**Tech Stack**

**Node.js & npm:**   
Ugye mennyire irritalo, hogy midenele JS libraryket kezzel kell copy-pastelni a weboldal HTML-jebe? Fear not, erre van az **npm** (Node Package Management). Ez manage-eli a kulonbozo JS package-eket. A build fazisban a megirt scripteket automatikusan bundle-olja (osszerakja egy bazi nagy JS fajlba, ami emberek szamar olvashatatlan), atalakitja vanilia JS-e es igy lesz publikalva.  
Az **npm** a Node.js resze, ami JavaScript-eket futtat lokalisan. Majd le kell toltenetek…

**TypeScript:** JavaScript (ES6) + tipusok.   
Itt neked kell definialni a tipusokat (ez rengeteg bugot kiszur).

**Mi a fo ujdonsag ES6-ban?**

- Valtozo definiciok:   
**let** – ugy mukodik, mintha a C-ben definialnad;   
**const** – konstans, azonban array elemet vagy object field-et valtoztathatsz rajta (i.e. csak a reference allando) => const preferalt, var meg semmikeppen sem!

- Arrow functions:  
const print = (n: number) => console.log(n);

- String interpolation:  
const myConst = ‘almafa’;  
console.log(`Az alma az ${myConst}n terem`);  
Fontos, hogy `-t hasznaltam (magyar billentyuzeten: alt gr + 7)

- Ternary statement:  
Olyan mint a Java-ban.  
*condition* ? *if\_true* : *if\_false*;  
const myConst = (today === ‘2018-09-18’) ? ‘It is today’ : ‘It is not today’;

<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>  
<https://medium.com/javascript-scene/javascript-es6-var-let-or-const-ba58b8dcde75>  
<https://codeburst.io/javascript-arrow-functions-for-beginners-926947fc0cdc>

**React:** HTML a JS kodban 😉  
Na ez egy kicsit hosszabb…  
<https://reactjs.org/docs/introducing-jsx.html>  
A main concepts kategoriaban egeszen a a Handling Events-ekig.  
Amit tudnod kell:  
- Mi az a JSX, miert jo?  
- Hogyan tudsz JS expression-t irni egy HTML reszletbe?  
- Mik azok a component-ek?  
- Props, state, setState, mikor melyiket hasznald.  
- Mikor renderel egy component?

**File extensions:**  
- .js: JavaScript;  
- .ts: TypeScript;  
- .jsx/.tsx: JS vagy TS, ami JSX-et (HTML-t) tartalmaz;

**Gatsby:** Ez varazslatos modon a markdown fajlokbol weboldalakat general.  
A grafikon a weboldalon nagyon jo: <https://www.gatsbyjs.org/>

Majd latni fogjatok a projektben van 3 fontos konyvtar:  
- **src:** Itt van a forraskod, ebben dolgozunk.  
- **static/assets:** Itt vannak a asset-ek (kep, zene, video, … foleg kepek)  
- **public:** Itt vannak a generalt oldalak. Ez az, amit publikussa teszunk

A gyokerben vannak a config fajlok:  
gatsby-\*.js -> gatsby   
package.json & package-lock.json -> node.js   
tsconfig.json & tslint -> TypeScript linting

**npm run develop:** Console command, ami build-eli a weboldalt es azonnal lathatod (localhost:8000). Raadasul amint megvaltoztatsz valamit a kodban egybol latszodik a valtozas.

**Most egy kicsit beszeljunk az src konyvtarrol:**

Ha megnezed 4 konyvtar van benne:  
**- components:** Ide mennek a a React component-ek  
**- layouts:** Ez a nagyjaboli layout, amit (majdnem) minden oldal kovet (e.g. metadata, footer, esetleg navbar).  
**- pages:** Itt talahatoak az egyes oldalak. Minden egyes fajlhoz lesz egy legeneralt oldal a konyvtar-szerkezettek megegyezo URL-en. Egy oldal lehet: \*.tsx (tehat egy teljes komponens) vagy \*.md (vagyis csak adat, ehhez kell egy tamplate, ami alapjan az oldal le less generalva).   
**- templates:** Az egyes tematikus oldalak, mint pl. ujsagok ez alapjan lesznek legeneralva az md fajlokbol.

**Fontos, hogy ezeknek a konyvtaraknak a szerkezetet megertsd! Sok kesobbi problemat megeloz, tapasztalatbol tudom…**

**Hogyan lesz egy .md fajlbol egy oldal?**  
- Gatsby eszreveszi, hogy a fajl letezik az **/src/pages** mappan belul.  
- Beolvassa az adatokat es eltarolja oket a GraphQL adatbazisaban.  
- Megnezi, hogy melyik *template* es *layout* alapjan kell oket legeneralni.  
- A *template* egy GraphQL query alapjan lekerdez tetszoleges informactiot a fajl-rol (**az osszes tobbi fajlrol is tud!**) es a kapott adatok alapjan a *template*-ben meghatarozott modon legeneralja az oldal belsejet, majd beilleszti a *layout*-ba.

Hasonloan mukodik a .tsx fajlbol valo generalas is, a template-en belul le tudod kerni a tobbi .md fajlban tarolt adatokat is.