışmaz, fazladan herhangi bir ekleme veya program ihtiyacı yoktur, bu şekilde yapınca web tarayıcıları çalıştıran cihazlar kendi programımızı çalıştırır ve program uygulama gibi davranıp indirilebilir.

#### 4.6. Local Storage

Local storage ise web tarayıcılardan sağlanan bir storage yeridir, oraya eklenen herhangi bir bilgi kaybolmamaktadır ve program açılınca bir daha kullanılabilir, ancak bilgisayar üzerine herhangi bir şey yazmaz. Bu local storagei bazı kullanıcıya özel veriler için kullandım.

#### 4.7. Regex

Regular expression adlı bir teknolojidir, bu teknoloji özel karakter içinde verilen kurallara göre arama yapar, çok faydalı ve verimli özellikler metin içinde arama yapınca hem çok hızlı hem de optimal şekilde çalışır, yeni bir şey olmadığı için çok verimli olmaktadır. Karakterlerin arama yapması için sadece unicode bilgileri vermemiz yeterli olur, bu şekilde bir aralık belirlemiş oluruz, daha sonraki bölümlerde anlatacağım hareke sorununu bu teknoloji ile kontrol ettim.

#### 4.8. Modules

Modules bir javascript özelliklerinden birisidir, bu özellik ise küçük bir javascript kodu başka bir koda çağırma imkanı sağlar ve kolaylaştırır bu şekilde olunca kodun başında import kelimesi kullanabiliriz. Programımız çok farklı dosyalardan oluştuğundan dolayı bir arada olan kodların bağlantıları hangi dosyadan geldiğini ve gerekli olan dosyaları anlaması için kullanmayı düşündük, ancak bazı kodlar eski yapıyı zorladığı için ondan vazgeçtik. Bu özellikleri kullanınca kodu birçok kez tasarlayıp yazmamız gerekecekti.

#### 4.9. Git

Versiyon kontrol sistemidir, bu teknoloji olmadan şu anki gördüğümüz ve okudunuz rapor olmazdı veya olması için farklı bir düşünce ile projeye başlamam lazım olacaktı. Bu teknoloji ise her yaptığım değişiklik ve değişikliklerin yerlerini kodda gösterir; bu şekilde ne yapıldığını, ne zaman yapıldığını, kimin yaptığını öğrenebiliriz. Ayrıca de bir kod yazınca mesaj yazabilir ve bu mesajların bu kodu neden eklediğimizi ve ne fayda sağladığımızı yazabilir. Uzun vadede ve çok kodcu olan bir programlar için çok faydalı ve güzel olur.

#### 4.10. Git Stats

Git teknolojisi kullanarak bizim kullandığımız ve yazdığımız kod ile ilgili bir çok istatistik getiren bir teknoloji bunu kullanarak projeye ne zaman başlandığını ve ne kadar sıklıkta yazıldığını kiminle yapıldığını gösteren bilgiler oluşturur, bu bilgiler ise son raporda eklenir ve gösterilir.

#### 4.11. Git Hours

Oldukça basit ve faydalı bir araç yaptığım ve kullandığım git dataları göre ne kadar çalıştığını ve proje üzerinde ne kadar zaman harcadığını hesaplayan bir araçtır. Bu araç yine de raporlar için çok güzel ve faydalı olur.

#### 4.12. Git submodules

Birçok program üzerinde çalışınca ve bu programlar arasında bir bağlantı kurmak istenince kullanılan bir araç bu araç ise farklı seçtiğiniz bir git reposu seçip o repoyu kendi istediğiniz repoya eklenir, tüm kodlar olmasa bile referans olarak gösterir ve istediğiniz tarihte kalabilir, yani diğer projede gelişme olduğunda sizin projeniz bozulmaz ve sabit kalır. Bu şekilde herhangi bir bozulma sebebi mani olur ve kullanıma kolaylaştırır ve çok hızlı olur.

#### 4.13. Github

Github ise gitin SAAS (software as a service) versyonu github üzerinde gitin tüm ihtiyaçları ve bilgileri internet üzerinde tutulur ve gösterilir, onu kullanarak kod her zaman bültta olur ve hiç kaybolmaz, silinmez. Hem farklı insanlar için kod açık olur; hem istedikleri zaman kodu inceleyip, geliştirme yorumları gönderebilirler hem de sorunları ilgili bilgi verebilirler. Bu şekilde kod geliştirme toplulukça olur ve ne yapıldığını nasıl geliştirildiğini belli eder.

#### 4.14. Github Pages

Github pages ise github host özelliği bu özellik kullanarak github üzerinde paylaştığımız kodu döndürür ve herhangi bir tarayıcıdan okunabilir hale gelir, bu şekilde kodun sadece bir yere yükleyip ve internette açarsan tüm insanlar senin projene erişilebilir ve kullanılabilir hale getirir.

#### 4.15. Speech Recognition (chrome)

Ses tanımlama fonksiyonları ve özelliği maalesef şu anda sadece chrome üzerinde çalışmaktadır. Bu özellik ise sesi anlayıp ve metin haline çevirir. Bu metin Google'nin kullandığı api ile yapılmaktadır. Oldukça basit ve kolaylaştırıcı bir özelliktir, şu anda hem arapça hem de diğer diller ile çalışmaktadır.

#### **4.16. Html Frames**

HTML çerçeveleri ise var olan bir siteyi başka bir site içinde açma izni verir, bu çerçeveleri birkaç yerde hem projede hem de raporda kullandım, bu özellik ise çok basit ve sadece sayfa açma izni verir, sayfa açtıktan sonra istenilen data yazdırılabilir ve gösterebilir..

## 5. Kütüphaneler

Her program için bazı hazır ve iş kolaylaştıran kütüphaneler vardır. Burada, kullandığım ve faydalı olan bazı kütüphaneleri sizinle paylaşıyorum. Kütüphaneler insanların yaptığı işi bir daha yapmaması için yapılmıştır aynı zamanda herhangi bir kütüphane kullanınca temeli ve nasıl çalıştığını mühendislerin bilmesi önemli bir husustur çünkü sadece kullanıcı olmayı hedeflemiyoruz, üretici bir mühendis olmamız da gerekiyor.

### 5.1. Bootstrap

Bootstrap ise bir CSS kütüphanesidir, bu kütüphane içinde hazır elemanlar vardır ve bu elemanları kullanarak daha hızlı verimli ve güzel bir program çıkartılması mümkün olabilir, bu kütüphanede bulunan mantığı sayfa ise tablo gibi düşünüp onun üzerinde kaç sütun kaç satır verileri yazıp ve bu şekilde elementleri yerleştiriyor, oldukça basit ve verimlidir. Bunun kullanımında birçok satır kod ve bir çok beklenmeyen durumdan kurtarmış oluruz.

### 5.2. Revealjs

Bu kütüphanede bir sunum kütüphanesi neden ihtiyacı oldu mu? Çünkü normal sunumlarda yeterli olmadığını düşünüyoruz ve hem daha hareketli, verimli hem de kolay bir kütüphane ihtiyaç duyuldu, bu kütüphaneyi kullanarak hem web teknolojilerinin tüm özellikleri kullanılabilir aynı sunumda bir çok şey hareketlendirebiliriz hem de bir internet sayfası ekleyip sunumda gösterebiliriz ve sayfayı kullanabiliriz.

### 5.3. intro.js

Tur kütüphanesi bu kütüphane kullanarak yeni kullanıcı istediğiniz sayıda tur gösterebilir, bu tarz türler kullanıcıyı sayfaya özel ve nail çalıştığını anlatan bir bilgi verir ve onun birlikte sayfanın özellikleri ve elemanlarına gezer. Bu tarz kütüphaneler yeni kullanıcılara ve programa alışmayan ya da hiç kullanmayan kullanıcılara için kullanılabilir. Güzel bir eğitici turu yapabiliriz.

### 5.4. Mermaidjs

Bir graf çizme kütüphanesi grafların çizimi ve kullanımını daha da kolaylaştırır, bu rapordaki gördüğümüz flow chart grafları ve çizimleri bu kütüphane kullanarak yapıldı, kullanımı oldukça basit olduğu için çok faydalı oldu, kullanıcı tasarım ve şekil ile uğraşmaktan kendi bilgilerini yazınca kütüphane gerekeni yapıyor. Bu tarz bir kütüphane kullanınca onlarca saat kazandım ve raporları daha hızlı bir şekilde yazabildim.

### 5.5. Jsdocs

JS docs bir dokümantasyon oluşturma kütüphanesi bu kütüphane ise kodlar üzerinde geçiyor ve kodların içindeki bulunan tüm verimli ve faydalı yorumları alıp bir dokümantasyon hazırlıyor. Bu dokümantasyon ile bazı eğitici dokümanlar ve tav-

siyeler de eklenebilir. Bu tarz kütüphaneler kullanınca daha sonra kodlara bakmak ve geliştirmek kolaylaştır, hatta herhangi bir yeni kullanıcı için kodların anlaşılması kolaylaştır ve neden öyle çalıştığını gösterir.



## 6. Zorluklar

Her program geliştirildiğinde mutlaka zor günler gelir burada bu zor günleri ne olduğunu ve nasıl çözdüğümü bazı zamanlarda nasıl anladığımı da anlatıyorum, inşallah sizin için verimli ve güzel bir bilgi içerir ve faydalı olur.

### 6.1. Arapça Metni

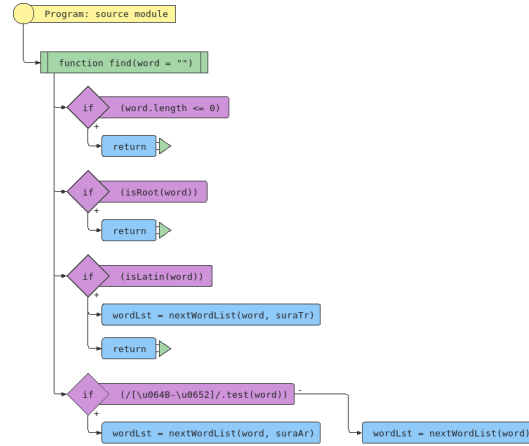
Arapça dilin özelliklerinden birisi aynı harf bir çok şekilde yazılabilir, bu sebepten dolayı Kurandaki yazılanların göstermesi ve işlemesi kolay değildi, aynı zamanda Latin alfabesi kullanılmadığından dolayı bazı web tarayıcıları içinde arapça bir kelime yazınca çok uzun ve anlamsız bir ifadeye çevirip kullanılabilir. Bu tarz yaklaşımlardan uzaklaşabilmek ve anlamlı bir link verebilmek için buckwalter transliteration yöntemi kullanmaya tercih ettik, aşağıdaki iki yazı ile buckwalter ile normal tarayıcılar arasındaki farkı gösteriyorum.

Yöntem	Arapça metni
Arapça	
Buckwalter	bsm Allah
Tarayıcı otomatik	%D8%A8%D8%B3%D9%85%20%D8%A7%D9%84%D9%84%D9%87

Bu şekilde daha anlamlı ve kullanıcıya uygun bir şekilde link ve data gösterme şansımız oldu.

Bunun yanında bazı hareketler (harf üstünde gelen işaretler) ve Kuran'a yönelik olan yazımlarda zorluk çektim, bu zorlukları ise harekesiz metin ile hareketli metin arasında nasıl bir arama ve kıyaslama yaparım, eğer kullanıcı hareketsiz metin girdiğini hareketli metin içinde nasıl aratabilirim? Gibi sorular ile karşılaştım bunu ise basit bir yöntem ile çözdüm.

Kuran verileri iki ayrı metinden oluşuyor: hareketli ve harekesiz. Kullanıcının girdiği dataya göre hareketli olduğunda hareketli Kuranda arama yapıp gösteriliyor, harekesizde ise harekesiz metinde arama yapıp ayet ve süre bilgileri dondurup ona göre işlem yapılır.



Şekil 6.1: Hareke kontrol algoritması

## 6.2. Türkçe Dili

Türkçe dilinde çok fazla bir sorun yaşamadım, ama bazı karakterler program çalıştırınca zorluk gösterdi, bu karakterlerden birisi ise i harfi. Bu harf büyük iken küçültünce onun önünde fazladan bir karakter eklenir bu karakter amacı bir daha büyütünce türkçede yazıldığı anlamamız içindir, ancak bir metin içinde arama yaptığımda bu karakter fazladan varken bazı aramalar ya hiç çıkmıyordu ya da hatalı oluyordu o yüzden bu karakter arama yaparken kaldırmam gerekti.

Bu karakter görebilmeniz için herhangi bir JS konsolda aşağıdaki kod satrı yazmanız yeterli olur.

```
‘ ‘ + ‘İ’.toLowerCase()[1] + ‘ ‘
```

gördüğünüz gibi, küçük harf yapınca iki karakterden oluşuyor ve bu iki karakteri ise farklı gösteriyor.

## 6.3. Html Sayfaların Manipülasyonu

Kullanıcıya uygun ve verimli bir sayfa nasıl göstereceğiz? O kadar basit bir sorunun cevabı oldukça zor! Bu sorunun cevabı için daha önce anlattığım CSS bilgileri ile Javascript sayfa manipülasyon bilgileri ile birleştirerek çözülebilir, ancak biraz düşünmemiz lazımdı. Bazı manipülasyonları dikkat etmeyince kullanıcının deneyim zor olur ve istediğimiz sonucu elde edemeyebiliriz. Basit bir şekilde anlatmaya gerekir ise bu kısımda sayfanın rengi seçilen ve aranan kelimelerin boyaması ve kullanıcıya geri bildirim vermesi gerekiyordu.

Bu sorunları tek tek incelemeye kalkarsam bazı elemanlar iç içe olup işten çıkamıycaz. O yüzden en basit örnekten başlıyorum: boyama. Bir metin boyaması için CSS içinde bu metin hangi renkte olacağını, hangi font kullanılacağını, hangi boyutta olduğunu belirleyebiliriz, ancak genel işaretler (tags) kullandığımda tüm sayfaya etkilemiş oluruz ve bazı gereksiz kısımları ve sayrı yazmamız lazım olur. Bu sorunun

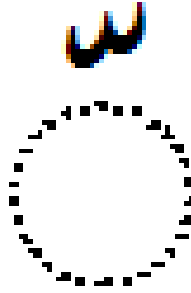
çözümü olarak kendis işaretçim yazıp ve sadece bu işaretçi üzerinde bir değişiklik yapmak en optimal çözüm olduğunu düşünüyorum. O yüzden CSS metni içinde istediğimi bir kısmı tanımlayıp yazdım. Bu kısmı ise <great> diye adlandırdım.

Diğer sorunu ise sadece özel bir buton için nasıl bir değiştirme yapabileceğimi ve bu butonun nasıl geri eski haline getirebileceğimizi de düşündüm, bunun için ise satır içinde yazılan CSS komutlarını kullandım, javascript kullanarak bu buton çağırdım ve sadece bu butonun görünüşünü değiştirince bu şekilde bu sorun çözüldü ve diğer elemanlara herhangi bir değişiklik olmuyor, bu sorunu sayılabilecek ve her yerde kullanılmayan elemanlar için kullandım.

Diğerleri için ise bazı özel elemanlara belli bir isim verip bu isim ile ulaştım, bu şekilde olunca sağdan soldan ya da soldan sağa okunan metinler için bir iyileştirme yapıp ve güzel bir görünüm gösterilir.

#### 6.4. Normalizasyon Fonksiyonları

Bu fonksiyonları Arapça ve Türkçe yazılan metinler için kullandım, bazı metinlerin yazılış farklı oluyor ve okuması veya bilgisayar için göstermesi zor oluyor. Örnek vermek gerekirse arapçadaki yazılan şedde.



Şekil 6.2: Şedde şekli

Aşağıdaki fotoğrafta bulunan iki kelime anlam olarak arasında hiçbir fark yoktur, ama bilgisayar yazılışı için ise şeddenin kelime içindeki yerleri farklıdır.



Şekil 6.3: öğretti kelimesi 1



Şekil 6.4: öğretti kelimesi 2

Yeri her zaman sabit olmuyor, bazen harekeden önce bazen harekeden sonra yazılır ve bu yazılışlar ise arama yapınca bazı zorluklar çıkartır ve kelime aynısı olsa da belli bir sonuç vermez.

Bu sorunu çözmesi için birkaç farklı yöntem kullandım ve denetim için ise en son javascript içinde bulunan normalizasyon fonksiyonunu kullandım, ancak ondan önce kendime göre bir normalizasyon fonksiyonu yazmaya çalıştım, o fonksiyon tüm hareketleri kaldırıyor ve ona göre bir arama yapıyor, ancak bu şekilde olunca ya Kurandaki tüm hareketleri okuyamadım ya bazı kelimeler için kaldırılması gereken şeyler kaldırılmadı ya da özel eliflerde sorun yaşadım, o yüzden bu yöntem hiç doğru değildi.

Ancak normalizasyon yapmak yerine bazı yerlerde metin içinde hareke var olup olmadığını kontrol ettim, bu şekilde hareketli metinler için farklı bir arama yapılabilirdi.

## 6.5. Sayfa Linkleri Manipülasyonu

Daha önce algoritma da anlattığım gibi kullanıcı bir link üzerinde komut verip bu komut programda istenilen bilgileri getirir, ancak bu iş anlatıldığı gibi kolay değil. Nedenini sorarsanız linkler küçük bir fark yada yanlış yapıldığında farklı bir sayfaya götürebilir, o yüzden sayfa değişmeden nasıl sayfayı ve komutları verebiliriz gibi sorular geliyor. Bu sorunu ise hash linki kullanarak çözdüm, her hangi bir link sonunda hash ekleyip okuttum. Bu hash daha önce anlatıldığı gibi bir çok fonksiyon için kullanılır ve işlem yapar.

## 6.6. Web Context Menü

Web tabanlı çalışan programların bağlam menüsü kolay bir şekilde yapılmıyor, ancak tamamen sıfırdan yazması ve tasarlanması gerekiyor. Bu tarz bir tasarım yapması için CSS bilgileri kullandım ve bağlam menüsünü liste olarak gösterip o liste üzerinde değişiklik yaptım, bu durum herhangi bir yere basıldığında neyin basıldığını anlayıp ve arka plandaki çalışan koda haber verip işletmeye yarıyor.

## 6.7. Google Forms

Google kullandığı formal normalde kullanıcıların anket ve sayrı için kullanılmaktadır ancak projemde kullanıcının girdiği veriler programcı ile iletişim kurmak için kullandım. Bu özellik ise Googlein kullandığı post linki kullanarak üzerinde işlem yapıp bir daha kod içinde çağırıyorum, çağırması için ise daha önce anlattığım HTML frames kullanarak yapıyorum. Ancak bu tarz bilgileri ve verileri kullanmak ya da toplamak bazı insanları rahatsız edebilir, o yüzden son versiyondan kaldırdım.

## 6.8. çok Dilli Web Sitesi

Web sitelerin birçok kullanıcıya uygun bir şekilde çalışması için ve çok fazla kişiye erişmesi için birçok dil ile yazıldı, bu şekilde kullanıcı bu siteyi kullanırken istediği dili seçer ve kolay rahat bir deneyim sahibi olur. Sitemizde bu deneyimin sağlaması için seçilen tefsire göre bir dil seçilir, bu dil seçilince daha önce anlattığım HTML tags ile bazı isimler json formatında bulunan bir dosya okuyup o dosya içindeki kelimelerin çevrimi kullanır, kelimeler yazınca sağdan soldan ya da soldan sağa dillerin hususiyeti dikkat ederek yazılmıştır.

## 6.9. Web Tarayıcı Geçmişi

Web tarayıcı geçmişi kaydetmesi kolay ve herkesin alıştığı bir şey ancak yaptığım program tek bir sayfa üzerinde çalıştığı ve farklı sayfalara gitmediğinden dolayı tarayıcı bu durumlar anlamaz ve bir önceki aramaya geri dönemez. Bu sorunu çözebilmek için hash linki kullandık, aynı zamanda her yeni bir arama için sayfanın ismi de değişir, bir şekilde daha sonra neyin aradığınızı ve ne zaman arattığınızı hatta o sayfayı geri açması için kullanıcıya ve tarayıcıya daha kolay ve uygun olur.

## 6.10. Tek Satır Mod

Tek satır modu ise kuran ayetleri kolay ve tek bir sayfada göstermesi için kullanıyoruz, bu şekilde kullanıcının ne yaptığı hangi ayetlerden geçtiğini daha hızlı görüp karşılaştırabilir. Bu tek satır modu küçük ekranlarda aratılan metni anlamlı bir şekilde göstermesi için bulunan metinden önce üç kelime gösterir ve metinden sonra tüm ayeti getirir. Bu şekilde sadece metin değil nereden geçtiğini ve bu ayetten çıkarabileceğimiz anlamlarda gösterir.

## 6.11. İşlemlerin önceliği

Programı geliştirdiğince son zamanlarda eski zamanlar gibi aynı hız ile çalışmadı, bu sorun ise programın açılması için yaklaşık olarak üç dört saniye gecikiyordu, bu tarz bir gecikme çok belli olmadığına göre önemli ve kullanıcı için zaman kazandırır. Bu sorunun çözümünü bulmaya çalışınca benim program, ilk açılışta üç farklı dosya okuyor bu dosyaların önemine göre herhangi bir şey yoktu. Dosyaların kullanıcının ihtiyacına göre okuması yapıldığında çok yüksek bir hız kazandık yaklaşık olarak 70

## 6.12. Arayüz Dönmesi

Bazı zamanlarda bir döngü içinde girildiğinde asenkron olmasına rağmen program döngüden çıkamaz ve farklı işlem yapamaz, bu sebepten dolayı ekran dönebilir ve kullanıcı ile arasındaki iletişim kopar, ayrıca herhangi bir yanıt gelmez. Bu tarz bir sorunun çözmesi için farklı bir thread ile çalışmamız lazımdı, ancak thread kavramı javascript içinde olmadığından dolayı daha önce bahsettiğim web workers kullandım bu özellik ise çok kolay ve basit bir şekilde yapılır.

Web worker ile normal program arasındaki iletişim kurması için normal programlama bilgileri kullandım, bir olay oluşturur bu olayın sonucuna göre hangi fonksiyonların çalışması gerektiğini söyler.

## 6.13. Sayfalama

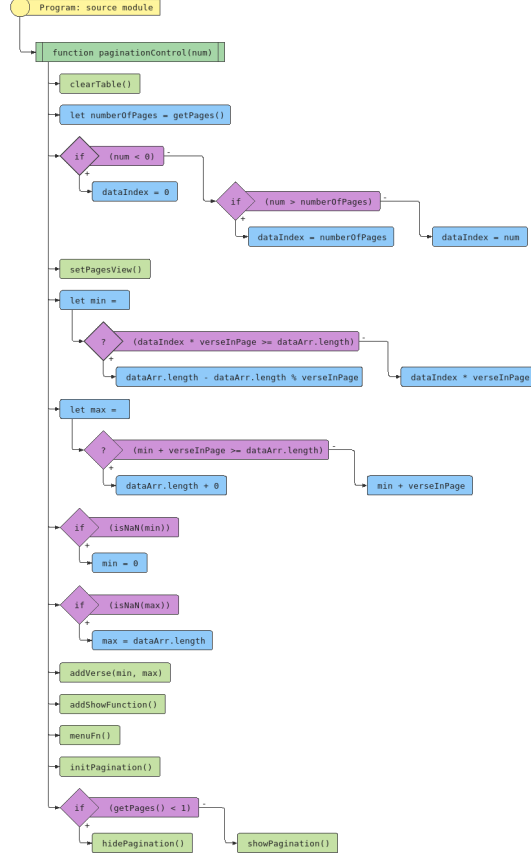
Bazı aramalar yapınca birçok sonuç geliyor, belki binlerce ya da yüzlerce bu tarz sonuç elde etmesi için yaklaşık olarak 20 milisaniye gerekiyordu, ancak programı çalıştırdıktan sonra yavaşlama ve bazı zamanlarda yanıt vermeme durumları oluyordu. Bu tarz durumların incelenmesi için ve çözüm bulması için kodları bir daha kontrol etmesi gerektirdi ve bu sonuçları karşılaştım: Sonuçları alınca herhangi bir sorun yoktu, ancak ekrana yazdırınca bu tarz bilgilerin görselleştirilmesi de bir zorluğa sebep oluyordu ve program hiç çalışmıyordu.

Bu tarz bir sorunun çözümü olarak birkaç yöntem denedim bu yöntemlerinden birisi ise sayfalama sistemiydi. Bir sayfa içinde belli bir sayıda (kullanıcının seçimine göre) sonuç gösteriyor ve sayfanın sonunda sayfaların arasındaki geçiş ya da istediği sayfaya gitmeye için bir seçenek bulunuyor, hangi sayfada olduğunu ve bu sayfanın içinde olan verileri göstermesi için aşağıdaki formül da çözdüm:

$$\text{Sonucuların boyutu} = s \cdot \text{İstenilen ayet sayısı} = i \cdot \text{İstenen sayfa} = K$$

$SayfaSayısı = \frac{s}{i}$  Gösterince de hangi sayfa gösterildiğini ise K diyelim bu şekilde oldu:  $lkayet = K \times i$   $Sonayet = ilkayet + i$

Tabi iki bazı özel durumları oldu, bunları ise aşağıdaki flowchart anlatıyor:



Şekil 6.5: Algoritma grafi

Bu yöntem kullanılıncaya aşırı yavaşlama ve yavaşlama sorunu çözüldü.

#### 6.14. Bilgi Edinme Yöntemleri Information Retrieval Importance And Usage (cosine Similarity, Bm25, Tf,idf)

Simi programı için ayetler arasında ne kadar bir benzerlik olduğunu ve nasıl bir şeye benzediğini kontrol etmesi için bilgi edinme ve benzerlik yöntemlerini kullandım, bu yöntemlerden birisi kosinüs benzerlik yöntemi oldu. Aşağıda çalışma şeklini gösterdim:

$$\cos(\mathbf{t}, \mathbf{d}) = \frac{\mathbf{t} \cdot \mathbf{d}}{\|\mathbf{t}\| \|\mathbf{d}\|} = \frac{\sum_{i=1}^n t_i d_i}{\sqrt{\sum_{i=1}^n (t_i)^2} \sqrt{\sum_{i=1}^n (d_i)^2}} \quad (6.1)$$

T= terim D= ayet/döküman

Bu fonksiyon ise ayetlerin kök vektörü ile kontrol ediliyor, onu geliştirmesi için terim frekansı ile ters doküman frekansı hesaplayarak bir daha hesaplatınca daha iyi sonuçlar verir.

$$tf(t, d) = \frac{\text{count of term1}}{\text{number of terms in document}} \quad (6.2)$$

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (6.3)$$

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) (\text{cosineTFDF}) \quad (6.4)$$

Ters doküman frekansı için birçok yöntem var, burada tercih ettiğim yöntem en çok kullanılanlardan birisi.

Terim frekansı ise bir ayette terimin frekans ile tüm terimlerin sayısı ile bölünerek bulunuyor.

Ters doküman frekansında ise bir kelime ne kadar çok tekrarlanırsa o kadar az bir değer veriliyor, eğer kelime dokümanda çok fazla kullanılmışsa o zaman benzeri az değer veriyor yani sayısı ile benzeme oranı ters orantılı.

Bir de onun üstünde simetrik çıkması için ve aynı sonuçları verebilmesi için algoritma başında hangi vektörün daha küçük olduğunu kontrol edip o vektöre göre hesaplama yapıyorum. Bu şekilde matris her zaman simetrik oluyor, aynı zamanda BM25 diye bir algoritma kullanarak algoritma text için en verimli olan verileri nereden alacağımı görüyorum ve bu algoritma ise aşağıdaki denklem ile yapıldı:

$$\sum_i^{n} IDF(q_i) \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

Şekil 6.6: BM25 algoritması

## 6.15. Sürekli Kaydırma

Oku programı için sürekli kaydırma özelliği getirmek önemli bir şey olduğunu düşünüyorduk, bu şekilde hem Kuran okuyunca ne kadar zaman harcadığı hem hissetmiyorsun hem de dönmelerde çok fazla zaman kaybedilmiyor. Ancak bunun gibi bir özellik getirebilmesi için ekran ve cihazın sürekli çalışması ve programın hiç kapanmamasını düşünmem gerekiyordu.

Kuran tek getirip hemen ekrana bastırınca çok büyük olduğun için program kapanıyordu, ancak bu sorunun çözümü olarak aynı anda en fazla üç adet süre göstererek yapıldı, bu şekilde ekran ve program hiç dönmüyor ve sürekli çalışır halde oldu.



## 6.16. Event Programlama Dilleri

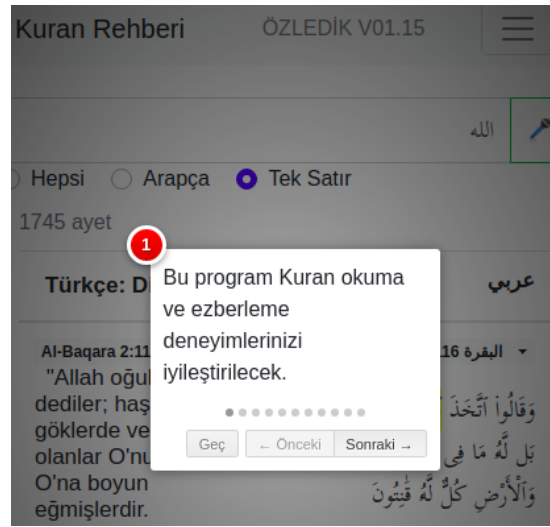
Javascript gibi bir programlama dili ile çalıştığım için en çok karşılaştığım sorun ise undefined element sorun, bu sorun programın akışı hızlı çalışması için satır satır çalışmıyor beklediğim fonksiyon belki erken çalıştırıp ve oradan çıkması beklediğim ve bir önceki hesaplamalara bağlı olan sonuç yanlış çıkabilirdi. Bunun gibi bir sorunu çözmesi için birçok çözüm var bunlardan en önemlisi ise await ve promise kullanmak, bunlarıda Javascript gücü olarak sayılır. Daha önce detaylı bir şekilde anlatmışım.

## 7. Uygulamalar:

Bu bölümde uygulamaların nasıl çalıştığını ve nasıl kullanılabileceğini kullanıcıya anlatıyorum.

### 7.1. Ara

Ara bir arama motorudur. Aynı kelimeyi direkt aratabilirsin, ilk açılışta F1 tuşu basarak bir tür menüs açılır ve program içinde gezerek interactive bir şekilde programı anlatır.



Şekil 7.1: Tur girişi

Programı tamamen anlatır. Bundan ziyade programın linkinde bazı kuralları yazabilirsin ve arama başlatabilirsin. Aşağıda nasıl olduğunu gösterdim:

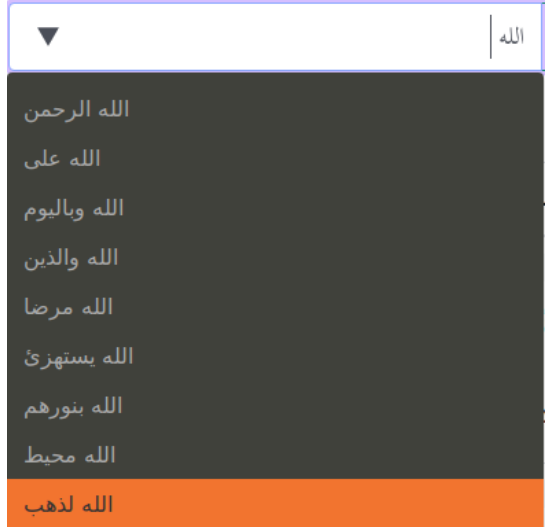
[/finder#b=Allh](#)

Şekil 7.2: Link komutları

Girebileceği komutlar ise aşağıdaki tabloda belirtmektedir (table) r kelimesinin kökü girilir b buckwalter olarak aranması gereken kelime t kelimenin tefsirede araması w kelimenin kendisi yazılır

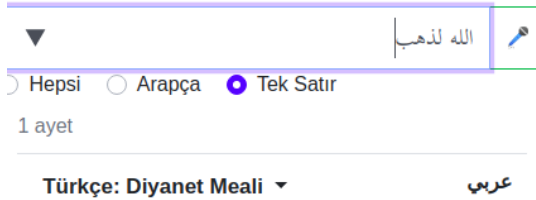
komut	açıklama
r	kelimesinin kökü girilir
b	buckwalter olarak aranması gereken kelime
t	kelimenin tefsirede araması
w	kelimenin kendisi yazılır

Program açılınca gördüğünüz gibi sadece arama yeri ve sonuçları veriliyor. Onun dışında arama yerinde herhangi bir harf yazınca bu harfe göre Kurandaki bulunan kelimeler gösterilir ve tavsiye edilen kelimeler çıkar.

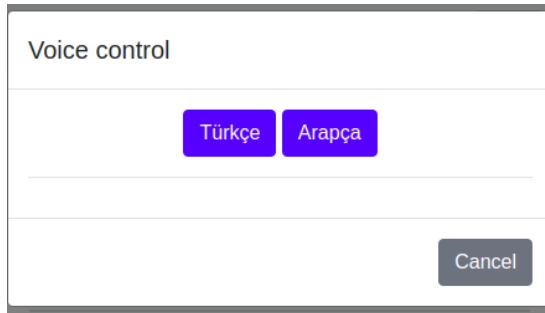


Şekil 7.3: Metin tamamlama

Arama yeri yanında bir konuşma butonu vardır ve bu buton kullanılarak konuşarak bir arama başlatabilirsiniz. Bu konuşma ise tefsire seçilen dili ve arapça olarak iki seçenek göstermektedir.



Şekil 7.4: Ses butonu



Şekil 7.5: Ses seçenekleri

Sayfanın sonunda ise bir sayfalama sistemi bulunuyor.



Şekil 7.6: sayfalama sistemi

Arapçada herhangi bir kelime seçerek sağ tıklayıp context menü çıkıyor, bu context menü üzerinde bu tarz seçenekleri var:Arama, googleda ara, kok ile arat, kelime kopyala

En son süre yerindeki sağ tarafta bulunan ok kullanarak birçok kaynağa gidebilir, bu kaynakları ise en son kullanılan süre butonu olur ve bu şekilde daha hızlı bir şekilde ulaşabiliriz.



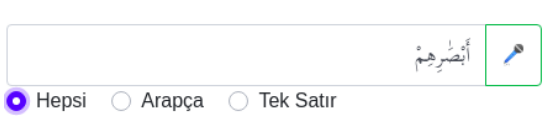
Şekil 7.7: çok kaynak

Kullanıcı, hangi tefsir istediğini bir listede bakabilir ve değiştirebilir; bu liste şu anda 4 farklı tefsir ve dil içermektedir.



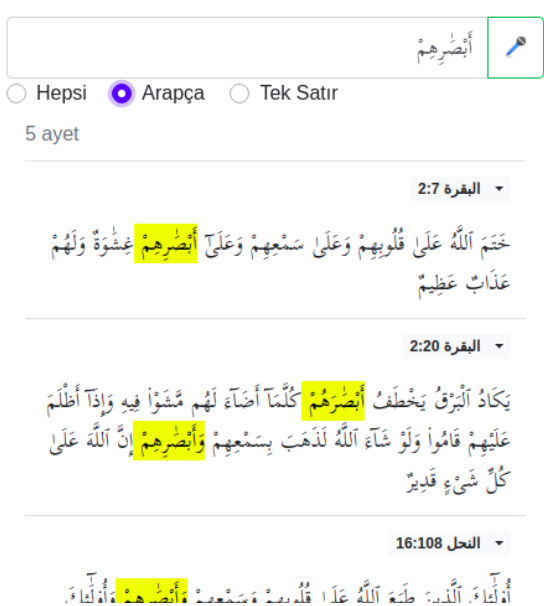
Şekil 7.8: Tefsire seçenekleri

Aynı zamanda görselleri daha uygun ve basit bir şekilde olması için aşağıda üç seçenek bulunur, onlar hemen arama yerinin altında yer alır.



Şekil 7.9: tek satır seçenekleri

Sadece Arapça ise sadece Kuranı gösterir ve tefsiri gizler:



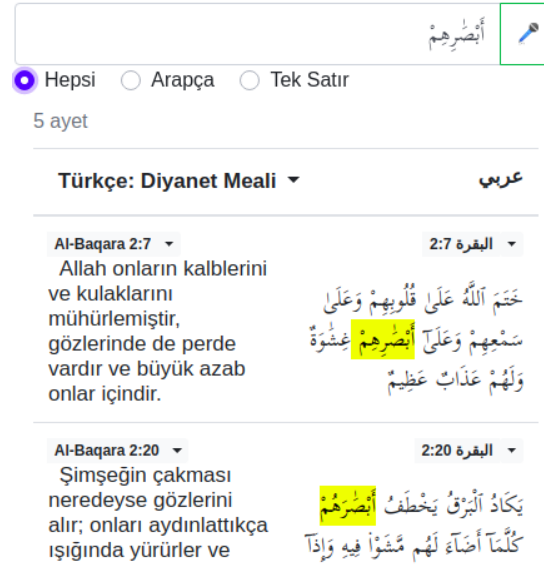
Şekil 7.10: sadece arapça seçeneği

Tek satır ise Arapçayı sadece bulunan kelime arkasına üç kelime getirerek ve bir satıra sığacak şekilde yapar.



Şekil 7.11: Tek satır seçeneği

Tüm seçimde tüm Arapça metin ve tefsir metni bir yerde gösterir.



Şekil 7.12: Tefsire gösterme

Programın küçük ve büyük ekranları arasında farkı aşağıdaki fotoğrafları göstermektedir:

Kuran Rehberi ÖZLEDİK V01.15

الْحَمْدُ لِلَّهِ

Hepsi  Arapça  Tek Satır

23 ayet

Türkçe: Diyanet Meali عربي

Al-Faatiha 1:2 الفاتحة 1:2  
Hamd, Alemlerin  
Rabbi Allah'a  
mahsustur.

Al-An'aam 6:1 الأنعام 6:1  
Hamd, gökleri ve yeri  
yaratan, karanlıkları ve  
aydınlığı vareden  
Allah'a mahsustur.  
Öyle iken, inkar  
edenler Rablerine  
başkalarını eşit  
tutuyorlar.

الْحَمْدُ لِلَّهِ الَّذِي خَلَقَ  
السَّمَوَاتِ وَالْأَرْضَ وَجَعَلَ  
الظُّلُمَاتِ وَالنُّورَ ثُمَّ الَّذِينَ  
كَفَرُوا بِرَبِّهِمْ يَعْدِلُونَ

Şekil 7.13: küçük ekranları

Kuran Rehberi ÖZLEDİK V01.15 Ana Oku Sili İletişim Ayarlar Diğer

Hepsi Arapça Tek Satır

1 ayet

Türkçe: Diyanet Meali عربي

الْحَمْدُ لِلَّهِ الَّذِي خَلَقَ السَّمَوَاتِ وَالْأَرْضَ وَجَعَلَ الظُّلُمَاتِ وَالنُّورَ ثُمَّ الَّذِينَ كَفَرُوا بِرَبِّهِمْ يَعْدِلُونَ

Şekil 7.14: büyük ekranları

## 7.2. Oku

Oku programı Kuran okumasını kolaylaştıran ve tefsiri gösteren bir programdır, bu program sadece ve sadece iki yerde oluşmaktadır, bu yerler Kuran ve tefsiri göstermektedir.

Kuran Rehberi ÖZLEDİK V01.15 Ana Oku Sili Diğer

1 Go to Sura

The Opening (Al-Faatiha) تفسير القرآن

1- Rahman ve Rahim olan Allah'ın adıyla;  
2- Hamd, Alemlerin Rabbi Allah'a mahsustur.  
3- O Rahmet ve Rahimdir.  
4- Din Gününün salıhtır.  
5- Ancak Sana kâilül eder ve yalnız Senden yardım alırım.  
6- Biz doğru yolda erdir.  
7- Nîmetle endişgin kimselerin yoluna; gazaba uğrayanların, ya da sapmalarını yoluna değil.  
1- Elif, Lam, Mim.  
2- Bu, doğruluğu yâğha görmeyen ve Allah'a karşı gelmekten sakınmayan yâğı gönen Kitâb'tır.  
3- Önter, gâşbe harurur, ramazn kılar, kendilerine verdîğmiz rızıkta yenî yemice sarf ederler.  
4- Önter, sana indiren Kitâb'a da, senden önce indirilenlere de inanır; ahirete de yalnız önter kesimle inanır.  
5- İyâ Rablerinin yolunda olanlar ve saadete erişenler bunlardır.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ (1)  
الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ (2)  
الرَّحْمَنِ الرَّحِيمِ (3)  
مَلِكِ يَوْمِ الدِّينِ (4)  
إِذَا نُنَادَى يَوْمَ الْقِيَامَةِ (5)  
كَذَّبْنَا كَذَّبَتْ أَهْلُ الْمَدِينِ (6)  
سِرًّا أَوْ أَبْرًا فَاتَّبَعَ اللَّهُهُمْ وَقَدْ كَفَرُوا رَبَّهُمْ فَلَا كَيْفَ لَهُمْ (7)  
لَهُ الْعِشْقَانِ (8)

Şekil 7.15: Ekran görüntüsü

### 7.3. Simi

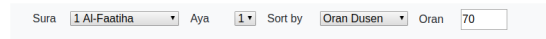
Simi programı için de basit bir şekilde arayüz tasarlamaya çalıştım, bu arayüz kullanıcıya dört farklı seçenek verir.

Ayet : hangi ayet gösterileceğini

Süre : hangi sure olacağını

Benzerlik oranı : gösterilen ayetlerin benzerlik oranı

Sıralama şekli : Benzerlik ya da ayet sırasına göre dört seçenek vardır, yükselen ya da düşen olarak seçilebilir.



Şekil 7.16: Simi menüsü

Aynı zamanda link her zaman değişiyor, kullanıcı isterse linkten istediği ayet gider, isterse arayüzü kullanarak gider.

</simi.html#1:1>

Şekil 7.17: Simi Linki

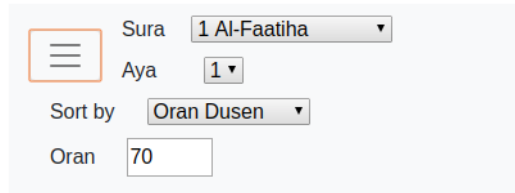
Aşağıdaki fotoğraf örnek bir sonuç listesi göstermektedir:





Şekil 7.18: Fatiha suresi ilk ayetin Sonuçları

Burada ise küçük ekran görünümü ve büyük ekran görünüm arasındaki fark:



Şekil 7.19: Küçük ekran menüsü

## 8. Gelecek Yönelik çalışmalar

Bu bölümde geleceğe yönelik çalışmalar neyin daha iyi olacağını ve daha iyi olması için ne yapılabileceğini anlatıyorum, bu tarz çalışmalarını Allah izin verirse ilerleyen yıllarda devam edeceğim.

### 8.1. Konu Bazlı Arama

Şu anki kullanıcı sadece kelime ile arama yapabilir, program kendisi hangi konu işlendiğini ve nasıl olduğunu anlamaz. Konu bazlı arama geliştirme gerektirmektedir, kullanıcı istediği konuyu girerek bu konunun Kuranda var olup olmadığını kontrol ederek çıkarması gerekmektedir, bu tarz bir geliştirme ise onunla birlikte gelen hız boy ve arayüz zorluğuna sebep olabilir. Bunların çözümüne yönelik bir çalışmalar ilerleyen dönemlerde yapılacaktır.

### 8.2. Bilgi Edinme Ve Bilgi çıkartma

Şu anda kullanıcının girdiği bilgiye göre tek bir şekilde bilgi edinme ve benzerlik oranı çıkarılıyor algoritmaların arasındaki oranlar hangisi daha iyi olduğunu ve nasıl geliştirebileceğimizi ve araştıracağımızı, bu tarz bir gelişmeler daha önce yapılan araştırmalara göre yapılacaktır. Belki de farklı bir sistem geliştirmemize sebep olabilir. Onunla ilgili detaylar için İngilizcede information retrieval ve information extraction olarak geçen bazı algoritmaları araştırmam gerekecek.

Bu tarz algoritmaları en iyi şekilde geliştirmesi için bulunan Arapça bilgileri kullanarak yapılacaktır, kelimelerin niteliklerini ve türlerini anlatan bir program de eklenebilir.

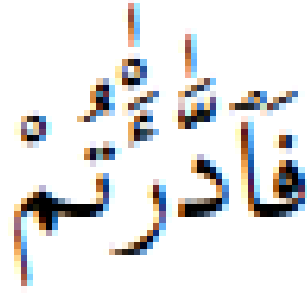
### 8.3. Test

Büyük bir program geliştirince mutlaka test yöntemleri bulunması gerekiyor ve yeni bir fonksiyon eklendiğinde bir hata çıkmasını diye bakmamız gerekiyor, bu tarz testleri makine olarak bir değer verir ve fonksiyonları beklentilere göre çalışır. Bazı kütüphaneleri vardır, ancak bu projede manual yapılmaktadır, daha sonra bu testlerin nasıl otomatikleştirebileceğinin ve nasıl daha iyi olacağını araştırılması gerekiyor yeni bir şey ekleyince eski kodların sabit ve kalıcı olabilmesi için.

### 8.4. Font

Arapçanın fontu hem bilgisayarlara geç gelen bir şey hem de fontun çok farklı yazılışları olduğundan dolayı tamamen kapsamıyor bu alanla ilgili Kuranın nasıl daha iyi gösterileceğini ve nasıl daha güzel yaptıracağımız için bir font araması olacak, eğer istediğimiz fontun bazı özel durumları olursa o durumlarda nasıl geliştireceğimize bakmamız lazım.

Font meselesi neden mi önemli? Çünkü yanlış okumalara sebep olabilir, aşağıda bir kaç örnek veriyorum:



Şekil 8.1: idarattum kelimesi

Yukarıda gördüğünüz kelime içinde bir hemze var, ancak her font bu hemzeyi gösteremedi, o yüzden hem yanlış okuma sebebi olabilir, hem de bazı kelimelerde özel harfler var bu harfler normal fonta gösterilmiyor. Aşağıdaki namaz kelimesi onların bir örneği:

الصلاة

الصَّلَاةَ

الصَّلَاةَ

الصَّلَاةَ

Şekil 8.2: namaz kelimesi

Font sorunu görsel ve kolaylaştırma için daha iyi bir şekilde olması için yapılacak bir geliştirmedir.

### 8.5. Bot

Kuran da bulunan metinlere göre sosyal medyada paylaşılacak bir bot yazılabilir, bu bot belli zamanlarda belli konulara göre bilgi paylaşır bu şekilde insanlar bu bilgi ile ilgili düşüncelerini ve fikirlerini paylaşabilir. Bot kendi kendine bir tez olabilir, o yüzden şu anda çok fazla anlatmayacağım.

**9. Elhamdülillah**

Bu şekilde güzel bir dönem ve programı anlatmış oldum, inşallah faydalı olur ve insanlar bu sayede güzel fikir sahibi ve düşünce sahibi olabileceklerini düşünürler. Konu basit olarak gözüke bile gördüğünüz gibi birçok teknoloji ve birçok zorluğu sebep oldu. İnşallah ileriki günlerde çalışanlara bu tarz araştırmalar faydalı olur ve insanlara verimli bir şey çıkar.

## 10. KAYNAKLAR

- [1] J. Wakefield, "Phone and internet use: Number of mobile calls drops for first time," Aug 2018. [Online]. Available: <https://www.bbc.com/news/technology-45033302>
- [2] E. Schmidt, A. Eagle, and J. Rosenberg, *Google: how Google works*. Grand Central Publishing, 2017.
- [3] "Speechrecognition." [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition>
- [4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107,117, 1998.
- [5] J. Hartnell, "The continuous page," *Continuous Page: Scrolls and Scrolling from Papyrus to Hypertext*, Jun 2019.
- [6] "Jsdocs." [Online]. Available: <https://jsdoc.app/>
- [7] "Learn, build, & test regex." [Online]. Available: <https://regexr.com/>
- [8] "Markdownish syntax for generating flowcharts, sequence diagrams, class diagrams, gantt charts and git graphs." [Online]. Available: <https://mermaidjs.github.io/#/>
- [9] A. Mehrabani, "Intro.js." [Online]. Available: <https://introjs.com/>
- [10] M. Otto and J. Thornton, "Bootstrap." [Online]. Available: <https://getbootstrap.com/>
- [11] S. Robertson and H. Zaragoza, *The Probabilistic relevance framework: BM25 and beyond*. Now Publishers, 2009.
- [12] "The html presentation framework." [Online]. Available: <https://revealjs.com/>
- [13] "Githubpages." [Online]. Available: <https://pages.github.com/>