İçindekiler

[Dependencies (Bağımlılıklar) 2](#_Toc139163443)

[Proje Mimarisi 3](#_Toc139163444)

[Proje Dosyaları 5](#_Toc139163445)

# Dependencies (Bağımlılıklar)

* "next": "13.4.7",
* "react": "18.2.0",
* “react-dom": "18.2.0",
* "sass": "^1.63.6",
* "@fortawesome/free-solid-svg-icons": "^6.4.0",
* "@fortawesome/react-fontawesome": "^0.2.0",

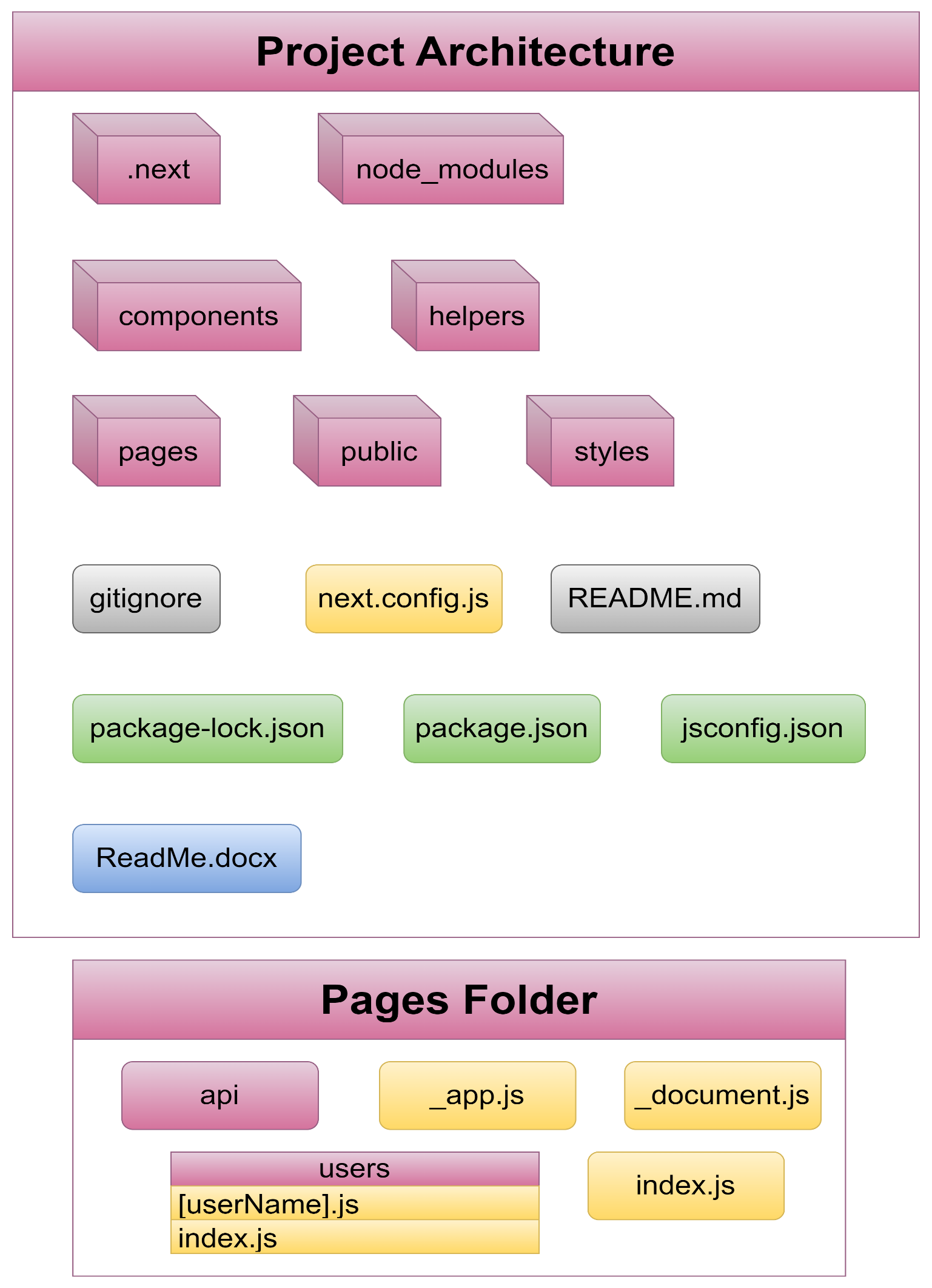
VS Code programında Terminal içerisinde:

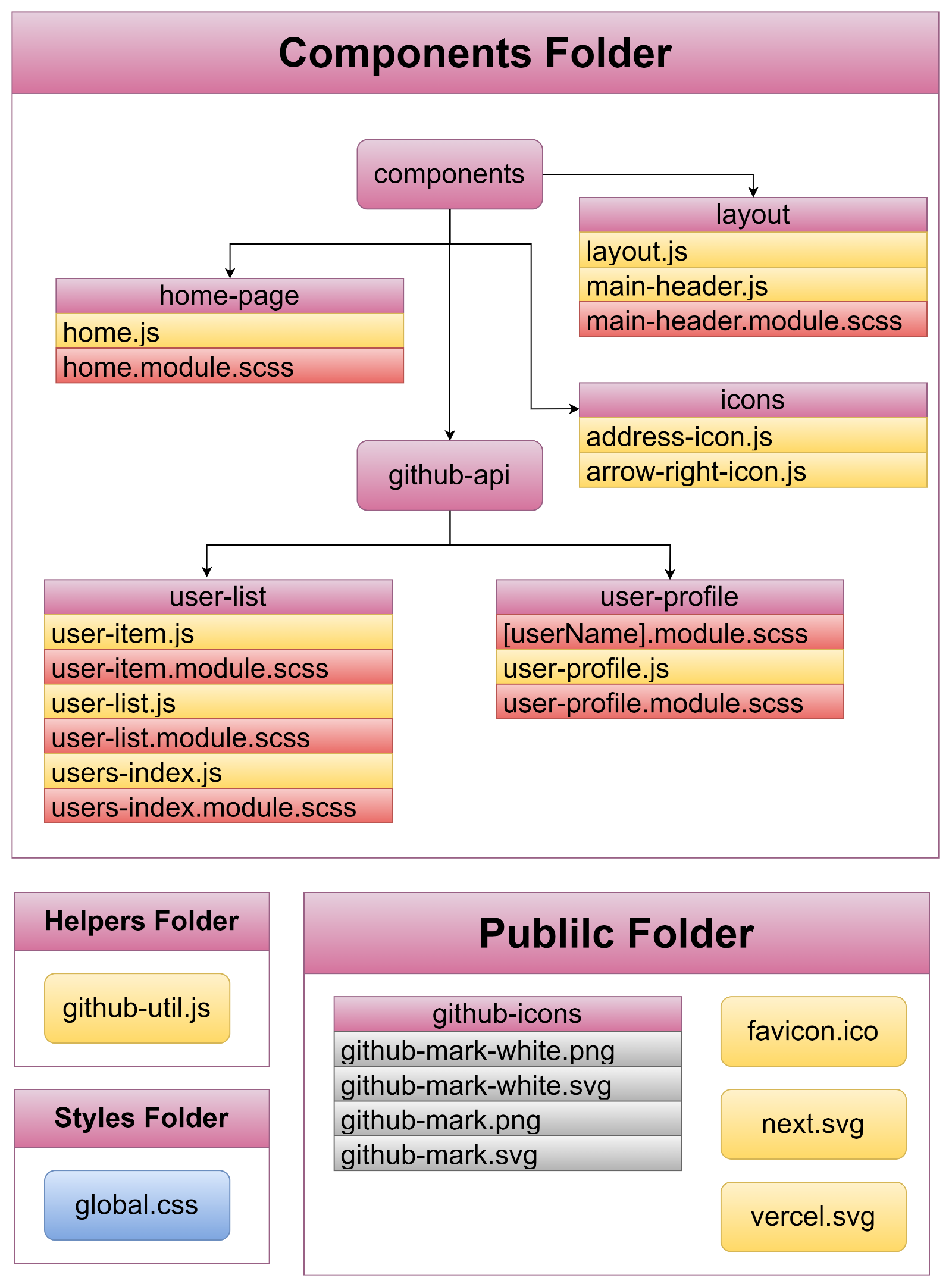
* “npx create-next-app” komutu ile NextJS framework kurulumu yapıldı. Bu kurulum işlemi ile react, react-dom ve next paketleri yüklendi.
* “npm install sass” scss dosyalarını kullanabilmek için SASS(Syntactically Awesome Style Sheets) paketi yüklendi.
* "npm install @fortawesome/free-solid-svg-icons" ve "npm install @fortawesome/react-fontawesome" komutları ile ikonların kullanılabilmesi için font paketleri yüklendi.

VS Code programında Terminal içerisinde sık kullanılan komutlar:

* “npm run build” komutu ile proje build edilir.
* “npm run start” komutu ile proje başlatılır.
* “npm run dev” komutu ile proje çalıştırılır.
* Ctrl + C komutu ile proje durdurulur.

# Proje Mimarisi





# Proje Dosyaları

**NOT:** SSR (Sever-side Rendering) işlemi için ilgili sayfalarda getServerSideProps fonksiyonu

kullanıldı. İlgili sayfalarda fortawesome kütüphanesi, components/icons klasörü içerisinde .js

dosyaları, helpers/github-util.js dosyasının içerisindeki fonksiyonlar, public/github-icons

klasörünün içerisindeki .png ve .svg dosyaları import edilerek kullanıldı.

* **pages/index.js:**
  + IndexPage fonksiyonu default olarak export edildi.
  + home.js dosyasının içeriğini render edebilmek için <HomPage> import edildi ve <Fragment> JSX etiketi içerisine alındı. Fragment sayesinde extra DOM kullanmadan <HomePage> ile gelen birden fazla elemanı bir arada gruplandı.
* **styles/global.css:**
  + \_app.js dosyasına import edilerek proje genelinde geçerli olan CSS düzenlemeleri bu dosya içerisinde tanımlandı.
* **components/layout:**
  + layout.js:
    - main-header.js dosyasındaki içeriğin render edildiği kısımdır. \_app.js dosyasına import edilerek diğer içeriklerin Layout içerisinde render edilebilmesini sağlar. Bu işlem için <main>{props.children}</main> kod satırı ile alt bilgilerin gösterilebilmesi sağlanır.
  + main-header.js:
    - layout için sabit ve bazı sayfa yönlendirmeleri ile arama işlemlerini içeren kod bloklarının yazıldığı kısımdır.
    - handleSearch fonksiyonu ile arama çubuğuna yazılan bilgileri yakalayarak kişinin profil sayfasına yönlendirme işlemini gerçekleştirir.
    - CSS bilgilerini main-header.module.scss dosyasından almaktadır.
* **pages/\_app.js:**
  + Tüm sayfaların Layout içerisinde gösterilmesi için <Layout><Component> … şeklinde düzenleme yapıldı
* **public/github-icons:** main-header.js ve home.js dosyaları için kullanıldı.
* **components/home-page:**
  + home.js:
    - Ana sayfa tasarımı bu dosya içerisinde tanımlandı. Github kullanıcı listesi sayfasına geçiş işlemi için next/link kütüphanesi kullanıldı.
    - CSS bilgilerini [home.module.scss](http://home.module.scss) dosyasından almaktadır.
* **pages/users:**
  + index.js:
    - github-util.js dosyasından getUsers fonksiyonu getServerSideProps fonksiyonunun içinde kullanıldı.
    - user-list.js ve users-index.js dosyalarındaki bilgiler render edildi.
    - handleGoToPage, handleNextPage, hadlePreviousPage, handlePageNumberChance fonksiyonları bu dosya içerisinde tanımlandı.
      * Bu fonksiyonlar sayfa numarası değiştiğinde, sayfa numarası girildiğinde, önceki ve sonraki sayfalara gidilmek istendiğinde ilgili sayfaya yönlendiren fonksiyonlardır.
  + [userName].js:
    - Dinamik javascript dosyası olarak düzenlendi (ex: [dynamic].js).
    - user-profile.js dosyasındaki bilgiler render edildi.
    - getServerSideProps fonksiyonu ile api response bilgileri kontrollü bir şekilde profil sayfası olarak düzenlendi.
    - CSS düzenlemeleri [userName].module.scss dosyasından alındı.
* **components/github-api:**
  + user-list:
    - user-item.js:
      * Kullanıcı bilgilerinde gösterilmesi istenen bilgiler düzenlendi, kullanıcının site içerisindeki profil sayfasına yönlendirilebilmesi için buton düzenlendi.
      * CSS düzenlemeleri user-item.module.scss dosyasından alındı.
    - user-list.js:
      * user-item.js dosyasındaki bilgiler içeri aktarıldı. Gelen bilgiler map fonksiyonu ile birlikte liste halinde görüntülenebilecek formatta olması için düzenlendi.
      * CSS düzenlemeleri user-list.module.scss dosyasından alındı.
    - users-index.js:
      * users/index.js sayfasında render edilmek istenen bilgilerin display formatı hazırlandı
      * handleGoToPage, handleNextPage, hadlePreviousPage, handlePageNumberChance fonksiyonları ve page, pageNumber değerleri parametre olarak bu dosyada kullanıldı.
      * CSS düzenlemeleri için users-list.module.scss dosyası kullanıldı.
  + user-profile:
    - user-profile.js:
      * [userName].js dosyası için render edilmesi istenen bilgiler bu dosyada düzenlendi.
      * Kullanıcının github profil sayfasına yönlendirebilmesi için buton düzenlendi.
* **helpers/github-util.js:**
  + perPage, page, since parametrelerini içeren getUsers fonksiyonu tanımlandı.
    - Github api üzerinden sayfada gösterilecek kullanıcı listesini getiren bir fonksiyondur.
  + {username} parametresi içeren findUser fonksiyonu tanımlandı.
    - Github api üzerinden kullanıcı adı ile eşleşen kullanıcının profil bilgilerini getiren bir fonksiyondur.