

به نام خدا

# مستندات کد منبع

## EXTRA TIME

پروژه مهندسی نرم افزار

زمستان 99

محمد حسین شریعتی پور

پوریا ذوالفقاری

سجاد شهرابی

محمد رضا نظری

# فهرست

۱	مقدمه
۲	ساختار درختی پروژه
۳	فایل index.html
۳	کامپوننت‌ها
۴	کامپوننت App.js
۵	Style کامپوننت ها
۷	dependency ها
۸	bootstrap
۸	swiper
۱۰	Fortawesome
۱۱	استفاده از API ها

## مقدمه

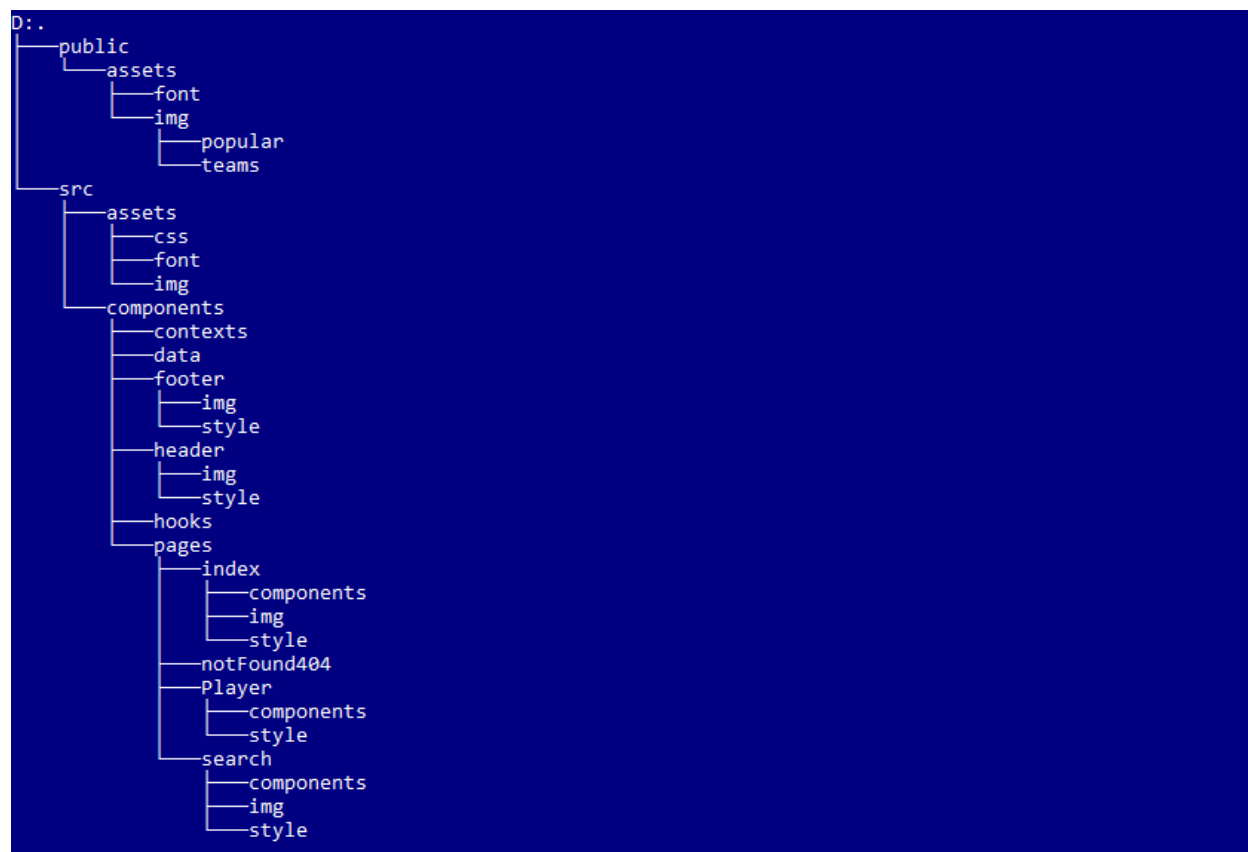
پروژه پیاده سازی شده یک اپلیکیشن تحت وب می باشد که به کمک کتابخانه React.js توسعه پیدا کرده است.

دیتاست مورد استفاده در این پروژه شامل اطلاعات بیش از 17 هزار بازیکن بازی FIFA 2021 می باشد که به کمک این اپلیکیشن می توان اطلاعات هر یک از این بازیکنان را بصورت بصری مشاهده نمود.

از دیگر خدمات این اپلیکیشن می توان به موارد زیر اشاره نمود:

- جستجو در میان بیش از 17 هزار بازیکن
- افزودن بازیکنان به لیست علاقه مندی
- مشاهده برترین بازیکنان در هر پست
- قابلیت مرتب سازی بازیکنان بر حسب قدرت و سن
- و ...

## ساختار درختی پروژه



Public: این پوشه شامل فایل های استاتیک مانند فونت ها، لوگو و... برنامه می باشد.

Src: این پوشه شامل فایل های داینامیک مانند کامپوننت ها، استایل های هر کامپوننت و... می باشد.

src/components : محل قرار گیری کلیه کامپوننت های نوشته شده می باشد، هر کامپوننت یک پوشه به نام خودش دارد.

src/components/contexts: محل قرارگیری stste های سراسری

## فایل index.html

در این فایل HTML یک div با id="root" وجود دارد. این عنصر، عنصری است که کل برنامه ی ما درون آن قرار می گیرد:



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <title>EXTRA TIME</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>
</html>
```

مواردی همچون تعریف متاتگ ها، عنوان سایت نیز در این فایل نوشته می شود.

## کامپوننت ها

کامپوننت ها قسمت اصلی هر اپلیکیشن توسعه داده شده توسط React js می باشند. کلیه کامپوننت های نوشته شده در این پروژه بصورت Functional Components هستند.

هر کامپوننت شامل یک فایل با فرمت JS می‌باشد که شامل موارد زیر می‌باشد:

1. در ابتدای فایل کلیه کتابخانه‌ها، داده‌ها، style ها و سایر نیازمندی‌ها برای پیاده سازی هر کامپوننت import می‌شوند.

2. بدنه کامپوننت که در واقع یک تابع JavaScript می‌باشد.

3. Export کردن کامپوننت برای استفاده آن در سایر کامپوننت‌ها.

سورس کد زیر نمونه‌ای از کامپوننت نوشته شده می‌باشد:

```
import React from 'react'
import ContainerPage from "../ContainerPage"
import {Link} from "react-router-dom";

const NotFound = (props)=>{
  return(
    <ContainerPage playerName="player">
      <div className="not-found-player">
        <div className="img">
          
        </div>
        <h3 className="font-title color-brown">
          Page Not Found
        </h3>
        <Link to="/" className="btn-home">Go to home</Link>
      </div>
    </ContainerPage>
  )
};

export default NotFound;
```

## کامپوننت App.js

خارجی ترین کامپوننت ما می‌باشد که سایر کامپوننت‌ها در داخل این کامپوننت مورد استفاده قرار می‌گیرند. این کامپوننت در فایل index.js به فایل اصلی برنامه یعنی index.html که در پوشه Public قرار دارد اضافه می‌شود:



```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import './assets/css/init.css';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

reportWebVitals();
```

## Style کامپوننت ها

Style ها بخش مهمی از پیاده سازی هر کامپوننت محسوب می شوند که به هر کامپوننت جلوه و ظاهری کاربر پسند می دهند. معمولا برای نوشتن style ها از CSS که یک زبان نشانه گذاری محسوب می شود استفاده می شود.

امروزه فریمورک ها و کتابخانه های مختلفی برای سهولت استفاده از CSS در پروژه ها توسعه داده شده است. در این پروژه نیز برای سهولت کار کلیه کدهای مربوط به CSS به صورت SCSS که یک پیش پردازنده برای CSS محسوب می شود استفاده شده است. به کمک این پیش پردازنده امکان تعریف مواردی مانند متغیرها، توابع و موارد مختلف دیگری در CSS فراهم می شود. کدهای نوشته شده بصورت SCSS را ابتدا به CSS کامپایل کرده و سپس از آن ها در کامپوننت ها استفاده خواهیم کرد.

واکنش گرا بودن کامپوننت‌ها موضوع مهم دیگری است که امروزه با توجه به گسترش استفاده از موبایل‌ها و تبلت‌ها در جامعه باید به آن دقت نمود. در CSS از مدیا کوئری‌ها بدین منظور استفاده می‌شود.

پروژه شامل دو فایل SCSS مشترک برای استفاده در سایر فایل‌های SCSS می‌باشد:

1. `responsive.scss`: شامل کدهای مربوط به استفاده از مدیا کوئری‌ها

2. `variables.scss`: شامل متغیرهای سراسری مانند رنگ‌ها، تابع‌های سراسری



```
$darkBlue : #255965;  
$white : #FFFFFF;  
$brown : #746223;  
$dark: #707070;  
$gold: #FADA5E;  
  
$boxShadowWhite : 5px 12px 20px rgba(36,37,38,.13);  
$boxShadowWhite2: 0 1px 5px rgba(0, 0, 0, 0.15);
```

این دو فایل در کلیه فایل‌های SCSS دیگر `import` می‌شوند تا مورد استفاده قرار گیرند:




```
@import "../assets/css/responsive";  
@import "../assets/css/variables";
```

علاوه بر این دو فایل برای کامپوننت‌های مربوط به هر یک از صفحات سایت یک فایل `style` جدا تعریف می‌شود همچنین یک فایل `style` برای استفاده مشترک در کلیه صفحات نوشته شده است که در خارجی‌ترین کامپوننت برنامه یعنی `App.js`، فراخوانی می‌شود (`init.scss`).



## dependency ها

dependency های استفاده شده در پروژه در فایل package.json مشخص شده است:



```
{
  "name": "fifa",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@fortawesome/fontawesome-svg-core": "^1.2.32",
    "@fortawesome/free-solid-svg-icons": "^5.15.1",
    "@fortawesome/react-fontawesome": "^0.1.12",
    "@testing-library/jest-dom": "^5.11.5",
    "@testing-library/react": "^11.1.2",
    "@testing-library/user-event": "^12.2.2",
    "bootstrap": "^4.5.3",
    "react": "^17.0.1",
    "react-bootstrap": "^1.4.0",
    "react-dom": "^17.0.1",
    "react-id-swiper": "^4.0.0",
    "react-input-range": "^1.3.0",
    "react-loader-spinner": "^3.1.14",
    "react-router-dom": "^5.2.0",
    "react-scripts": "4.0.0",
    "react-select": "^3.1.1",
    "swiper": "^6.3.5",
    "universal-cookie": "^4.0.4",
    "web-vitals": "^0.2.4"
  },
}
```

## bootstrap

این کتابخانه شامل مجموعه ای از کلاس های آماده CSS می باشد که به منظور سهولت و تسریع در فرآیند توسعه برنامه از آن استفاده شده است.

برای استفاده از کلاس های این کتابخانه کافیست فایل CSS این کتابخانه را در کامپوننت مورد نظر import کنیم سپس نام کلاس های موجود در این کتابخانه را در عناصر مختلف قرار دهیم. با توجه به اینکه می خواهیم از کلاس های این کتابخانه در همه کامپوننت ها استفاده کنیم، این کتابخانه را در خارجی ترین کامپوننت (App.js) فراخوانی می کنیم.



```
import 'bootstrap/dist/css/bootstrap.min.css';
```

## swiper

از این کتابخانه جهت پیاده سازی اسلایدرهای واکنش گرا استفاده می شود. برای استفاده از این کتابخانه ابتدا کامپوننت های مختلفی که در این کتابخانه وجود دارد را import می کنیم:



```
import { Swiper, Navigation, Pagination } from 'swiper/swiper.esm';
```

پارامترهای مورد نیاز را تعریف می کنیم، برای مثال به کمک پارامتر breakpoints مشخص می کنیم با توجه به عرض های مختلف صفحه چند اسلاید نمایش داده شود، پارامتر autoplay مشخص می کند که اسلایدر بصورت خودکار به اسلاید های بعدی برود.

```

const params = {
  // Provide Swiper class as props
  Swiper,
  // Add modules you need
  modules: [Navigation, Pagination],
  pagination: {
    el: '.swiper-pagination',
    type: 'bullets',
    clickable: true,
  },
  autoplay: {
    delay: 2500,
    disableOnInteraction: false
  },
  navigation: {
    nextEl: '.swiper-button-next.btn-slider',
    prevEl: '.swiper-button-prev.btn-slider'
  },
  renderPrevButton: () => <button className="swiper-button-prev btn-slider"></button>,
  renderNextButton: () => <button className="swiper-button-next btn-slider"></button>,
  spaceBetween: 30,
  breakpoints: {
    // when window width is >= 320px
    320: {
      slidesPerView: 1,
      spaceBetween: 20
    },
    577: {
      slidesPerView: 1,

    },
    // when window width is >= 480px
    768: {
      slidesPerView: 2,
      spaceBetween: 50
    },
    // when window width is >= 640px
    992: {
      slidesPerView: 2,
    },
    1200: {
      slidesPerView: 3,
    }
  },
};

```

در آخر لازم است این پارامترهای تعریف شده را به کامپوننت `ReactIdSwiperCustom` بدهیم:

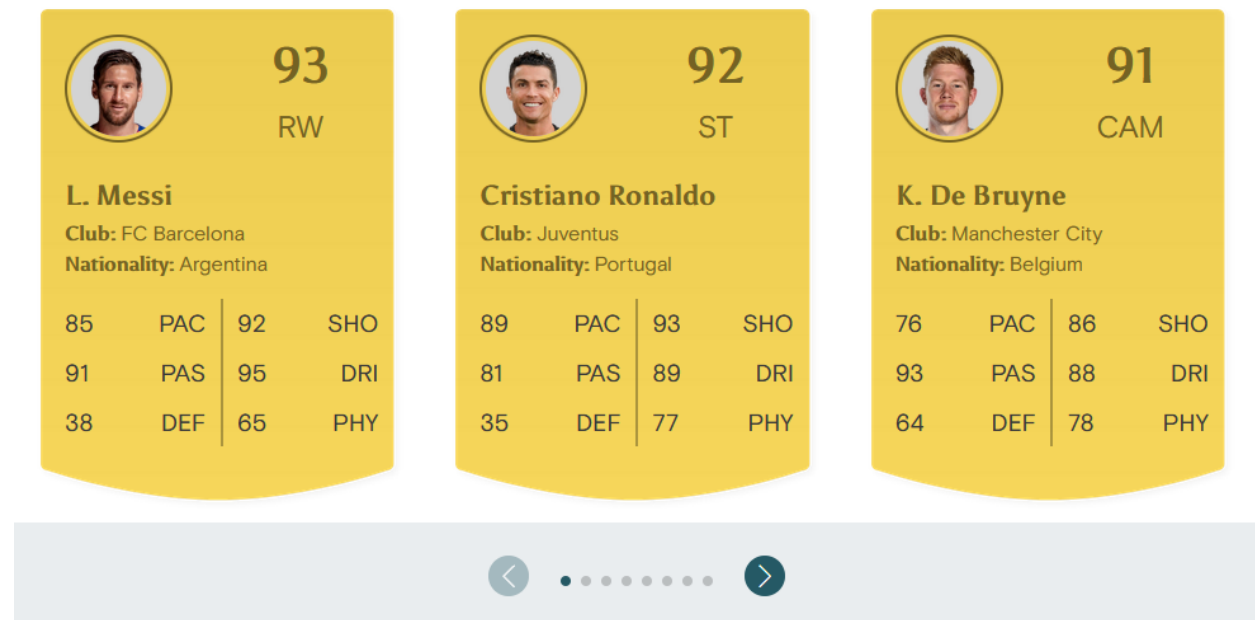
```

<ReactIdSwiperCustom {...params}>
  ...
</ReactIdSwiperCustom>

```

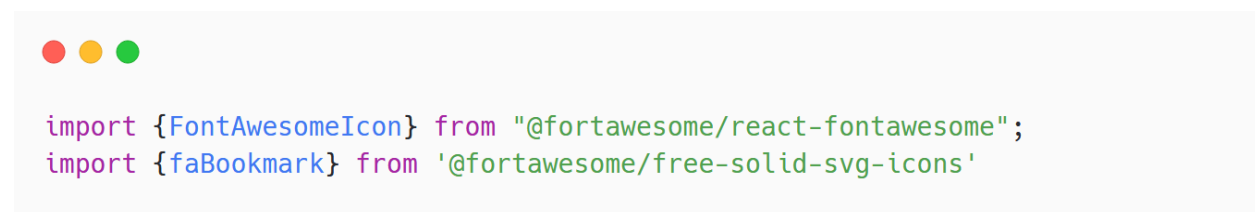
نمونه ای از اسلایدرهای پیاده سازی شده به کمک این کتابخانه در پروژه در تصویر زیر آمده است، این اسلایدر نشان دهنده 10 بازیکن برتر از نظر قدرت می باشد:

## Top Players



## Fortawesome

این کتابخانه شامل مجموعه ای از آیکون ها در دسته بندی های مخلف می باشد، برای استفاده از آیکون های موجود کامپوننت Fortawesome و کامپوننت مربوط به آیکون مورد نظر را در کامپوننت مقصد import می کنیم، برای مثال برای استفاده از آیکون :Bookmark

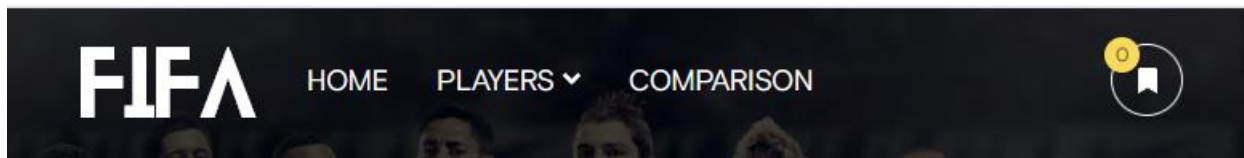


سپس در بدنه کامپوننت مقصد بصورت زیر از آن استفاده می کنیم:



```
<FontAwesomeIcon icon={faBookmark} />
```

از آیکون BookMark در منو برنامه استفاده شده است:



## استفاده از API ها

برای استفاده از اطلاعات دیتاست، API های مختلفی به زبان PHP نوشته شده است، برای ارسال درخواست به API ها در کامپوننت ها از fetch استفاده شده است، برای مثال برای درخواست اطلاعات 5 دروازه بان برتر بصورت زیر عمل شده است:



```
fetch("http://nabzsalamati.ir/FIFA/?Best=GK&num=5")
  .then(res => res.json())
  .then(
    (result) => {
      setPlayerGK(result);
    },
    (error) => {
    }
  );
```