

Lauri Kemppainen

Avoimen taloustiedon hyödyntäminen:

Julkisen datan rajapinnat ja integraatiot

Insinööri, (AMK)

Tieto- ja viestintäteknikka

Kevät 2025



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä(t): Lauri Kemppainen

Työn nimi: Avoimen taloustiedon hyödyntäminen: Julkisen datan rajapinnat ja integraatiot

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: Ohjelmistokehitys, Ohjelmointirajapinta, Avoin data

Opinnäytetyö toteutettiin toimeksiantona ohjelmistoprojektien alihankintaan erikoistuneelle KajaPro Oy:lle. Työn tavoitteena on tutkia, mitä talouteen liittyvää julkista dataa on saatavilla ja kuinka siihen liitytään ohjelmallisesti.

Työssä kartoitettiin erilaisia avoimen datan rajapintoja, niiden tarjoamia mahdollisuuksia sekä teknisiä toteutustapoja. Työssä käytiin läpi julkisen datan saatavuutta, tietolähteitä ja niiden käyttöönottoon liittyviä haasteita. Lisäksi tarkasteltiin käytännön esimerkkejä ja toteutettiin talousdataan liittyvä demosovellus erilaisten rajapintojen avulla. Teknologioiksi sovellukselle valittiin Python-ohjelmointikieli ja sen sisältämät datan käsittelyyn tarkoitetut kirjastot Numpy, Pandas ja Matplotlib.

Demosovelluksen toteutuksessa hyödynnettiin useita eri rajapintoja, joiden avulla haettiin ja esitettiin talousdataa havainnollisessa muodossa. Talouteen liittyvää dataa haettiin pääasiassa Suomen tilastokeskuksesta, Eurostat-tilastopalvelusta ja Yahoo Financesta. Työn aikana havaittiin, että vaikka julkista talousdataa on runsaasti saatavilla, sen löydettävyyden ja yhtenäisyyden vaihtelevat merkittävästi lähteittäin. Erityisiä haasteita aiheuttivat muun muassa eri rajapintojen dokumentaation puutteellisuus sekä tietorakenteiden erilaisuus.

Työstä muodostui katsaus julkisen taloustiedon hyödyntämiseen, ja toteutuksena syntyi toimiva demo taloustiedon hakemiseen. Sovelluksesta saadusta datasta voidaan tarkastella suurimmaksi osaksi Suomen talouden tilaa korkealla tasolla. Tärkeimmäksi jatkokehityskohteeksi jäi sovelluksen tai sovelluksesta saadun datan integroiminen muihin järjestelmiin helpolla tavalla. Opinnäytetyö tarjoaa perustan julkisen talousdatan ohjelmointirajapintojen hyödyntämiselle sekä datan käsittelyyn ja esittämiseen.

Abstract

Author(s): Lauri Kemppainen

Title of the Publication: Utilization of open financial information: Interfaces and integrations of public data

Degree Title: Bachelor of Engineering, Information Technology

Keywords: Software development, Application programming interface, Open data

This thesis was carried out as a commissioned project for KajaPro Oy. The objective of the thesis is to examine what publicly available economic data exists and how it can be accessed programmatically.

Various open data APIs, their capabilities, and technical implementation methods are explored. The availability of public data, sources of information, and the challenges related to them are reviewed. In addition, practical examples are examined and a demo application related to economic data is implemented using different APIs. The technologies chosen for the application include the Python programming language and its data processing libraries: Numpy, Pandas, and Matplotlib.

In the implementation of the demo application, several APIs were utilized to fetch and present economic data. The economic data was primarily sourced from Statistics Finland, the Eurostat statistical service, and Yahoo Finance. It was observed during the process that although a vast amount of public economic data is available, its discoverability and consistency vary significantly depending on the source. Specific challenges were caused by incomplete API documentation as well as differences in data structures.

The outcome of the thesis is a review of how public economic data can be utilized, and a functional demo for retrieving economic data was successfully developed. The data retrieved through the application mainly allows for a high-level overview of Finland's economic situation. The main area identified for future development is making it easier to integrate the application or its data into other systems. This thesis provides a foundation for utilizing APIs, handling data, and presenting data effectively.

Sisällys

1	Johdanto	1
2	Julkinen data.....	2
2.1	Avoin data	2
2.2	Talouteen liittyvä data	3
2.2.1	Tilastokeskus.....	4
2.2.2	Yahoo Finance.....	5
2.2.3	Eurostat.....	6
2.2.4	Datalähteiden vertailu	7
2.3	ETL-prosessi.....	8
3	Liitynnät dataan.....	10
3.1	Tiedostona ladattavat tietojoukot ja niiden formaatit.....	10
3.2	Ohjelmointirajapinta	12
3.3	Web scraping.....	13
4	Esimerkkisovellus julkisen taloustiedon hakemiseen.....	15
4.1	Tavoitteet ja suunnittelu	15
4.2	Toteutus	18
5	Yhteenveto	22
	Lähteet	24

Liitteet

Symboliluettelo

API	Application programming interface. Ohjelmointirajapinta.
REST	Representational state transfer. Yleisin ohjelmistorajapintojen arkkitehtuurimalli.
CRUD	Create Read Update Delete. Standardisoidut perustason tietokantaoperaatiot.
HTTP	Hypertext transfer protocol. Protokolla resurssien jakamiseen WWW-järjestelmässä.
HTML	Hypertext markup language. Verkkosivujen merkintäkieli.
dataframe	Taulukkomainen tietorakenne Python-ohjelmointikielessä.
ETL-prosessi	Extract, transform, load. Tiedon varastoinnissa käytetty tekniikka.
SQL	Structured Query Language. Relaatietietokantojen hallintaan tarkoitettu kyselykieli.

1 Johdanto

Nykymaailmassa datavetoiset päätöksentekoprosessit ovat nousseet keskeiseksi osaksi julkishallinnon, yritysten ja tutkimuslaitosten toimintaa. Erityisesti talouteen liittyvää dataa on saatavilla yhä enemmän eri julkishallinnon ja kansainvälisten organisaatioiden tuottamana. Tällaisen datan hyödyntäminen on tärkeää mm. taloudellisten päätösten tueksi, sillä se mahdollistaa ajankohtaisen ja tarkasti analysoidun tiedon pohjalta tehtävät strategiset valinnat.

Avoimen datan hyödyntäminen ohjelmallisesti tarjoaa tehokkaita keinoja taloudellisten kehitysuuntien analysointiin ja visualisointiin. Useat datalähteet tarjoavat ohjelmointirajapintoja (API), joiden avulla tietoa voidaan kerätä ja käsitellä automaattisesti, esimerkiksi Python-ohjelmointikielillä.

Työn toimeksiantaja oli KajaPro Oy, vuonna 2004 perustettu ohjelmistoprojektien alihankkija. Toteutuksen tavoitteena oli kehittää prototyyppisovellus julkisesti saatavilla olevan taloustiedon haakuun.

Työn teoriaosuudessa määritellään julkinen ja avoin data sekä tarkastellaan työssä tutkittavaa taloudellista dataa. Tämän jälkeen käydään läpi ja vertaillaan yleisiä datalähteitä ja niiden tarjoamia liityntöjä. Lopuksi kartoitetaan toteutusosiossa käytettäviä tekniikoita ja teknologioita. Toteutusosiossa käydään läpi tavoitteita, suunnittelua ja valittuja teknologioita, minkä jälkeen esitetään sovelluksen tärkeimpiä osia ja toimintaa. Toteutuksen kuvaamisen jälkeen pohditaan sovelluksen parannuskohteita ja jatkokehityssuuntia, jotka mahdollistaisivat sen kehittämisen helposti integroitavaksi analytiikkapalveluksi.

2 Julkinen data

Julkisella datalla tarkoitetaan kaikkea tietoa, jonka organisaatio on antanut kaikille saatavaksi. Data on yleensä peräisin julkiselta sektorilta, mutta myös yksityiset organisaatiot voivat julkaista sitä säännöksiä tai läpinäkyvyyden parantamisen takia. Sen saatavuus voi kuitenkin vaihdella ja siihen pääsyä voidaan rajoittaa esimerkiksi pyynnöllä tai kirjautumisella. Julkisen datan käsite on laaja, mutta yleisesti siitä puhuttaessa viitataan valtioiden julkaisemiin tilastoihin tai asiakirjoihin. [1.]

Datan käytettävyys vaihtelee, eikä se ole aina helposti analysoitavassa muodossa, kuten skanna-
tuissa PDF-dokumenteissa, joten hyvin ylläpidetyn tietolähteen etsiminen voi olla haasteellista. Kaupalliseen käyttöön, uudelleenjulkaisuun tai muokkaamiseen voi myös liittyä rajoitteita. [1.]

Julkisen datan merkitys kasvaa jatkuvasti, kun yhteiskunnat digitalisoituvat ja tietoon perustuva päätöksenteko korostuu. Hyvin hallinnoitu julkinen data voi edistää innovaatiota, tieteellistä tutkimusta ja kansalaisyhteiskunnan osallistumista. Se voi auttaa kehittämään uusia sovelluksia, jotka parantavat kansalaisten elämänlaatua tai tehostavat yritysten toimintaa. Samalla se tukee demokratiaa ja läpinäkyvyyttä, kun kansalaiset ja tutkijat pääsevät käsiksi päätöksenteon taustalla olevaan tietoon. Julkisen datan tehokas hyödyntäminen edellyttää kuitenkin, että data on saatavilla käyttökelpoisessa ja analysoitavassa muodossa sekä selkeästi lisensoitu sen käyttötarkoitusten mukaisesti. [1.]

Yritykset voivat hyödyntää julkista dataa kysynnän arviointiin tai liiketoimintariskien tunnistamiseen ja hallitsemiseen. Esimerkiksi julkisesti saatavilla olevat tiedot taloudellisista indikaattoreista, toimialojen suhdannevaihteluista ja sääntelymuutoksista voivat auttaa yrityksiä ennakoimaan mahdollisia riskejä ja reagoimaan niihin ajoissa. Toiminnan tehostamiseksi sekä päätöksenteon parantamiseksi yrityksiä kannattaa yhdistää julkisista lähteistä saatavaa dataa niiden omaan toiminnanohjausjärjestelmästänsä tai asiakkuudenhallintajärjestelmästänsä tulevaan dataan. [1.]

2.1 Avoin data

Avoin data eroaa julkisesta datasta saavutettavuudessa ja uudelleenkäytön mahdollisuuksissa. Avoin data tarkoittaa sitä, että kaikki voivat käyttää sitä tasavertaisesti ja kaikilla on maksuton

pääsy siihen. Avoimeksi määritellyn datan tulee myös olla koneluettavassa muodossa ja saavutettavissa esimerkiksi CSV-, XLS- tai XML-tiedostomuodoissa taikka ohjelmointirajapintojen kautta. Edellä listattujen kriteerien lisäksi tiedon rakennetta ja merkitystä pitää kuvata sen käyttäjälle selvästi. Tiedon ymmärrettävyyttä käyttäjälle helpotetaan metadatan avulla. [2.]

Avointa dataa voidaan hyödyntää julkisen datan lailla liiketoiminnan parantamiseen. Sen käytöstä syntyvät kustannukset ovat pienet vapaan käytön lisenssin sekä koneluettavuuden takia, joten oikein käytettynä sillä voidaan luoda yritykselle lisäarvoa kustannustehokkaasti. Vain pieni osa julkisesta tiedosta voidaan luokitella avoimeksi dataksi (Kuva 1).



Kuva 1. Vain pieni osa julkisesta tiedosta voidaan luokitella avoimeksi dataksi.

Avoimen datan juuret juontavat 1980- ja 1990-luvun vapaita ohjelmistoja ja avointa lähdekoodia edistäviin liikkeisiin. Nykyään avoimuus on eräänlainen megatrendi, joka vahvistuu useilla eri aloilla. Suomessa datan avaaminen alkoi kiihtyä vuonna 2009, osittain Euroopan Unionin PSI-direktiivin (julkisen sektorin tiedon uudelleenkäytön helpottaminen 2003 ja uudistus 2013) ansiosta sekä erityisesti Britannian ja Yhdysvaltojen esimerkkien innoittamana. [3.]

2.2 Talouteen liittyvä data

Talouden kehitystä kuvaavat keskeiset indikaattorit voidaan jakaa johtaviin ja jäljessä seuraaviin mittareihin. Johtavat indikaattorit, kuten osakemarkkinat, kuluttajien luottamusindeksit ja uusien

rakennuslupien määrä, voivat antaa viitteitä tulevasta talouskehityksestä. Ne heijastavat talouden odotuksia ja voivat ennakoida kasvua tai taantumaa, sillä esimerkiksi osakemarkkinat ennakoivat usein yritysten tulevia tulosodotuksia. Näitä mittareita käytetään erityisesti lyhyen aikavälin talousennusteissa. [4.]

Lyhyen aikavälin bruttokansantuotteen ennusteet perustuvat yleensä menoerien, kuten kulutuksen, julkisten menojen, investointien ja ulkomaankaupan, kehitysoletuksiin. Pitkän aikavälin ennusteet taas perustuvat tarjontatekijöihin, kuten pääomaan, työvoimaan ja kokonaistuottavuuteen. Pääoman kasvu kuvastaa rakennusten, koneiden ja ohjelmistojen tarjonnan kasvua, työvoiman kasvu työntekijöiden määrän ja taitotason parantumista, millä pääomaa ja työvoimaa hyödynnetään. Työvoiman määrää ja laatua ennustetaan erikseen esimerkiksi väestönkasvun ja koulutustason perusteella. Lopulta nämä tekijät yhdistetään BKT:n potentiaalisen kasvun ennustamiseksi huomioiden lyhyen aikavälin suhdannevaihtelut erikseen. [5.]

Jäljessä seuraavat indikaattorit, kuten bruttokansantuote (BKT), työttömyysaste ja inflaatio, kertovat talouden tilasta jälkikäteen. Ne mittaavat talouden jo tapahtuneita muutoksia ja auttavat ymmärtämään, millaisessa vaiheessa taloussuhdanne on. BKT kuvaa kokonaistuotannon kehitystä ja on keskeinen talouskasvun mittari, kun taas työttömyysaste heijastaa työmarkkinoiden tilannetta. Inflaatio puolestaan antaa tietoa hintatason kehityksestä ja ostovoimasta. Kokonaiskuvaa muodostettaessa talousanalyttikot yhdistelevät sekä johtavia että jäljessä seuraavia indikaattoreita saadakseen paremman käsityksen talouden suunnasta ja nykytilanteesta. [4.]

Seuraavissa luvuissa esitellään tämän työn näkökulmasta keskeisimpiä avoimen datan lähteitä ja lopuksi vertaillaan niitä keskenään.

2.2.1 Tilastokeskus

Tilastokeskus on itsenäisesti toimiva tilastotietoja tuottava valtion virasto. Sen tuottamat tiedot kattavat muun muassa väestönkehityksen, talouden tilan ja alueellisen kehityksen. Tilastokeskuksen tehtävänä on tarjota luotettavaa ja ajankohtaista tietoa, joka tukee niin tutkimusta, päätöksentekoa kuin yleisyyä tiedontarvetta. [6.]

Tilastokeskus julkaisee lukuisia tilastoja, jotka liittyvät esimerkiksi väestön rakenteeseen ja muutoksiin, työmarkkinoihin ja työllisyystilanteeseen, kansantalouden kehitykseen, yritystoimintaan

ja innovaatioihin ja paikkatietoihin [7]. Tilastokeskus tarjoaa myös pitkän aikavälin tilastoja, jotka ovat hyödyllisiä historiallisessa tutkimuksessa ja trendianalyyseissä. Tilastot noudattavat myös kansainvälisiä standardeja, kuten SDMX (Statistical Data and Metadata Exchange).

Tilastokeskuksen tarjoamista tilastoista suuri osa on vapaasti käytettävissä avoimen tietoaistien käyttöluvan – CC BY 4.0 -lisenssin alla, mutta osa tietokannoista on maksullisia. Tilastokeskus tarjoaa maksuttoman PxWeb-rajapinnan, jolla voidaan hakea tilastoja StatFin-tietokannasta HTTP-pyyntöjä käyttämällä. [8.]

2.2.2 Yahoo finance

Yahoo Finance on verkkosivusto ja mobiilisovellus, joka tarjoaa taloudellisia uutisia, tietoja, kommentteja ja työkaluja. Se sisältää osakekurseja, taloudellisia uutisia ja analyysejä sekä rahastosalkkupalveluita. [9.]

Yfinance on avoimen lähdekoodin Python-kirjasto, jolla voidaan hakea tietoa Yahoo Finance -palvelun kautta osakkeista. Kirjasto käyttää web scraping -tekniikoita tiedon hakuun ja palauttaa sen helposti käytettävässä dataframe-tietomuodossa. Alustamalla kirjaston sisältämä Ticker-luokan olio voidaan päästä käsiksi osakedataan (Kuva 2). Olion luonnissa annetaan halutun osakkeen lyhenne argumenttina ja sen sisältämät metodit, kuten `history`, `dividends` tai `income_stmt`, palauttavat tietoa osakkeesta. [10.]

```

import yfinance as yf

msft = yf.Ticker("MSFT")

# get all stock info
msft.info

# get historical market data
hist = msft.history(period="1mo")

# show meta information about the history (requires history() to be called first)
msft.history_metadata

# show actions (dividends, splits, capital gains)
msft.actions
msft.dividends
msft.splits
msft.capital_gains # only for mutual funds & etfs

```

Kuva 2. Yfinance-kirjaston peruskäyttö ja ticker-olion luonti Microsoft-yrityksen osakkeelle.

2.2.3 Eurostat

Eurostat on Euroopan unionin tilastovirasto, joka tuottaa ja jakaa tilastoja EU-jäsenmaiden taloudellisesta, väestöllisestä ja sosiaalisesta tilanteesta. Se on osa Euroopan komissiota ja sijaitsee Luxemburgissa. Eurostatin tehtävänä on tarjota vertailukelpoisia ja laadukkaita tilastotietoja EU:n toimielimille ja kansalle. [11.]

Eurostatin kaikki tiedot ovat ilmaisia ja saatavilla sen verkkosivuilla. Se myös tuottaa raportteja, artikkeleita ja julkaisuja, joissa analysoidaan tilastoja ja niistä saatuja johtopäätöksiä. Vertailukelpoisuus on varmistettu laadunvarmistuskehyksellä, johon kuuluu esimerkiksi tarkkojen ohjeiden antaminen jäsenmaille siitä, kuinka tietoa kerätään ja raportoidaan. Se perustuu myös kansainvälisiin standardeihin, kuten Euroopan tilinpidon järjestelmään (European System of Accounts, ESA). [12.]

Eurostat tarjoaa REST-tyyppisen rajapinnan, jonka kautta voidaan hakea ja käyttää tilastoja suoraan ohjelmistoista ja sovelluksista [13].

2.2.4 Datalähteiden vertailu

Tilastokeskus, Yahoo Finance ja Eurostat ovat kaikki tärkeitä talousdatan lähteitä, mutta niiden painopisteet ja käyttötarkoitukset eroavat. Tilastokeskus keskittyy Suomen kansalliseen talous- ja väestödataan. Yahoo Finance on puolestaan globaali kaupallinen palvelu, joka tarjoaa reaaliaikaista markkinadataa, osaketietoja ja talousuutisia sijoittajille ja kuluttajille. Eurostat toimii EU:n tilastovirastona, ja sen vahvuus on jäsenvaltioiden välisessä vertailutiedossa. Näin ollen Tilastokeskus on paras paikallisiin analyysiin, Yahoo Finance sijoitus- ja markkinaseurantaan ja Eurostat EU-tason makrotaloudellisiin vertailuihin.

Julkisen datan lähteitä verrattuna. Taulukko 1.

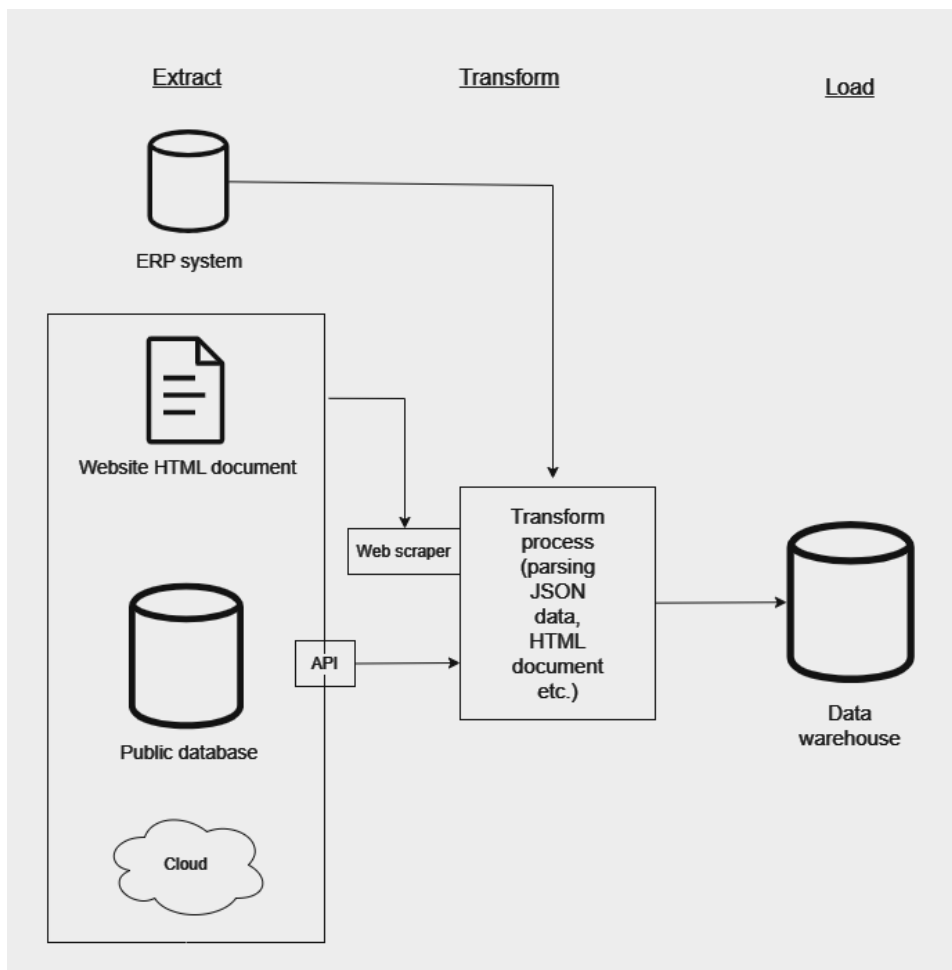
Palvelu	Tilastokeskus	Yahoo Finance	Eurostat
Käyttöliittymä	Tilastot saatavilla verkossa ja rajapinnan kautta	Verkkosivusto ja mobiilisovellus	Tilastot saatavilla verkossa ja rajapinnan kautta
Data	Tilastot taloudesta, väestöstä, ympäristöstä jne.	Taloudelliset tiedot, kuten osakekurssit ja analyysit	Tilastot EU taloudesta, väestöstä, työmarkkinoista
Maksullisuus	Osa tiedoista maksullisia, suuri osa ilmaisia	Ilmainen, mutta osalle datasta ja työkaluista maksullinen versio	Ilmainen, kaikki tiedot saatavilla
Rajapinnat	PxWeb-rajapinta	Virallisen rajapinnan tuki lopetettu. Yfinance-kirjastolla koneellisesti haettavissa.	REST API
Dataformaatti	CSV, JSON	HTML-dokumentti (haravoitavissa verkkosivulta)	CSV, JSON, XML

2.3 ETL-prosessi

ETL-prosessi on tiedon varastoisessa käytetty tekniikka, jolla tarkoitetaan tiedon hakemista, sen muuntamista ja lataamista kohdejärjestelmään. Prosessissa haetaan tietoa verkon julkiselta puolelta ja yrityksen sisältä, kuten toiminnanohjausjärjestelmästä. Tiedon muuntamisessa varmistetaan tietojen yhtenäisyys, ajankohtaisuus sekä virheettömyys ja muutetaan tieto kohdejärjestelmän vaatimaan muotoon. Kohdejärjestelmä yleensä sisältää yhden tai monia tietokantoja [14]. Tässä opinnäytetyössä voidaan puhua yksinkertaisen ETL-prosessin toteuttamisesta.

ETL-prosessin tavoitteena on varmistaa, että organisaatiolla on käytössään luotettavat ja yhtenäiset tiedot päätöksenteon ja analytiikan tarpeisiin. Tyypillisesti ETL-prosessia käytetään osana suurempaa tietovarastointiratkaisua. [14.]

Haettua tietoa voidaan suodattaa tietokantatauluihin sopivaksi yleisesti käytetyn objekti-relaatiokartoitustekniikan avulla. Objekti-relaatiokartoituksella tarkoitetaan yhteyden luontia olio-ohjelmointikielen olion ja tietokannan entiteetin välillä, mikä sujuvoittaa tiedonsiirtoa [15]. Muita ETL-prosessissa käytettäviä muuntamistekniikoita ovat mm. ryhmittely, aggregointi, suodattaminen ja liittyminen [16]. Ryhmittelyllä viitataan tietojen ryhmittelyyn tiettyjen arvoalueiden perusteella. Aggregoinnilla tarkoitetaan tietojen tiivistämistä tilastollisten suureiden kuten keskiarvon tai summan perusteella. Suodattamisella tarkoitetaan vain tiettyjen rivien valitsemista tai tyhjien poistamista. Liittymisellä tarkoitetaan monen taulukon yhdistämistä yhteisten sarakkeiden perusteella. Nämä tekniikat voidaan melkein suoraan yhdistää SQL-kielen toimintoihin, joten ne voidaan toteuttaa sillä tehokkaasti. ETL-prosessin päävaiheet on esitetty kuvassa 3 (kuva 3).



Kuva 3. Esimerkki ETL-prosessista.

Kohdejärjestelmä on usein liiketoimintatiedon hyödyntämiseen tarkoitettu palvelu. Yksi suosituimmista työkaluista tähän on Power BI. Power BI on Microsoftin kehittämä pilvipohjainen liiketoimintatiedon hallintatyökalu, joka mahdollistaa tietojen keräämisen, analysoinnin ja interaktiivisten raporttien sekä koontinäyttöjen luomisen. [17.]

3 Liitynnät dataan

Julkiseen tai avoimeen dataan koneellisesti liityttäessä voidaan valita monia tapoja tiedon saavutettavuuden tai tietojoukon luonteen perusteella. Yksi yleisimmistä tavoista on hyödyntää avoimia ohjelmointirajapintoja (API), jotka tarjoavat ohjelmallisen pääsyn dataan ja mahdollistavat tietojen hakemisen ja päivittämisen reaaliajassa [18]. Vaihtoehtoinen tapa on ladata tiedostomuotoisia tietojoukkoja, kuten CSV-, excel- tai JSON-tiedostoja, jotka tarjoavat helpon tavan tallentaa suuria määriä tietoa.

Lisäksi voidaan käyttää web scraping -tekniikoita, mikäli data ei ole suoraan saatavilla rajapinnan kautta, mutta se on julkaistu verkkosivustoilla. Tämä menetelmä vaatii kuitenkin erityistä huomiota käyttöehtoihin. Jotkin tietojoukot voivat myös olla saatavilla tiedonhakupöytäkirjojen kautta, jotka mahdollistavat tietojen etsimisen ja noutamisen erilaisista lähteistä. Joissakin tapauksissa data voi olla saatavilla myös julkisten tietokantojen kautta, joita voidaan käyttää suurien kyselyjen tekemiseen esimerkiksi SQL-kielillä.

Kaiken kaikkiaan valitun tavan tulisi perustua tietojoukon luonteeseen, tietojen käytön vaatimuksiin sekä käytettävissä oleviin teknisiin resursseihin.

3.1 Tiedostona ladattavat tietojoukot ja niiden formaatit

Tilastoja tuottavat organisaatiot voivat julkaista dataa suoraan pakattuina tiedostoina. Tämän tavan etuna voidaan pitää helppokäyttöisyyttä, sillä tiedon hakemiseen ei tarvita ohjelmointia, ja tieto on usein valmiiksi käsiteltyä. Tiedostona ladattavien tietojen käyttöä rajoittaa menetelmän staattisuus, sillä koko tiedosto joudutaan lataamaan uudelleen, jos tietoa halutaan päivittää. Tiedostosta ei voida myöskään hakea vain tiettyjä rivejä, kuten aikaväliä, sekä tiedostot voivat olla haasteellisia käsitellä koneellisesti, jos ne sisältävät monia taulukoita.

JSON (JavaScript Object Notation) on kevyt tiedonvaihtoformaatti. Se on koneille yksinkertainen jäsentää ja tuottaa. JSON on johdettu JavaScriptin ohjelmointikielistandardista ECMA-262 3. painos - joulukuu 1999. Tekstimuotoisuudestaan huolimatta JSON on täysin kieliriippumaton ja käyttää konventioita, jotka ovat tuttuja ohjelmointikielissä, kuten C, C++, C#, Java, JavaScript, Perl ja Python [19]. JSON on suosituin tietoformaatti API-ratkaisuissa, koska se muistuttaa läheisesti ohjelmointikielissä yleisesti käytettyjä tietorakenteita.

JSON perustuu kahteen pääasialliseen rakenteeseen:

1. Kokoelma nimi-/arvopareja. Eri kielissä tämä tunnetaan objektina, tietueena, rakenteena, sanakirjana, hajautustaulukkona, avainlistana tai assosiatiivisena taulukkona.
2. Järjestetty arvojoukko. Useimmissa kielissä tämä tunnetaan taulukkona, vektorina, listana tai sekvenssinä.

Nämä rakenteet ovat yleisesti tuettuja moderneissa ohjelmointikielissä, mikä tekee JSON-formaatista monipuolisen tiedonvaihtoformaatin ohjelmoinnissa.

JSON-formaatissa nämä rakenteet ovat seuraavanlaisia:

- Objekti on järjestämätön kokoelma nimi-/arvopareja. Objekti alkaa vasemmalla aaltosulkeella ja päättyy oikealla aaltosulkeella. [19]
- Jokaisen nimen jälkeen tulee kaksoispiste, ja nimi-/arvoparit erotetaan pilkuilla. [19]

JSON-tiedosto voi sisältää monia taulukoita (Kuva 4).

```
{
  "nimi": "Matti Meikäläinen",
  "ikä": 30,
  "sähköposti": "matti.meikalainen@example.com",
  "osoite": {
    "katu": "Esimerkkikatu 123",
    "kaupunki": "Helsinki",
    "postinumero": "00100"
  },
  "harrastukset": [
    "jalkapallo",
    "lukeminen",
    "ruoanlaitto"
  ]
}
```

Kuva 4. Esimerkki JSON-objektista.

Vaikkei opinnäytetyössä suoraan ladata JSON-tiedostoja, on silti tärkeää ymmärtää sen rakenne etenkin hyödynnettäessä ohjelmointirajapintoja.

CSV-tiedostomuoto (comma separated value) on yleisesti taulukoinnissa käytetty tiedostomuoto. Jokainen tietue sijaitsee omalla pilkulla erotetulla sarakkeella ja rivit erotetaan rivinvaihtomerkillä (Kuva 5). [20.]

```
nimi,ikä,sähköposti  
Matti Meikäläinen,30,matti.meikalainen@example.com
```

Kuva 5. Esimerkki CSV-tiedostosta.

CSV-tiedostoja käytetään laajalti tietojen siirtoon ja tallentamiseen eri ohjelmistojen välillä, koska ne ovat yksinkertaisia ja yhteensopivia useiden sovellusten kanssa. CSV sopii parhaiten suuren tietomäärän tallentamiseen, mutta se ei mahdollista monimutkaisten tietorakenteiden tallentamista JSON-formaatin tapaan.

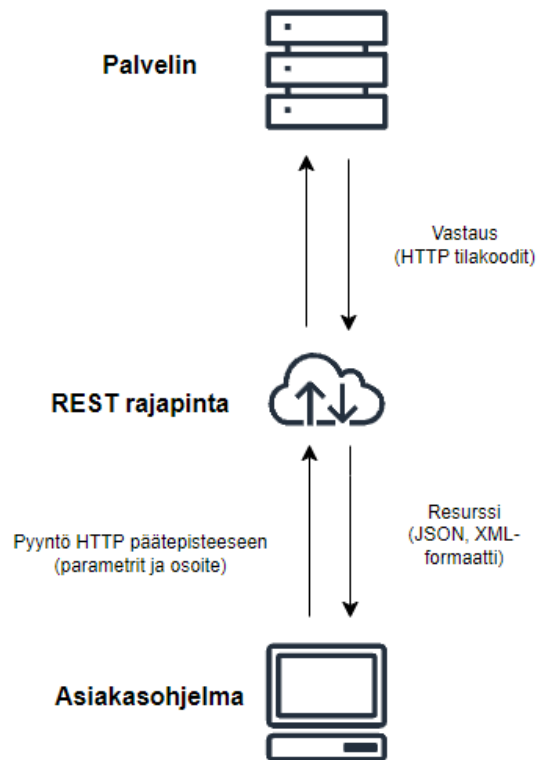
JSON ja CSV ovat käytetyimpiä tiedonvaihtoformaatteja yksinkertaisuuden ja helpon luettavuuden takia. Laajan käytön seurauksena kyseisten tiedonvaihtoformaattien ympärillä on olemassa ohjelmointikielissä monia työkaluja, jotka helpottavat niiden käsittelyä.

3.2 Ohjelmointirajapinta

Ohjelmointirajapinta on yleensä helposti toteutettava ja hyvin skaalautuva ratkaisu tilastojen haakuun. Ohjelmointirajapinnalla tarkoitetaan väylää, jonka avulla ohjelmat jakavat tietoa toistensa välillä. Yleisin lähestymistapa ohjelmointirajapinnan arkkitehtuuriin on REST (Kuva 6). Kyseisten rajapintojen avulla voidaan esimerkiksi tehdä tietokantoihin liittyviä toimintoja HTTP-pyyntöjä käyttäen. Tyypillisesti REST-rajapinnalla tehdyt CRUD-operaatiot rinnastetaan HTTP-metodeihin. Käytetyimmät HTTP-pyyntöihin liittyvät metodit ovat POST, jota käytetään tiedon luomiseen, GET tiedon hakemiseen, PUT tai PATCH päivittämiseen ja DELETE tiedon poistamiseen. HTTP-pyyntöihin liittyvien parametrien ja haettavien resurssien yhdenmukaisuus on tärkeä osa hyvin suunniteltua rajapintaa. [21.]

REST-rajapinnat käyttävät yleensä JSON-muotoista dataa, mikä tekee tietojen välittämisestä selkeää ja yksinkertaista eri ohjelmointikielillä. Tämä formaatti mahdollistaa helpon integraation ja datan vaihdon erilaisten sovellusten ja palveluiden välillä. Lisäksi REST-arkkitehtuurin keveys ja resurssilähtöisyys tekevät siitä suorituskykyisen ratkaisun, joka pystyy käsittelemään suuria tietomääriä ja monimutkaisia operaatioita ilman merkittäviä suorituskykyongelmia. REST-rajapinnan

dokumentointi on myös avainasemassa sen hyödyntämisessä, sillä selkeä ja kattava dokumentaatio auttaa kehittäjiä ymmärtämään ja käyttämään rajapintaa tehokkaasti. Yhteenvetona voidaan todeta, että hyvin suunniteltu ja dokumentoitu REST-rajapinta tarjoaa tehokkaan ja joustavan tavan toteuttaa monenlaisia sovellusten välisiä integraatioita. [21.]



Kuva 6. REST-tyyppisen web-rajapinnan toiminta.

3.3 Web scraping

Verkkosivujen haravoinnilla (engl. Web scraping) tarkoitetaan tiedon hakemista verkkosivun HTML-dokumentista. Haravointia tehdään ohjelmistorobotilla hakemalla verkkosivusta halutun osion koko HTML-dokumentti, minkä jälkeen siitä etsitään haluttu data, muutetaan se jäseneltyyn muotoon ja tallennetaan esimerkiksi Excel- tai JSON-tiedostoon. [22.]

Verkkosivulta haravoidun tiedon käyttöön voi liittyä rajoitteita eikä haravointi saa estää sivun normaalia toimintaa. Web scraperin toteuttamiseen liittyy myös monia haasteita, joita voivat olla botin rikkoutuminen sivun HTML-dokumentin muuttuessa tai sivun botineston läpäiseminen. Bo-

tin toteuttaminen on myös työlästä, sillä se joudutaan räätälöimään jokaiselle verkkosivulle sopivaksi. Sivuilla, joissa esitetään tietoa dynaamisesti, esimerkiksi JavaScriptin avulla, voivat olla haasteellisia haravointiboteille.

Tiedon hakeminen tällä tavalla on huomattavasti työläämpää verrattuna muihin tapoihin, joten sen toteuttamista kannattaa harkita vain, jos muuta liityntätapaa ei ole saatavilla. Haravointia tehtäessä tulee noudattaa tietosuojalakia, tekijänoikeuslainsäädäntöä ja verkkosivun käyttöehtoja. Botteihin liittyviä ehtoja voi tarkastella verkkosivun robots.txt-tiedostosta (Kuva 7). Tiedosto voi sisältää sallittuja tai ei-sallittuja verkkosivun osoitteita boteille.

```
# robots.txt file for YouTube
# Created in the distant future (the year 2000) after
# the robotic uprising of the mid 90's which wiped out all humans.

User-agent: Mediapartners-Google*
Disallow:

User-agent: *
Disallow: /api/
Disallow: /comment
Disallow: /feeds/videos.xml
Disallow: /get_video
Disallow: /get_video_info
Disallow: /get_midroll_info
Disallow: /live_chat
Disallow: /login
Disallow: /qr
Disallow: /results
Disallow: /signup
Disallow: /t/terms
Disallow: /timedtext_video
Disallow: /verify_age
Disallow: /watch_ajax
Disallow: /watch_fragments_ajax
Disallow: /watch_popup
Disallow: /watch_queue_ajax
Disallow: /youtubei/

Sitemap: https://www.youtube.com/sitemaps/sitemap.xml
Sitemap: https://www.youtube.com/product/sitemap.xml
```

Kuva 7. YouTube-videopalvelun robots.txt-tiedosto.

4 Esimerkkisovellus julkisen taloustiedon hakemiseen

Alaluvuissa käsitellään opinnäytetyön käytännön osion tavoitteita sekä sovelluksen suunnittelua, sen keskeisiä osa-alueita ja jatkokehitysmahdollisuuksia. Suunnitteluosiossa käydään läpi arkkitehtuuria, valittuja teknologioita ja datalähteitä. Toteutusosiossa esitetään sovelluksen työnkulun tärkeimpiä vaiheita koodiesimerkeillä. Jatkokehitysosiossa pohditaan nykyisen toteutuksen puutteita ja rajoitteita sekä tarkastellaan sovelluksen kehittämismahdollisuuksia, kuten uusia ominaisuuksia.

4.1 Tavoitteet ja suunnittelu

Opinnäytetyön kehittämisosuuden tavoitteena oli luoda yritykselle ratkaisu taloustiedon hakemiseen verkon julkiselta puolelta ohjelmallisesti. Sovelluksesta saatua dataa tulisi voida yhdistää toiminnanohjausjärjestelmästä saatuun dataan, mikä mahdollistaisi päätelmien ja ennusteiden luomisen talous- ja markkinatilanteesta.

Sovelluksen tärkeimmäksi toiminnoksi määriteltiin taloustiedon hakeminen pääasiassa tilastokeskuksesta, Yahoo Financesta ja Eurostatista.

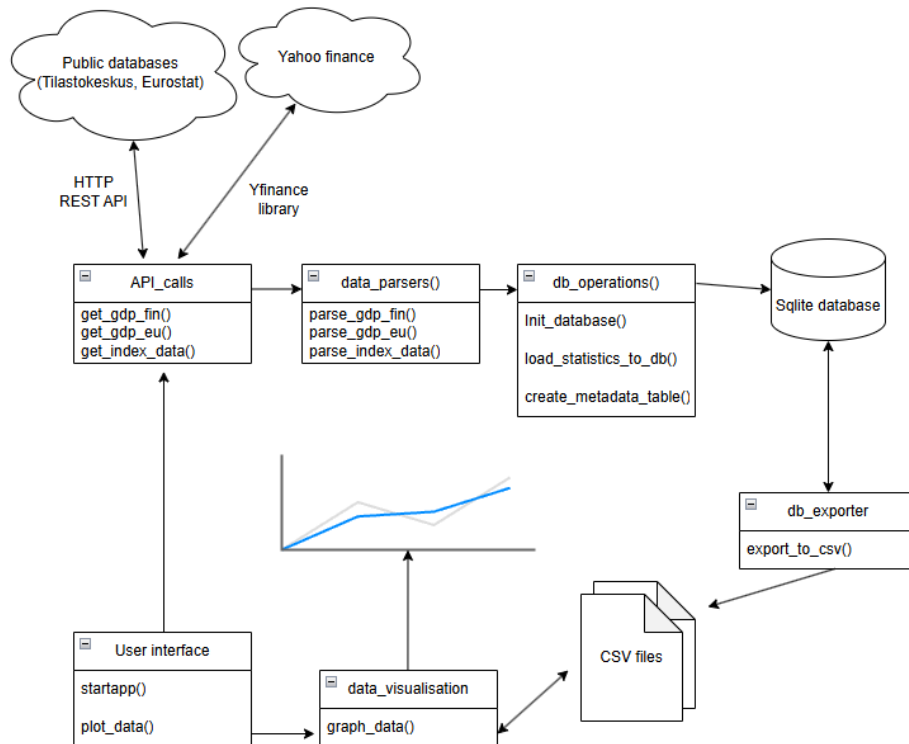
Julkisten osakkeiden historiallisia tietoja tulisi voida hakea myös sen lyhenteellä eli osakesymbolilla. Ratkaisun täytyi sisältää kevyt graafinen käyttöliittymä, josta ilmenee tiedon haun ja suodattamisen vaiheet sekä mahdolliset virheet suorituksessa. Käyttöliittymälle vaatimuksiksi asetettiin valintaruudut visualisoitaville tilastoille ja datan ajanjakson valinta pudotusvalikosta. Dataa täytyy pystyä aggregoimaan yhdelle kaavioille, jotta siitä voidaan muodostaa kokonaisvaltaisempi kuva talouskasvun kehityksestä.

Tietoa tulee myös voida päivittää pidemmällä aikavälillä, sillä API:t yleensä rajoittavat pyyntöjen määrää palvelunestohyökkäysten torjumiseksi. Tilastojen tulisi olla yhtenäisessä muodossa jatkokäyttöä ja sujuvaa datan visualisointia varten, mikä mahdollistaisi erilaisten talouskasvun mittausten ja niiden keskinäisten vaikutusten tarkastelun tarkoituksenmukaisten kaavioiden avulla.

Sovelluksen suunnittelu alkoi korkean tason arkkitehtuurin kartoittamisella. Arkkitehtuurissa kuvataan ohjelmiston osia ja niiden liittymistä toisiinsa. Ylätason kuvan muodostamisen jälkeen kehitettiin julkisia ohjelmointirajapintoja Postman-työkalulla, joka on tarkoitettu pääasiassa ohjelmointirajapintojen testaamiseen.

Demosovellukselle ohjelmointikieleksi valittiin vuonna 1991 julkaistu Python, joka on dynaamisesti kirjoitettu ja tulkittu kieli. Lisäksi se tukee olio-ohjelmointia ja sisältää automaattisen muistin vapauttamisen. Python on yksi suosituimmista ohjelmointikielistä datan käsittelyyn, koska se tarjoaa monipuolisen ja tehokkaan ympäristön datan muokkaamiseen, analysointiin ja visualisointiin. Pythonissa on useita datan käsittelyyn erikoistuneita kirjastoja, kuten NumPy, Pandas, Matplotlib ja Seaborn. NumPy ja Pandas tarjoavat numeeriseen laskentaan ja taulukoiden käsittelyyn tehokkaita työkaluja. Matplotlib- ja Seaborn-kirjastoilla voidaan toteuttaa datan visualisointia. Kuvassa 8 esitellään ohjelman työnkulkua, Python-moduuleita ja niiden sisältämiä funktioita (Kuva 8).

Ohjelman työnkulku alkaa käyttöliittymämoduulista (User interface). Seuraavaksi API_calls-moduuli hakee kaiken datan HTTP-pyynnöillä, data_parsers parsii JSON-datasta aika-arvoparin, db_operations alustaa sqlite tietokannan ja lataa parsitun datan siihen. Db_exporter-moduuli vie tietokannan sisällön CSV-tiedostoiksi, jonka jälkeen käyttäjä voi visualisoida valitut tilastot käyttöliittymästä. Visualisointi on toteutettu data_visualisation-moduuliin.



Kuva 8. Python-moduulit ja ohjelman työnkulun vaiheet.

Käyttöliittymää varten valittiin Tkinter Python-kirjasto. Tkinter-paketti on Pythonin standardira-japinta Tcl/Tk-graafisen käyttöliittymän työkaluihin [23]. Tkinter tarjoaa yksinkertaisen tavan luoda graafisia elementtejä ja soveltuu yksinkertaisiin käyttöliittymiin.

Requests-kirjasto mahdollistaa HTTP-pyyntöjen suorittamisen Python-koodissa. Se tukee kaikkia HTTP-metodeja, kuten GET, POST, PUT, DELETE, antaen kehittäjälle mahdollisuuden kommuni-koida verkkopalvelujen kanssa monipuolisesti. Requests-kirjastolla voidaan käsitellä erilaisia HTTP-vastauksia, kuten JSON-dattaa, joka voidaan suoraan parsia Python-olioiksi (Kuva 9). Kirjasto tukee myös HTTP-pyyntöjen uudelleenohjauksia ja automaattista sisältökoodausta. Lisäksi Sessi-ons-objektia käyttäen, voidaan säilyttää tila ja evästeet useiden pyyntöjen välillä, mikä tekee siitä erityisen hyödyllisen sovelluksissa, jotka vaativat kirjautumista tai muuta käyttäjäkohtaista kon-tekstia. [24.]

```

>>> import requests

>>> r = requests.get('https://api.github.com/events')
>>> r.text
'{"repository":{"open_issues":0,"url":"https://github.com/...

```

Kuva 9. GET-pyyntö Requests-kirjastolla sekä vastauksen tulostus tekstimuodossa Pythonin konsoliin.

4.2 Toteutus

Sovellus hakee Suomen ja Euroopan bruttokansantuotteen, Suomen kuluttajahintaindeksin sekä indeksirahastojen tietoja. Requests-kirjastolla tehdään HTTP-pyyntöjä kaikkiin osoitteisiin, joista tieto halutaan ohjelman käynnistyessä. Palvelimen päässä olevat virheet käsitellään, jottei ohjelman odotettu toiminta häiriintyisi. Tilastokeskuksen ja Eurostatin tilastoihin liityttiin niiden tarjoamilla ohjelmointirajapinnoilla ja Yahoo Financen dataan liityttiin yfinance-kirjastolla. Suomen talouteen liittyviä tilastoja haettiin tilastokeskuksesta, Euroopan talouteen Eurostatista ja osake- sekä indeksirahastojen tilastoja Yahoo Financesta.

Tieto suodatetaan valmiiksi määriteltyihin Python-olioihin, mikä helpottaa sen käsittelyä sekä laa- taamista tietokantaan ja tukee myös koodin modulaarisuutta. Tietokanta toimii ei-elintärkeänä välikätenä tämänhetkisessä toteutuksessa, mutta sitä voidaan käyttää dataväylänä sekä datamal- lin esimerkkinä jatkokehityksessä. Tietokantaratkaisuna käytettiin sqlite-tietokantamoottoria sekä Pythonin sqlite3-rajapintaa. Sqlite soveltui toteutukseen parhaiten, sillä monimutkaisia re- laatioita ei tarvittu. Sqlite tarjoaa myös helpon tavan viedä datan muunlaisiin tietokantoihin, esi- merkiksi CSV-tiedostona. Tietokantaan alustetaan myös metadata-taulu tiedon ymmärtämisen helpottamiseksi. Metadata voi sisältää tietoa taulukon sarakkeista, tietolähteestä ja yleisen ku- vauksen taulukosta.

Ohjelman käynnistyessä ensimmäinen vaihe on raa’an taloustiedon hakeminen. Tätä voidaan myös kuvailla ETL- prosessin extract-vaiheena. Kuvassa 10 on koodiesimerkki Tilastokeskuksen tietokantaan HTTP-pyyntön suorittamisesta, joka hakee Suomen bruttokansantuotteen vuosinel- jänneksittäin. Poikkeuksellisesti Tilastokeskuksesta haetaan tietoa POST-tyyppisellä pyynnöllä, GET-pyyntö palauttaa metadatan kyseisestä taulukosta. Funktio palauttaa API:lta saadun datan JSON-muodossa.

```

8 def get_gdp_fin():
9     url = constants.FIN_GDP_URL
10
11     payload = json.dumps({
12         "query": [
13             {
14                 "code": "Taloustoimi",
15                 "selection": {
16                     "filter": "item",
17                     "values": [
18                         "BIGMH"
19                     ]
20                 }
21             },
22             {
23                 "code": "Tiedot",
24                 "selection": {
25                     "filter": "item",
26                     "values": [
27                         "kausitcp"
28                     ]
29                 }
30             }
31         ],
32         "response": {
33             "format": "json-stat2"
34         }
35     })
36     headers = constants.JSON_HEADER
37
38     try:
39         response = requests.request("POST", url, headers=headers, data=payload)
40         response.raise_for_status()
41
42         return response.json()
43
44     except RequestException as e:
45         logging.error(f"Request to StatFin API failed: {e}")
46
47     return None

```

Kuva 10. HTTP-pyyntö Tilastokeskuksen julkiseen tietokantaan.

Transform-vaiheessa API:lta saadusta JSON-datasta muodostetaan selkeä aika-arvo-pari ja se muutetaan Python-objekteiksi. Python-objekteina yksittäisiä rivejä pystytään käsittelemään helposti ohjelmointikielen tietotyyppien ominaisuuksia käyttäen. Ohjelmointikielen tietotyyppiä muuntaminen mahdollistaa myös vaivattoman validoinnin, joka parantaa myös sovelluksen tietoturva. Kuvassa 11 on esitetty Suomen bruttokansantuotteen parsimisfunktio, joka ottaa JSON-datan argumenttina, hakee siitä ajan sekä arvon, muuntaa sen listaksi Python-objekteja, poistaa tyhjät arvot ja palauttaa listan.

```

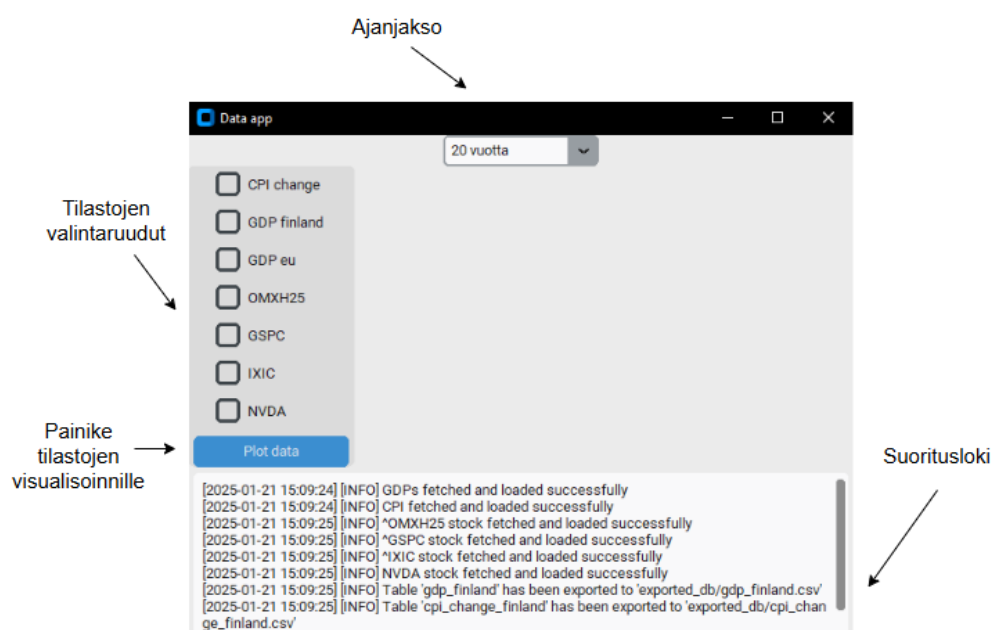
5 def parse_gpd_fin_json(raw_data) -> List[data_objects.Gdp_fin]:
6     quarters_gdp = list(raw_data['dimension']['Vuosineljannes']['category']['label'].values())
7     gdp = raw_data['value']
8
9     # Values to Gdp_fin objects
10    gross_domestic_products = [data_objects.Gdp_fin(quarter, value) for quarter, value in zip(quarters_gdp, gdp)]
11
12    for gross_domestic_product in gross_domestic_products:
13        if gross_domestic_product.value is None:
14            gross_domestic_products.remove(gross_domestic_product)
15
16    return gross_domestic_products

```

Kuva 11. Esimerkki parsimisfunktio.

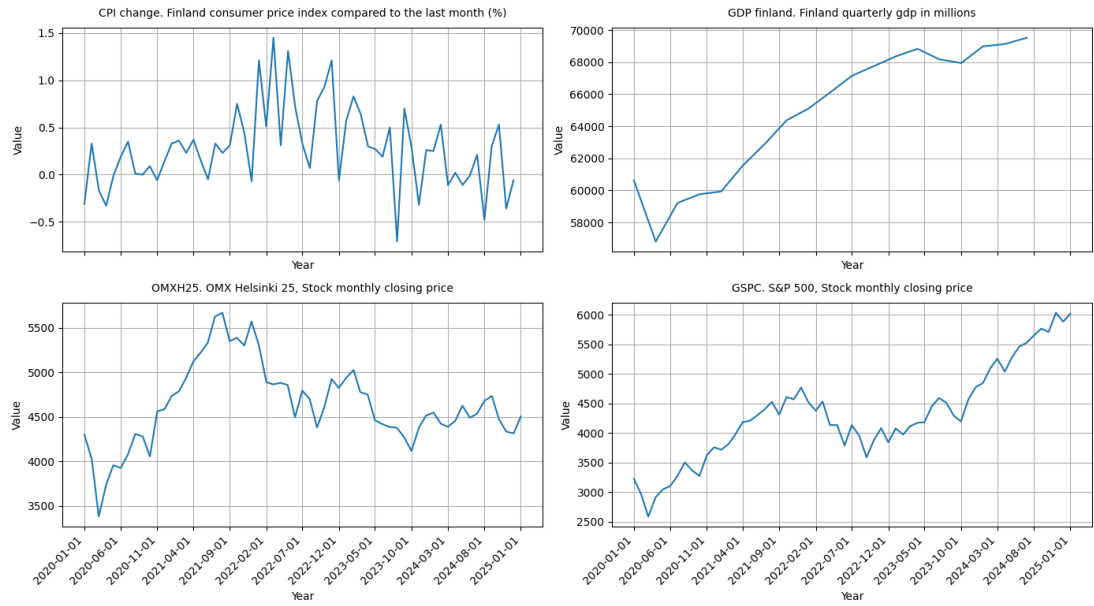
Parsimisen jälkeen alustetaan tietokantataulut sekä ladataan data tietokantaan, minkä jälkeen tietokannan sisältö vietään CSV-tiedostoiksi.

Ohjelman käyttöliittymästä nähdään suorituksen vaiheet sekä mahdolliset virheet suorituslokkissa. Visualisoinnin parametrit, kuten ajanjakso ja esitettävät tilastot, voidaan päättää pudotusvalikosta ja valintaruuduista (Kuva 12).



Kuva 12. Sovelluksen käyttöliittymän rakenne

Visualisointia varten kaikkien tilastojen aikavälit muutetaan täysin numeerisiksi, esimerkiksi 1.7.2024 -> 2024.5, jotta niitä on helpompi käsitellä rinnakkain. Plot data -painike esittää kaikki valitut tilastot kaavioina (Kuva 13).



Kuva 13. Suomen kuluttajahintaindeksi, Suomen bruttokansantuote, Helsingin pörssi ja S&P 500 indeksi viime viiden vuoden aikana.

5 Yhteenveto

Kaiken kaikkiaan työn lopputuloksena saatiin toteutettua toimiva ja laajennettavissa oleva talousdataa hyödyntävä sovellus. Sovellus osoittautui käyttökelpoiseksi pohjaksi julkisen taloustiedon käsittelyyn, ja sitä voidaan hyödyntää jatkossa esimerkiksi visualisointien tai analyytiikkapalvelujen kehittämisessä. Työ toimii myös esimerkkinä siitä, kuinka avoimen datan lähteitä voidaan yhdistellä ja hyödyntää ohjelmallisesti osana tietopohjaista päätöksentekoa.

Sovelluksen suunnitteluvaiheessa tutkittiin myös web scraping -tekniikoita osana vaihtoehtoisia tiedonhankintamenetelmiä. Näiden menetelmien käyttöön ei kuitenkaan päädytty, sillä niiden toteuttaminen olisi vaatinut huomattavaa työmäärää sekä jatkuvaa ylläpitoa, esimerkiksi muuttuvien verkkosivurakenteiden takia. Lisäksi web scrapingiin liittyy usein oikeudellisia ja eettisiä kysymyksiä. Kyseisiä haasteita jouduttiin pohtimaan yfinance-kirjaston käytön yhteydessä.

Sovelluksen yhdistämistä Power BI -palveluun tutkittiin yhtenä mahdollisena jatkokehityssuuntana. Integraatiota kokeiltiin, mutta sen toteuttaminen sujuvalla ja käyttäjäystävällisellä tavalla osoittautui haasteelliseksi opinnäytetyöhön käytettävissä olevilla resursseilla. Mahdollisuus hyödyntää Power BI:tä raportoinnin ja visualisoinnin välineenä jäi kuitenkin avoimeksi jatkokehitysideaksi.

Koodista voisi tehdä modulaarisemman ja uusien tilastojen lisäämiselle voisi dokumentoida tai kehittää "kehiksen", joka tekisi tilastojen lisäämisestä suoraviivaisemman kehittäjälle. Kehys voisi olla konfiguroitava ja automatisoitu prosessi, joka poistaisi melkein kaiken manuaalisen ohjelmoinnin tarpeen. Koodin muuntaminen yleiskäyttöisemmäksi parantaa myös sen testattavuutta sekä ylläpidettävyyttä. Vaikeammin saavutettaville tilastoille, kuten verkkosivuilla esitettäviin kaavioihin, voitaisiin myös kehittää web scraping -liityntä. Tämänhetkisessä toteutuksessa kaikki tilastot eivät ole myöskään samalla aikavälillä, joten suuremman aikavälin tilastoille voisi etsiä tarkemman tietolähteen tai dataa voisi interpoloida mahdollisesti kuukauden aikaikkunaan.

Talousdataa voitaisiin hakea vielä monipuolisemmin. Mukaan voitaisiin lisätä esimerkiksi työllisyyteen ja työmarkkinoihin liittyviä tilastoja, kuten työttömyysaste ja avoimien työpaikkojen määrä. Lisäksi kattavuutta voitaisiin laajentaa muihin talouden osa-alueisiin, kuten julkiseen talouteen, rahoitusmarkkinoihin, kulutuskäyttäytymiseen, kansainväliseen kauppaan sekä palkka- ja tulokehitykseen.

Sovelluksesta tuotetun tietokannan voisi siirtää Azure-pilvipalveluun. Pilvipalvelusta sitä voitaisiin hyödyntää esimerkiksi reaaliaikaiseen Power BI -raporttiin, joka olisi mahdollista upottaa esimerkiksi toiminnanohjausjärjestelmän käyttöliittymään. Toinen vaihtoehto tilastojen esittämiselle muussa sovelluksessa olisi viedä tilastojen kaaviot kuvina ja upottaa ne verkkosivulle. Tämä tapa vaatisi visualisoinnin ja erityisesti ulkonäön parantamista. Visualisointia voitaisiin parantaa esimerkiksi lisäämällä kaaviotyyppejä.

Opinnäytetyö tarjosi tekijälleen kattavan katsauksen avoimen datan hyödyntämiseen ja taloudellisen tiedon käsittelyyn ohjelmallisesti. Työn aikana kertyi merkittävää teknologiaosaamista erityisesti ohjelmoinnista ja REST-rajapinnoista sekä niiden tehokkaasta käytöstä. Lisäksi projektin aikana syvennyttiin datan esikäsittelyyn, visualisointiin ja eri tietolähteiden arviointiin. Yhteenvedon voidaan todeta, että työ tarjosi vahvan pohjan datalähtöisen sovelluskehityksen ymmärtämiseen ja toteuttamiseen.

Lähteet

1. What is public data. Tech target. [Internet]. [viitattu 1.7.2024]. Saatavilla: <https://www.techtarget.com/searchcio/definition/public-data>
2. Mitä on avoin data. HRI. [Internet]. [viitattu 10.6.2024]. Saatavilla: <https://hri.fi/fi/ohjeet/mita-on-avoin-data/>
3. Mitä hyötyä on avoimesta datasta. Avoindata. [Internet]. [viitattu 11.6.2024]. Saatavilla: <https://www.avoindata.fi/fi/tietoa-avoimesta-datasta/mita-on-avoin-data#avoin-data-hyodyt>
4. IG. What are the key macroeconomic indicators. [Internet]. [Viitattu 13.11.2025]. Saatavilla: <https://www.ig.com/en/trading-strategies/what-are-the-key-macroeconomic-indicators-to-watch--191014>
5. The conference board. How to forecast GDP in the long run: A primer. [Internet]. [Viitattu 22.1.2025]. Saatavilla: <https://www.conference-board.org/publications/how-to-forecast-GDP-in-the-long-run>
6. Tilastokeskus. Tietoa meistä. [Internet]. [Viitattu 22.1.2025]. Saatavilla: <https://stat.fi/org/index.html>
7. Tilastokeskus. Tilastotieto. [Internet]. [Viitattu 22.1.2025]. Saatavilla: <https://stat.fi/tilastotieto>
8. Avoin data. Tilastokeskus. [internet]. [viitattu 13.6.2024]. Saatavilla: <https://stat.fi/org/avoindata/index.html>
9. About yahoo finance. Yahoo finance. [Internet]. [viitattu 18.7.2024]. Saatavilla: <https://finance.yahoo.com/about>
10. Yfinance readme. GitHub. [Internet]. [viitattu 18.7.2024]. Saatavilla: <https://github.com/ranaroussi/yfinance>
11. Eurostat. Who we are. [Internet]. [Viitattu 30.9.2024]. Saatavilla: <https://ec.europa.eu/eurostat/web/main/about-us/who-we-are>
12. Eurostat. Quality Assurance Framework of the European Statistical System. [Internet]. [Viitattu 22.1.2025]. Saatavilla: <https://ec.europa.eu/eurostat/documents/64157/4392716/ESS-QAF-V2.0-final.pdf>
13. Eurostat. API. [Internet]. [Viitattu 30.9.2024]. Saatavilla: <https://wikis.ec.europa.eu/display/EUROSTATHELP/API+-+Getting+started>
14. What is ETL. Codeless platforms. [Internet]. [viitattu 19.6.2024]. Saatavilla: <https://www.codelessplatforms.com/blog/what-is-etl/>

15. What is an ORM. Freecodecamp. [Internet]. [viitattu 8.7.2024]. Saatavilla: <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/>
16. Rivery. Different types of data ETL-data transformation. [Internet]. [Viitattu 22.1.2025]. Saatavilla: <https://rivery.io/data-learning-center/types-of-etl-data-transformation/>
17. Luo datapohjaista kulttuuria kaikille BI:n avulla. Microsoft. [Internet]. [viitattu 22.8.2024]. Saatavilla: <https://www.microsoft.com/fi-fi/power-platform/products/power-bi>
18. What is an API. IBM. [Internet]. [Viitattu 29.8.2024]. Saatavilla: <https://www.ibm.com/topics/api>
19. Introducing JSON. [Internet]. [Viitattu 22.7.2024]. Saatavilla: <https://www.json.org/json-en.html>
20. Y. Shafranovich. Common Format and MIME Type for Comma-Separated Values (CSV) Files. [Internet]. [Viitattu 29.9.2024]. Saatavilla: <https://www.ietf.org/rfc/rfc4180.txt>
21. What is a REST API. IBM. [Internet]. [viitattu 11.6.2024]. Saatavilla: <https://www.ibm.com/topics/rest-apis>
22. What is web scraping and how to use it. Geeksforgeeks. [Internet]. [viitattu 29.8.2024] Saatavilla: <https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/#what-is-web-scraping>
23. Python docs. Tkinter – Python interface to Tcl/Tk. [Internet]. [Viitattu 13.1.2025]. Saatavilla: <https://docs.python.org/3/library/tkinter.html>
24. Requests: HTTP for Humans. [Internet]. [viitattu 28.6.2024]. Saatavilla: <https://requests.readthedocs.io/en/latest/>

Kuvaus tiedonhaun prosessista

Tietoa opinnäytetyöhön haetaan suurimmaksi osaksi avoimesta verkosta. Tietokantoina voidaan käyttää valtion virastojen tai niiden tuottamien verkkopalveluiden, kuten tilastokeskus tai suomi.fi-avoindata, dokumentaatioita ja/tai käyttöohjeita. Ohjelmistokehityksessä käytettäviin teknologioihin tai tekniikoihin liittyvää tietoa haetaan alaan liittyvistä blogeista tai ohjesivuista, kuten IBM, Freecodecamp tai codelessplatform. Tietoa teknologioista, kuten ohjelmistokirjastoista, saadaan kyseisen teknologian virallisesta dokumentaatiosta, kuten Github tai json.org-verkkosivulta.

Lähdekriittisyyttä noudatetaan valitsemalla konkreettisille aiheille sen virallinen dokumentaatio, jos se on saatavilla. Ohjekirjoista tai blogeista tietoa haettaessa tärkeimpinä kriteereinä ovat kirjoittajan tausta ja julkaisualustan maine. Ajantasaisuuteen julkaisupäivämäärinä tai versiotietoina kiinnitetään myös huomiota.

Tiedonhaussa käytettäviä asiasanoja voivat olla:

- Data, avoin data, julkinen data
- ETL, data warehouse, data warehousing process
- Web scraping, application programming interface, JSON