**Sisukord:**

* Aruanded igast päevast
* Kokkuvõte
* Lisamärkused

**ARUANDED IGAST PÄEVAST:**

18.04.2022 (1 h)

Alustasin projektiga tegelemist. Ise olen varem tegelenud vaid Java koodiga, nii et antud proovitöö kõlab nagu päris paras ’challenge’, kuna mul pole eriti aimu kuidas miskit teha. Aga olen päris harjunud guugeldamisega (õppisin programmeerimist iseseisvalt, nii et Google on mu parim sõber) ja olen kindel, et suudan kasvõi midagigi ära teha.

Alustan esimesest etapist ja kui see enam-vähem tehtud saab, siis alles siis vaatan edasi, et mitte ülekoormata ennast.

Mõtlesin alustada sellega, mida enam-vähem oskan ja alustasin Spring Boot projekti IntelliJ-s. Homme püüan seal midagi teha.

22.04.2022 (3 h)

Mõtlesin alguses kasutada „Lisalugemist“ peatükis antud ’TimeAndDate’ API-d päikesetõusu ja päikeseloojangu aja saamiseks, kuid millegipärast ma ei suutnud seda normaalselt tööle panna. Guugeldades leidsin väga lihtsa API (https://sunrise-sunset.org/api), mis on tasuta ja mida on väga lihtne kasutada ning otsustasin kasutada seda. Siin aga tekkis väike probleem seetõttu, et see API ei anna aegu lokaalses ajas, vaid ajas GMT + 0 ning mul pole aega sellega midagi teha, sest on niigi palju veel teha. Nii et lihtsalt kirjutan pärast juurde, et ajad pole lokaalses ajas.

Päris kiiresti tegin ära back-end’i, nii et klient saadab koordinaadid ja kuupäeva ning saab tagasi päikesetõusu ja päikeseloojandu ajad (öö pikkust veel ei arvutanud). Back-end’i tööd kontrollisin Postman’i programmiga, kuna veebilehte mul veel ei ole.

25.04.2022 (3 h)

Kuidagi väga palju aega läks täna öö pikkuse arvutamise meetodi tegemiseks. See ei olnud raske, pigem oli väga palju pusimist. Püüdsin alguses aega formaadis AM/PM muuta 24h formaadiks, kulutasin selleks juba pea tund aega, aga siis nägin et API veebilehel võib saata request ’formatted’, nii et saabki kätte kohe 24h formaadi. Kahju, et ma seda kohe tähele ei pannud, aga väga tore, et lõpuks panin.

Nüüd tuli kuidagi teha arvutusi kellaaegadega, mis oli raskem kui arvasin. Mõtlesin, et saan poole tunniga hakkama, aga läks pea kaks tundi. Tuli mõnda aega guugeldada ja lõpuks leidsin kuidas kellaaegu formateerida, nendega arvutada ja siis millisekunditest tunnid, minutid ja sekundid tagasi saada.

See ei olnud raske, aga pikemat aega võttis küll.

26.04.2022 (4h)

Nüüd tuleb teha veebileht. See on see raske osa, mille kohta ma ei tea pea midagi. Ma olen väga ammu paar nädalat tegelenud HTML-i ja CSS-iga, kuid JavaScripti ma pole kunagi kirjutanud ja ei tea kuidas see töötab. Aga see on programmeerimiskeel, nii et kasvõi natuke see peab Java-ga sarnane olema.

Ma tean, et JavaScripti abiga ma pean kuidagi saatma requeste brauserist oma serverile, nii et alustan sellega, et teen mingisuguse GET requesti, sest tean, et see peaks lihtsam olema. Paralleelselt kontrollin Postman’i kaudu, kas server töötab normaalselt või mitte.

Guugeldan non-stop. Püüan, et veebileht kasvõi kuidagi serveriga räägiks, aga miskit talle ei meeldi ja ma üldse ei tea mis. Ma ei oska eraldi HTML-i või JavaScript’i dokumente teha, nii et kirjutan kõike NotePad-is, sest see tundub praegu kõige lihtsam variant olevat. Guugeldades leidsin, et veebilehel olles võib vajutada F12 ja siis sealt saab igasugu tarka infot, väga kasulik.

Lõpuks! Proovisin palju eri asju ja see nüüd millegipärast hakkas tööle! Ju vist oli Spring Boot’i @CrossOrigin tõesti tähtis. Ei tea, mida ma täpselt tegin, aga kui palju proovida ja erroreid vaadata, siis tavaliselt ikka miskit hakkab tööle.

See oli praegu kõige raskem, aga ka kõige rõõmsam päev: mu üli-lihtne veebileht ja server räägivad üks teisega.

27.04.2022 (5h)

Alles nüüd panin tähele, et unustasin Git’i kohta ja ei teinud commit’e viimased paar päeva. Tegin Git’i kursuse läbi vaid kuu aega tagasi ja ma veel ei ole harjunud seda kogu aeg kasutama.

Täna peaks siis tegema mingi vormi HTML’is, kuhu saab koordinaate ja kuupäeva sisestada ja siis GET requesti asemel tuleb saata POST request koos JSON-iga. Kuidagi mulle tundub, et POST request tööle saada on palju raskem GET request-ist.

Vormi tegin HTML-is kiiresti ära, olen seda varem teinud ja see on väga lihtne. Nüüd tuleb see vorm kuidagi JSON’ks muuta ja serverile saata.

See ei olnudki nii raske, guugeldades leidsin enam-vähem täpselt seda, mida vaja ja see nägi päris sarnane GET requestiga. Selleks läks alla tund aega. Olen väga rõõmus, sest kartsin et sellega tuleb pusimist.

Kuna mul ’Etapp 1’ on valmis, siis võin proovida teha ’Etapp 2’, mul on veel kaks päeva olemas.

Kasutasin Leaflet API-d, mis oli „Lisalugemist“ peatükis antud. Kaardi lisamine veebilehele oli hämmastavalt lihtne, poole tunniga sain valmis. Alguses mu kaart oli tervenisti hall, aga siis lugesin täpsemini Quick Start Guide’i ja tekitasin endale acces token’i ning kõik hakkas tööle.

Kiiresti tegin nii, et kaardil klikates koordinaadid täitusid automaatselt. Siin aitas väga, et tean kuidas Java’s Stringe manipuleerida ning õnneks on see osa JavaScriptis väga sarnane.

Korrastasin natuke oma back-end’i, nii et seda on lihtsam ja mõnusam lugeda.

Olen väga õnnelik ja üllatunud, et jõudsin täna nii palju ära teha.

28.04.2022 (5 h)

Tahan täna juurde panna nuppu, nii et käsitsi sisestades koordinaate võib näha kus see koht kaardil asub.

See kaardi API on nii lihtne ja arusaadav, et sain sellega poole tunniga hakkama, tuli vaid natuke dokumentatsiooni lugeda ja näiteid vaadata markerite kohta.

Kõik mu programmi funktsioonid on nüüd valmis, ma rohkem teha ei jõua. Nüüd tuleb front-end kood ilusamaks teha ja ka veebileht ise ilusamaks muuta CSS abil. CSS’i olen natuke teinud, nii et see peab olema lihtne.

Panin CSS juurde, see oli lihtne. Nüüd kõik näeb natuke kaunim välja.

Leidsin probleemi, mida varem ei pannud tähele – vajutades nuppu „kuva koht kaardil“ saadab veebileht requesti serverile, kuigi peaks vaid lisama markeri kaardil. Ma ei tea miks, sest ma ei sidunud seda nuppu selle funktsiooniga.

Pea tund aega guugeldades leidsin vastuse – nupule tuleb juurde panna ’type=“button“’. Ei jõua vaadata, miks see nii töötab, aga olen väga õnnelik, et töötab.

Nüüd idee oli korrastada front-end’i faili kood, kuid kuna ma pole eriti HTML’i, CSS’i või JavaScripti varem kirjutanud, siis ma ei tea kuidas seal tavaliselt koodi vormistatakse. Tean, et HTML, CSS ja JS võib eraldada eraldi failidesse, kuid see mul ei tulnud praegu välja ja aega jääb napiks, nii et lihtsalt panen juurde kommentaarid ja loodan, et see ei näe väga kohutav välja. Igaks juhuks vabandan selle inimese ees, kes hakkab seda lugema. Faili lõppu lisasin ka lingid, kust ma leidsin erinevad koodijupid.

**KOKKUVÕTE:**

Selle proovitöö peale läks mul umbes 21 tundi. Ebaproportsionaalselt palju aega läks front-end’i peale, sest ma pole sellega pea kunagi kokku puutunud. Olen väga rõõmus, et mul tuli kõik enam-vähem välja.

**LISAMÄRKUSED:**

* Ma kasutasin oma projektis päikesetõusu (sunrise) ja loojangut (sunset), mitte koidikut (dawn) ja loojangut (sunset), sellepärast et päikesetõus ja loojang on mõlemad seotud konkreetselt päikese tõusmisega horisondist või kadumisega horisonti ehk need on omavahelt tihedalt seotud ja neid saab võrrelda. Koidik ja loojang aga ei ole nii täpselt omavahel seotud, sest koidik on seotud päikesevalguse ilmumisega taeval, mitte konkreetselt päikese endaga, nii et koidik ja loojang ei ole n-ö paar ja minu arvates tundub päikesetõusu ja loojangu kasutamine õigem.

Ma tean, et proovitöö dokumendis oli räägitud just koidikust ja loojangust, kuid samuti oli seal öeldud „Kui ülesande püstituses on sinu jaoks mitmeti mõistetavaid kohti, siis märgi ära eeldused, mida oled ülesande lahendamisel teinud.“ ning minu jaoks oligi see selline koht, mis oli natuke veider ja mitmeti mõistetav.

* Öise perioodi tähendust võtsin Wikipeediast.

*Wikipeedia:* ***Öö****on ajavahemik, mille jooksul taevakeha (planeet või tema kaaslane) pinna punkti jaoks on valguse peamine allikas (päike) horisondist allpool.*

Sellest tulenevalt on minu programmis öö päikeseloojangust päikesetõusuni ehk siis too aeg, mil päikest pole näha (aga valge võib ikka olla).

* Öö pikkus on arvutatud kasutades päikesetõusu aega ja päikeseloojangu aega, ehk ma eeldan et API poolt antud päikesetõusu aeg on aeg, mil päike hakkab alles välja tulema (mitte aeg, mil päike on täpselt täiesti välja tulnud) ning et päikeseloojang on aeg, mil päike on just täpselt horisondi alla läinud (mitte aeg, mil päike vaid alustab horisondi alla minema). Ehk ma eeldan, et need on täpselt kaks piiri-aega, millal päikest ei ole taevas.
* Öö pikkuse arvutamisel kasutan kliendi poole antud päeva päikeseloojangu ja päikesetõusu aega. Tean, et see pole päris korrektne, kuna konkreetse päeva öö algab eelmise päeva päikeseloojangust, kuid otsustasin et ei taha programmi keerulisemaks teha, sest et öö pikkus tuleb ikka enam-vähem õige, veaga +- 5 min.