

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristina Sizova 193922IADB

Veebirakendus elektriseadmete haldamise ja elektrimüügi automatiseerimiseks

Bakalaureusetöö

Juhendaja: Meelis Antoi
Magistrikraad

Tallinn 2023

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kristina Sizova

28.10.2022

Annotatsioon

Käesolevas lõputöös käsitletakse kaasaegseid elektrienergia tarbimiste ja tootmiste puudutavaid probleeme. Probleemidega on silmitsi seisnud paljud elektritarbijad, sealhulgas väiketootjad. Tänu taastuvenergiale on lisandunud palju uusi võimalusi, samas kui kaasaegsed tehnoloogiad võimaldavad lahendada need probleemid innovaatsiliselt. Üheks võimalikuks lahenduseks on käesolevas töös välja töötatud rakenduse prototüüp.

Arendusprotsessi käigus planeeritakse luua lahendus, mis vähendab elektriarvete maksumust, võimaldab tarbida elektrit efektiivsemalt ja säästlikumalt ning automatiseerib elektriseadmete tööd lähtuvalt kasutaja vajadustest. Lõputöö sisaldab hetkeolukorra uuringut, alternatiivsete lahenduste kirjeldust ja nõuete määratlemist loodavale seadmele. Töö kokkuvõttes osas vaadeldakse IT-lahenduse edasiarendamise väljavaateid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 4 peatükki, 17 joonist, 8 tabelit.

Abstract

Web Application for Automation of Electrical Equipment Management and Electricity Sales

This thesis deals with modern problems of consumption and production of electricity. The problem has affected many consumers of electricity, including small producers. Renewable energy has brought many new opportunities and modern technologies can solve this problem through innovations. One of the possible solutions is the application prototype developed in this work.

During the development process, it is planned to create a solution that will reduce the cost of electricity bills, consume electricity more efficiently and economically, and automate the operation of electrical equipment based on the needs of the user. The dissertation includes a study of the current situation, a description of alternative solutions and the definition of requirements for the device being created. Ultimately, the prospects for further development of the IT solution are considered.

The thesis is in Estonian and contains 30 pages of text, 4 chapters, 17 figures, 8 tables.

Lühendite ja mõistete sõnastik

Apache	tasuta avatud lähtekoodiga tarkvara, mis võimaldab kasutajatel oma veebisaite Internetis juurutada
API	<i>Application Programming Interface</i> , viis kahe või enama arvutiprogrammi omavaheliseks suhtlemiseks
Composer	käsurea tööriist sõltuvuse haldamiseks PHP-s
CRUD	akronüüm neljale andmebaasiga töötamisel kasutatavale põhifunktsioonile: loomine, lugemine, muutmine, kustutamine
cURL	<i>Client URL</i> , tööriist andmete edastamiseks serverist või serverisse
DOM	<i>Document Object Model</i> , HTML-dokumendi liides
Gii	automaatne koodi genereerimise tööriist, mis võeti kasutusele versioonis Yii 2.0
HTML	<i>HyperText Markup Language</i> , standardiseeritud hüperteksti märgistuskeel veebilehtede vaatamiseks brauseris
HTTP	<i>HyperText Transfer Protocol</i> , võrguprotokoll, mis aitab veebilehitsejatel serveritega suhelda veebis
IIS	<i>Internet Information Services</i> , serverite komplekt mitme Microsofti Interneti-teenuse jaoks
kW	Kilovatt, elektrienergia tarbimisvõimsuse mõõtühik
kWh	Kilovatt-tund, elektrienergia tarbimisvõimsuse mõõtühik ajas
MQTT	<i>MQ Telemetry Transport</i> , sõnumsideprotokoll, mis pakub piiratud ressurssidega võrguklientidele lihtsat viisi telemetria teabe levitamiseks madala ribalaiusega keskkondades
MVC	<i>Model View Controller</i> , rakendustes kasutatav arhitektuur, mis jaotab rakenduse kolmeks loogiliseks osaks
RDBMS	<i>Relational Database Management System</i> , tarkvarasüsteem, mis võimaldab juurdepääsu relatsioonilisele andmebaasile
REST	<i>REpresentational State Transfer</i> , arhitektuurne stiil standardite pakkumiseks arvutisüsteemide vahel veebis,

	hõlbustades süsteemide omavahelist suhtlemist
TOU	<i>Time-Of-Use</i> , kommunaalarbija energiatarbimise mõõtmise ja laadimise meetod, mis põhineb energia kasutusajal
Wi-Fi	<i>Wireless Fidelity</i> , traadita andmeside standard
WLAN	<i>Wireless Local Area Network</i> , traadita side kohtvõrk
VPS	<i>Virtual Private Server</i> , virtuaalmasin, mis pakub virtualiseeritud serveriressursse füüsilises serveris, mida jagatakse teiste kasutajatega
XSS-i rünnak	<i>Cross-Site Scripting</i> , rünnak veebisüsteemide vastu, mis seisneb pahatahtliku koodi sisestamises veebisüsteemi väljastatud lehele ja selle koodiga suhtlemises ründaja veebiserveriga

Sisukord

1 Sissejuhatus.....	10
1.1 Metoodika.....	11
2 Probleemi ülevaade.....	12
2.1 Olemasolevad lahendused.....	13
3 Loodava lahenduse analüüs.....	14
3.1 Nõuete määratlemine.....	14
3.2 Maksimaalse efektiivsuse saavutamine.....	15
3.2.1 Päikeseenergia.....	18
3.3 Loodava lahenduse tehnoloogia valik.....	19
3.3.1 Kasutatav riistvara.....	19
3.3.2 Programmeerimiskeele valik.....	21
3.3.3 Teenusepoolse raamistiku valik.....	24
3.3.4 Kliendipoolne tehnoloogia.....	27
3.3.5 Andmebaasi valik.....	28
3.4 Analüüsi kokkuvõte.....	30
4 Lahenduse loomine.....	31
4.1 Börsihindade saamine.....	33
4.2 Andmebaas.....	34
4.3 Valminud lahenduse tulemused.....	36
4.4 Valminud lahenduse võimalikud edasiarendused.....	38
5 Kokkuvõte.....	39
Kasutatud allikate loetelu.....	40
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	43
Lisa 2 - Fronius Symo 6.0-3-M.....	44
Lisa 3 – Sonoff TH-16.....	45
Lisa 4 – Yii ja Laravel raamistike võrdlus.....	46
Lisa 5 – Marsruutimine Sonoffi, kliendi ja VPS-i vahel.....	48
Lisa 6 – Päikeseivnerterilt saadud fail.....	49
Lisa 7 – Elektri börsihinnad 29.11.2022.....	50

Jooniste loetelu

Joonis 1. Börsihinnad Eestis 22.11.2022.....	16
Joonis 2. Elektritarbimine Eestis 22.11.2022	17
Joonis 3. Süsteemi plokk skeem	18
Joonis 4. Tarbimine ja tootmine kuude kaupa	19
Joonis 5. Yii rakendusobjekti loomise programmikoodi näide	31
Joonis 6. Muutuja, mis sisaldab Vaade sisu programmikoodis	32
Joonis 7. Projekti struktuur	33
Joonis 8. Börsihidade saamine programmikoodis	34
Joonis 9. Seadme konfiguratsiooni tabel	35
Joonis 10. Päikeseinverterilt tulevate andmete tabel	35
Joonis 11. Börsihidade tabel	36
Joonis 12. Boileri sisselülitamine	37
Joonis 13. Boileri väljalülitamine	37
Joonis 14. Fronius Symo 6.0-3-M	44
Joonis 15. Sonoff TH-16	45
Joonis 16. Marsruutimine süsteemis.....	48
Joonis 17. Programmikood päikeseinverterilt failis	49

Tabelite loetelu

Tabel 1. Java programmeerimiskeele plussid ja miinused	21
Tabel 2. Python programmeerimiskeele plussid ja miinused	22
Tabel 3. PHP programmeerimiskeele plussid ja miinused	22
Tabel 4. C# programmeerimiskeele plussid ja miinused.....	23
Tabel 5. Programmeerimiskeelte võrdlus.....	23
Tabel 6. Yii ja Laravel raamistike võrdlus	46
Tabel 7. Börsihinnad 29.11.2022 00.00–07.00.....	50
Tabel 8. Börsihinnad 29.11.2022 07.00–23.00.....	50

1 Sissejuhatus

Kogu meie igapäevaelu on seotud paljude elektriseadmete kasutamisega, mis tagavad meile vajaliku mugavuse, kuid paraku kahjustavad meie keskkonda. Sellega seoses on energiasäästu teema tänapäeval väga aktuaalne. Elektrienergia säästmine vähendab niigi piiratud loodusvarade tarbimist, kahjulike ainete õhku paiskamist ja annab seega mõistliku panuse meie looduse säilitamisse. Pealegi, vähendab elektri säästmine oluliselt kommunaalkulusid, mis on väga oluline, sest elektri hinnad on viimase aasta jooksul märgatavalt tõusnud.

Elektri börsihind muutub iga tunni järel nõudluse ja pakkumise tasakaalu tulemusena. Elektrimüüjad esitavad iga päev börsile tellimuse, lähtudes sellest, kui palju nende klientidel järgmisel päeval elektrit vaja läheb. Turunõudluse rahuldamiseks võetakse esmalt kasutusele taastuv- ja tuumaenergia tootmisvõimsused. Nende toodang on soodsam, kuna energiaallikad on odavad ja tootmise käigus ei eraldu süsihappegaasi. Kui taastuvenergiast piisab tarbimise katmiseks, siis on ka elektri hind turul madal. Kui taastuvad energiaallikad nõudlust ei rahulda, pääsevad turule põlevkivi-, söe- ja gaasielektrijaamad. Nende toodetav elekter maksab nii kütusekulu kui ka süsinikdioksiidi heitkoguste tasu tõttu rohkem. Elektri börsihinna määrab viimane tarnija. Kui näiteks gaasiküttega töötav kõrgemate energiakuludega elektrijaam katab turunõudluse viimasena, siis määrab see konkreetsel tunnil elektrihinna [1].

Kuna elektritarbijad ei suuda börsihinna muutustele piisavalt kiiresti reageerida, kaasneb sellega kulukas ja ebaefektiivne elektritarbimine. Infotehnoloogia võib aga aidata seda olukorda muuta. Sõltuvalt elektrivõrgu pingest võimaldab igal ajahetkel muutuv hind koos sellele kiiresti reageerinud tarbijate koormuse muutusega toota kogu vajaliku elektri taastuvatest allikatest.

Käesoleva lõputöö eesmärgiks on luua prototüüplahendus, mis võimaldab muutuvast börsihinnast ja kasutaja isiklikest vajadustest lähtuvalt elektriseadmeid sisse ja välja lülitada. Bakalaurusetöös välja pakutud IT-lahendus aitab kaasa elektrienergia

efektiivsemale tarbimisele, samuti vähendab elektritarbijate jaoks börsihinna alusel arvestatavaid kulusid.

1.1 Metoodika

Lõputöö kirjutamisel kirjeldatakse esmalt probleemi ja selle probleemi olemasolevaid lahendusi. Seejärel pakutakse välja uus IT-lahendus ja mõeldakse selle omadustele.

Töö käigus analüüsitakse, kuidas saavutada loodava prototüübi maksimaalset efektiivsust, määratakse funktsionaalsed nõuded ning põhjendatakse vajalike tööriistade valikut, millega rakendust arendatakse. Lisaks käsitletakse töös prototüübi edasiarendamise võimalusi.

2 Probleemi ülevaade

Elektrihindade tõusu probleem muutub üha aktuaalsemaks. Ebaratsionaalne elektritarbimine ei too kaasa mitte ainult kõrgeid elektriarveid, vaid ka globaalseid tagajärgi, mis tulenevad meie planeedi saastumisest taastumatutest energiaallikatest elektri tootmisel eralduvate kahjulike ainetega.

Üks ilmsemaid viise, kuidas energiakasutuse vähendamine keskkonda aitab, on elektrijaamade heitkoguste vähendamine. Elektri tootmiseks põletavad enamik elektrijaamu kivisütt, toornaftat või muid fossiilkütuseid. Kuigi see energia tootmise meetod on suhteliselt odav, kahjustab see meie keskkonda, vabastades süsihappegaasi, vääveldioksiidi, lämmastikoksiide ning muid traditsiooniliste energia tootmise meetodite kõrvalsaadusi [2].

Süsinikdioksiid, mis moodustab suurema osa kogu õhusaastest, on kasvuhoonegaas. Kui süsinikdioksiid paisatakse õhku, neelab see päikese soojust ja salvestab soojust meie atmosfääri. See "kasvuhooneefekt" on loomulik nähtus ja vajalik Maal ellujäämiseks. Kuid kuna elektrijaamad põletavad rohkem kütust, et toota rohkem energiat, neelavad täiendavad süsinikujäätmel liiga palju soojust. See võib viia katastroofiliste tagajärgedeni Maal ning seada ohtu meie elu. Kasvuhoonegaaside heitkoguste tagajärjed on järgmised [2]:

- Temperatuuri tõus ja põud;
- Kõrgem meretase;
- Ebatavalised ilmastikutingimused;
- Loodusõnnetuste suurenev intensiivsus;
- Sudu ja happevihmad.

Energiatarbimise langetamine vähendab elektrienergia kogust, mida elektrijaamad peavad tootma, ja seega ka iga päev põletatavate fossiilkütuste kogust. Tarbijad saavad vähendada sõltuvust fossiilkütustest, kasutades vähem energiat ja otsides jätkuvalt uusi

viise, kuidas parandada säästvate energiavõimaluste, nagu päikese-, tuule- ja hüdroenergia, kasutamist. Mida rohkem hakkame toetuma taastuvatele energiaallikatele, seda rohkem saame pikendada fossiilkütuste eluiga ja aeglustada fossiilkütuste hinnakasvu.

2.1 Olemasolevad lahendused

Käesoleva probleemi lahendamiseks on tänapäeval palju viise. Mõned neist on järgmised:

- **Teak Smart Cube modular cube-shaped power outlet** [3]: AI-toega eMission Control jätab meelde tarbijakäitumise, kohalike kommunaalteenuste andmed ning lülitab seejärel toite sisse ja välja;
- **Amazon Smart Thermostat** [3]: reguleerib sisetemperatuuri automaatselt vastavalt sellele, kas kasutaja on kodus, voodis või õues. Funktsiooni Hunch kasutades tagab see nutiseade sisekeskkonna sobiva temperatuuri seadistamise. Lisaks ühildub see nutikas sisetermostaat Alexaga, võimaldades teil ajastada kasutaja eelistusi. Samuti õpib see kohanema kasutaja rutiiniga.
- **iDevices Smart Wall Outlet** [3]: WiFi-toega tehnoloogia töötab koos HomeKiti, Alexa ja Google Assistantiga. See nutikodu vidin võimaldab juhtida koduseadmeid lihtsa häälkäsklusega ning individuaalse pistikupesa juhtimise abil juhtida kahte seadet eraldi. Seega saab erinevate seadmete töö seadistada selliselt, et need töötaksid erinevate graafikute alusel. Lisaks on võimalik rakenduse iDevices abil jälgida energiatõhusust ja automatiseerida oma kodu;
- **Ecobee3 lite smart Wi-Fi thermostat** [3]: kütab või jahutab kodu automaatselt, kui elekter on odavam. Veelgi enam, see seade jätab meelde kasutaja rutiini ja pakub energiasäästunõuandeid;
- **LED-valgustid** [4]: kasutavad 85% vähem energiat ja kestavad 25 korda kauem kui traditsioonilised lambid.
- **Kõrge kasuteguriga soojuspumbad** [4]: energiasäästlik alternatiiv kodu kütmiseks ja jahutamiseks, kõrge efektiivsusega soojuspumbad edastavad soojust. Talvekuudel toovad need soojust koju, kuid suvel eemaldavad selle. Soojuse ülekandmine kulutab palju vähem energiat kui selle genereerimine või kompressori kasutamine õhu jahutamiseks. Uuematel soojuspumpadel on tõhusamad ventilaatorid ja need kasutavad erinevaid energiaallikaid, sealhulgas maasoojust.

3 Loodava lahenduse analüüs

Käesolevas peatükis analüüsitakse tehnoloogilise lahenduse loomiseks vajalikke eeldusi. Analüüs hõlmab funktsionaalsete nõuete määratlemist, efektiivseima ja ökonoomseima kasutaja vajadusi arvestava lahenduse otsimist, prototüübi loomiseks vajalike vahendite valikut.

3.1 Nõuete määratlemine

Probleemi lahendamiseks on vaja määrata tehnilised spetsifikatsioonid, mille alusel töötatakse välja tehnoloogiline lahendus. Allpool on loetletud funktsionaalsed nõuded:

- Süsteemi kasutaja peab saama vajadusel seadmeid iseseisvalt sisse ja välja lülitada;
- Süsteemi kasutaja peab kalendrist saama vaadata valitud päeva elektri tarbimise, ostu ja müügi statistikat;
- Süsteem peab iga päev skaneerima elektrituruhindu ja kuvama need veebilehel;
- Süsteem peab saama päikeseinverterilt andmed, salvestama need andmebaasi ja kuvama veebilehel;
- Süsteemi kasutaja peab saama iseseisvalt seadistada elektritarbimise parameetreid;
- Süsteemi kasutajal peab olema võimalik valida endale sobiv alam- ja maksimumhinna piir¹, mille alusel lülitatakse kasutatava elektriseadme voolutarve;
- Süsteemi kasutaja peab saama valida endale sobiva maksimaalse temperatuuri;
- Kui temperatuur on üle temperatuuripiirangu, lülitub seade välja;
- Kui hetke börsihind on kõrgem maksimaalsest hinnapiirist, ilmub hoiatus boileri väljalülitamiseks;
- Süsteem peab võrdlema mõõdetud parameetreid määratud parameetritega ja selle põhjal seadme ümber lülitama;

¹ börsihinna arvuline parameeter, ühikuga [senti / kWh]

- Kui temperatuur on üle temperatuuripiirangu, kuvatakse teade seadme väljalülitamiseks;
- Süsteemi seadete muutmisel kuvatakse teade õnnestunud muudatuse kohta;
- Süsteem peab veebilehel kuvama kasutaja jaoks mõõdetud parameetreid ja nende põhjal tehtud ümberlülitusotsuseid;
- Kui elektri börsihinna väärtus ei ole väliste tegurite tõttu saadaval, lülitab süsteem seadmeid ümber seatud maksimumtemperatuuri alusel.

Allpool on loetletud mittefunktsionaalsed nõuded:

- Kui süsteem on välja lülitatud, peavad elektriseadmed töötama tavarežiimis;
- Süsteemi saab kasutada ainult internetiühendusega;
- Süsteem jääb Interneti-ühenduse katkemisel tööle ja ootab ühenduse taastumist;
- Süsteem on mõeldud võimsatele seadmetele, mis sisse- ja väljalülitamisel ei kahjusta inimest igapäevaelus;

3.2 Maksimaalse efektiivsuse saavutamine

Esiteks on tarvis analüüsida, kuidas maksimeerida prototüübi efektiivsust, mis optimeerib energiaarveid, võttes samal ajal arvesse kasutaja vajadusi ja soove.

Lahenduse väljatöötamisel on oluline arvestada järgmiste teguritega:

- Kasutaja vajadused ja mugavus;
- Elektri börsihind;
- Päikeseenergia.

Kasutusaja skaalad (lühendatult TOU) kõiguvad tipp tundide alusel. Need on kellaajad, mil elektritarbimine on suurim. Kasutusaja skaalad varieeruvad sõltuvalt aastaajast ja piirkonnast. Elektrifirmad maksavad elektri eest rohkem tipp tundidel ja väljaspool tipp tundi vähem [5]. Need skaalad eksisteerivad arusaadavatel põhjustel. Kui elektrivõrgule avaldatakse suuremat survet, kulub nõudluse rahuldamiseks täiendava elektrienergia tootmisele rohkem ressursi. Kõrgemad määrad motiveerivad tarbijaid teoreetiliselt vähendama oma kasutamist tippaegadel, vähendades seeläbi survet võrgule.

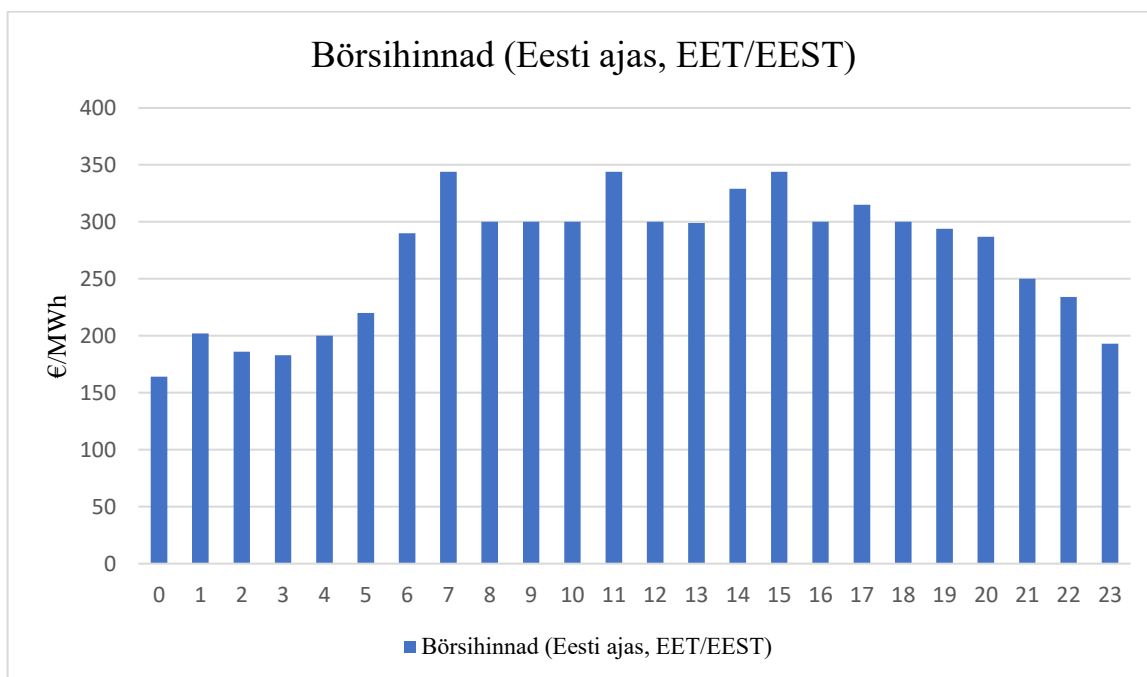
Enamik elektriettevõtteid, kes kasutavad kasutusaja tasu skaalat, jagavad päeva tipp-, öla- ja off-peak segmentideks. Näiteks võib suvise tööpäeva jagada järgmiselt [6]:

- Tipp: 13.00 kuni 18.00
- Õlg: 11.00–13.00 ja 18.00–20.00
- Väljaspool tipptundi: Ülejäänud päev

Talviste tööpäevade TOU ajakava võib olla järgmine:

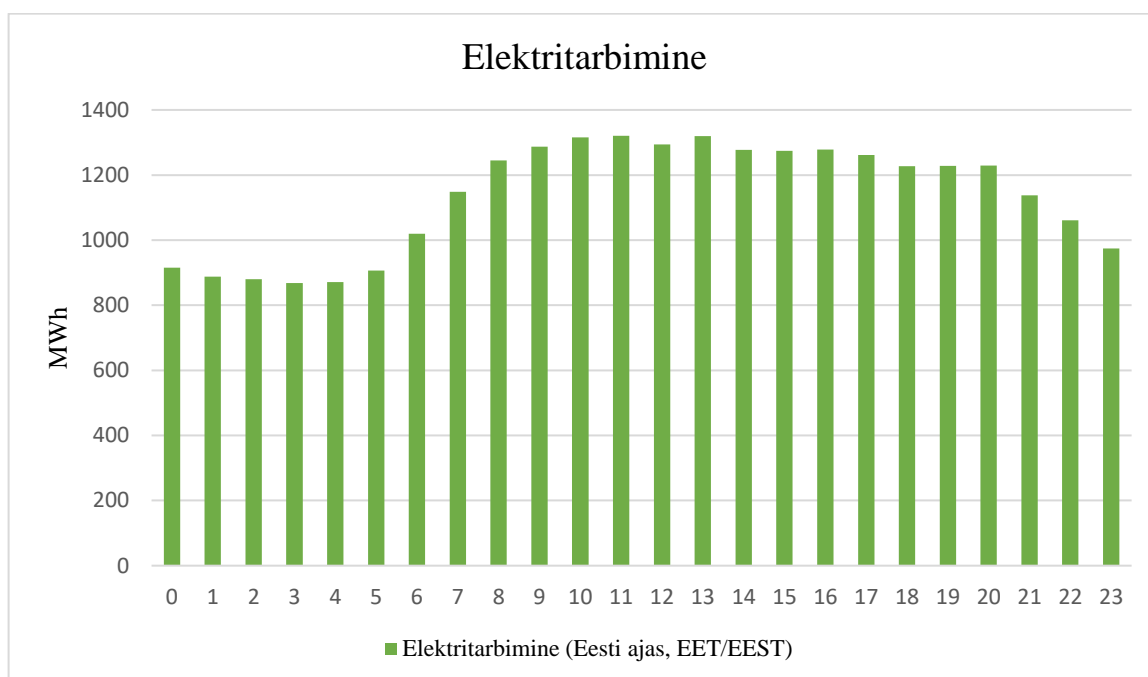
- Tipp: 6.00 kuni 9.00
- Õlg: 9.00–12.00, 17.00–20.00
- Väljaspool tipptundi: Ülejäänud päev

Joonisel 1 on kujutatud elektri börsihindade graafik 22. novembril 2022 [7].



Joonis 1. Börsihinnad Eestis 22.11.2022

Joonisel 2 on 22. novembri 2022 elektritarbimise graafik [7].



Joonis 2. Elektritarbimine Eestis 22.11.2022

Seetõttu on süsteemi efektiivseks toimimiseks TOU, tarbimisstatistika ja börsihindade põhjal vaja luua algoritm, mis võimaldaks vältida elektritarbimist kõige kallimatel ja suure nõudlusega tundidel ning leida kasutajale soodsaimad hinnad tema soovide alusel.

Lisaks peaks kasutajal olema võimalik iseseisvalt analüüsida hinnagraafikut ja valida tunnid, mil ta soovib elektriseadmeid kasutada. Neil valitud kellaegadeni peab süsteem leidma tarbimiseks soodsaima aja. Näitena töötatakse välja lahendus boileri jaoks ja tarbimistundide arv arvutatakse spetsiaalse valemiga [8]. Valem (1) esitab boileris vee lõpptemperatuurini soojendamiseks vajaminevate tundide arvu arvutust.

$$t^1 = 4.2 \cdot L^2 \cdot (T2^3 - T1^4) / 3600 / \text{Kütteelemendi võimsus}^5 \quad (1)$$

¹ Aeg tarbimiseks tundides

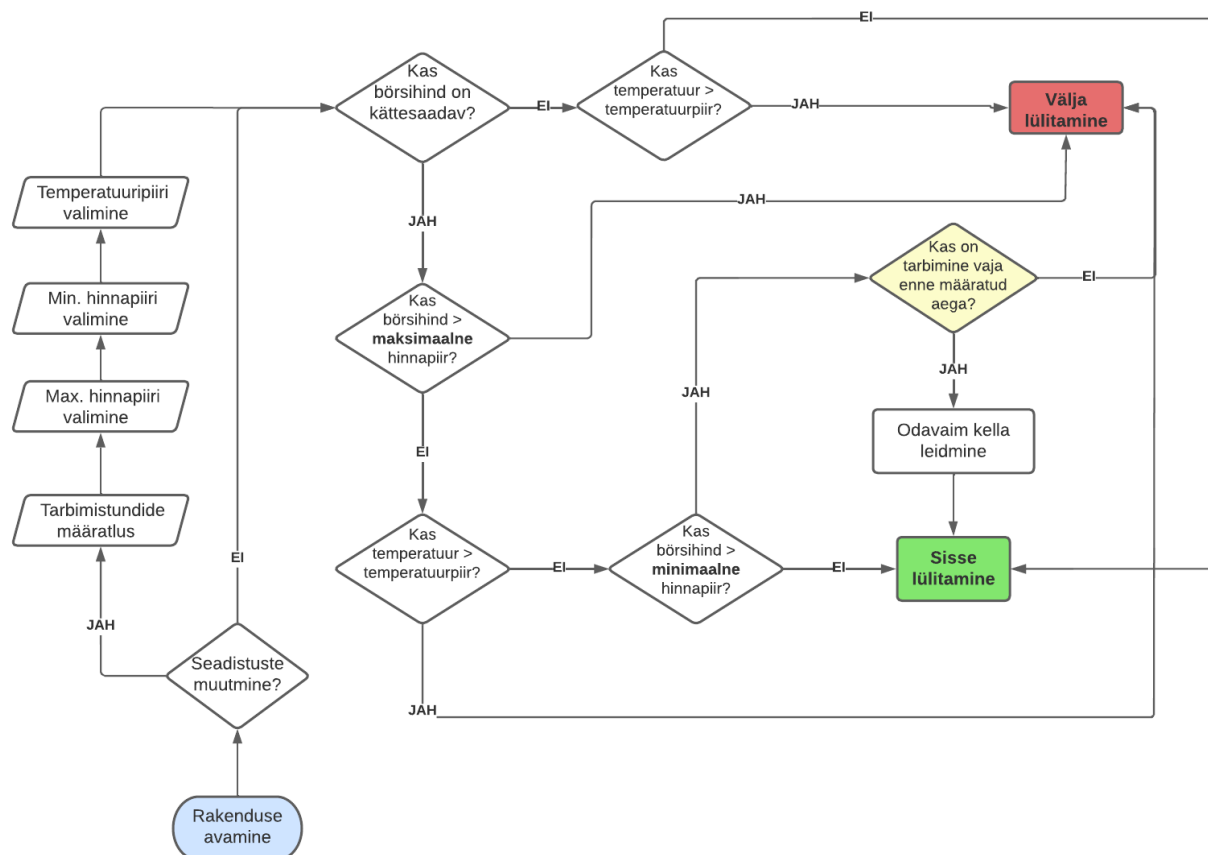
² Boileri maht liitrites

³ Lõplik veetemperatuur Celsiuse kraadides

⁴ Vee algtemperatuur Celsiuse kraadides

⁵ Boileri võimsus, ühikuga [kW]

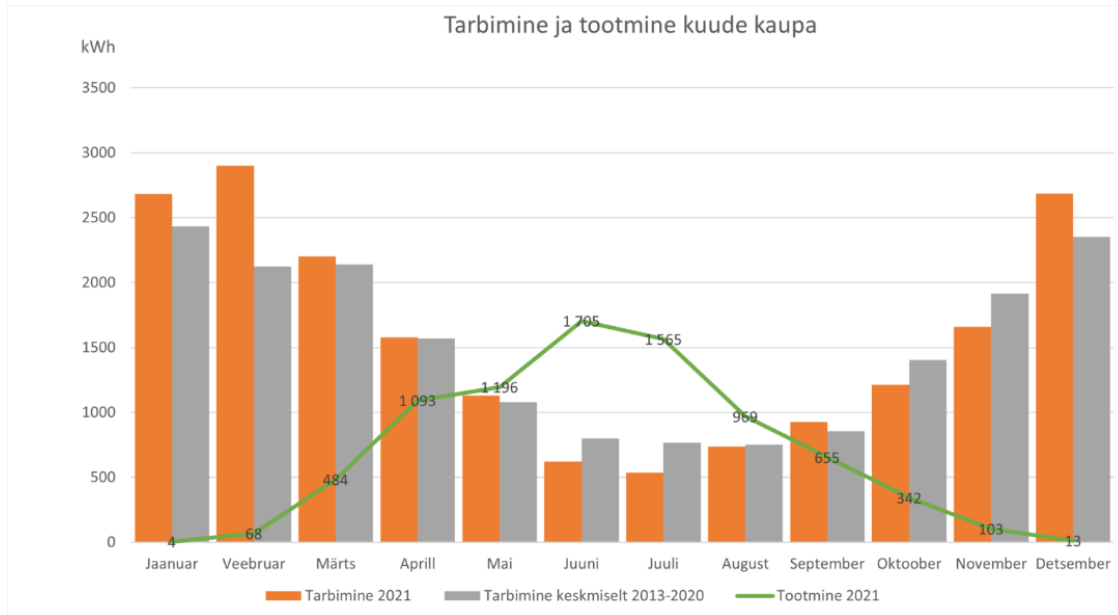
Joonis 3 esitab süsteemi tööalgoritmi plokkskeemi automaatrežiimis. Esiteks määrab kasutaja seaded vastavalt enda soovidele. Seejärel otsib süsteem seadistuste alusel kindla aja jooksul elektritarbimise odavaimad tunnid. Kui hind ületab lubatud maksimumi, ei lülita süsteem seadet automaatselt sisse ja seda saab teha käsitsi või teistsuguse maksimumhinna limiidi seadmisega. Kui hind jääb alla miinimumhinna piiri, lülitab süsteem seadme alati sisse.



Joonis 3. Süsteemi plokkskeem

3.2.1 Päikeseenergia

Joonis 4 näitab igakuist elektrienergia tootmist ja tarbimist. Lisaks on võrdluseks hallide tulpade kujul näha keskmine igakuine tarbimine aastatel 2013 kuni 2020. Nagu näha, on tarbimine aastate lõikes suhteliselt sama. Suured erinevused on seletatavad erineva küttevajadusega sõltuvalt ilmast [9].



Joonis 4. Tarbimine ja tootmine kuude kaupa

Nagu graafik näitab, siis üldiselt suudab päikeseenergia tootmine suvel tarbimist katta. Talvel on olukord vastupidine – tarbimine on väga suur ja päikeseenergia tootmine suhteliselt väike. Seetõttu töötab süsteem suvel ja talvel mõnevõrra erineva efektiivsusega. Talvel tuleb otsida odavaimat elektri ostmise võimalust ning suvel on mõistlik energia tarbimist võimalikult suurel määral katta päikeseenergiaga ja müüa ülejääv osa kõige kallima börsihinnaga.

3.3 Loodava lahenduse tehnoloogia valik

Punktis 2.2 kirjeldatud prototüübi loomiseks on tarvis valida õiged töövahendid. Lahenduse väljatöötamiseks on vaja järgnevaid vahendeid: päikesepaneelid, päikesepaneelide võrguinverter, releemoodul, temperatuuriandur, programmikood. Veebirakendust arendatakse päris päikeseelektrijaama jaoks, kus kogu vajalik riistvara on juba olemas. Sellest lähtuvalt selgitatakse ainult tarkvara jaoks mõeldud vahendite valikut.

3.3.1 Kasutatav riistvara

Päikeseelektrijaamas, millele lahendust välja töötatakse, on kokku 28 päikesepaneeli koguvõimsusega ligikaudu 7,3 kW:

- 24 paneeli võimsusega 285 vatti;

- 2 paneeli võimsusega 140 vatti;
- 2 paneeli võimsusega 100 vatti.

Võrku ühendatud päikeseenergiasüsteemi jaoks on vaja võrku paigaldatud inverterit, et muuta päikese poolt toodetud alalisvool 230-voldise vahelduvvooluks. Võrguinverter peab vastama jaotusvõrgu nõuetele. See peab andma sama sinusoidse vahelduvpinge, mis on võrgus saadaval, ja lisaks peab sagedus olema võrguga sünkroniseeritud [10].

Võrguinverter võib olla kolmefaasiline või ühefaasiline. Kolmefaasiline inverter on ühendatud maja elektrivõrgu kõigi faasidega ning suudab kompenseerida kõigis neis elektrienergia tarbimist ning ülejääva elektrienergia võrku suunata läbi mis tahes faasi. Kolmefaasilise võrguinverteri kasutamisel on võimalik jaotada koormus ühtlaselt kõigi kolme faasi vahel, mis võimaldab ära kasutada suurema osa päikeseelektrist [10].

Lahendust arendatakse päikeseelektrijaama jaoks, mis kasutab kolmefaasilist võrguinverterit Fronius Symo 6.0-3-M 6 kW (Joonis 14) [11]. Võrguinverteri omadused [12]:

- Puhas siinuslaine väljund;
- Maksimaalne efektiivsus 98,0%;
- Paigaldamine sise- ja välistingimustes;
- Väline võrguühendus;
- Võimsuse piiramine ülekoormuse korral;
- Vastupidise polaarsuse kaitse;
- Standardliides Interneti-ühenduse loomiseks WLAN-i või Etherneti kaudu.

Elektriseadmete ümberlülitamiseks kasutatakse Sonoff TH-16 (Joonis 15) [13] — sisseehitatud temperatuuri- ja niiskusanduriga juhtmevaba lülitit, mis võimaldab kaugjuhtida ja seadistada temperatuuri ja niiskuse vahemikku, samuti sisse lülitada ühendatud seadmetest välja. Algne Sonoffi püsivara võimaldab teil juhtida seadmeid ainult Interneti kaudu läbi rakenduse eWeLink. See muudab aga olemasolevate koduautomaatika serveritega integreerimise või lihtsalt seadmete lokaalse haldamise ilma Interneti-juurdepääsuta keeruliseks. Sel põhjusel on püsivara asendatud Tasmotaga, kuna see võimaldab hõlpsasti integreerida olemasolevate koduautomaatika serveritega ja võimaldab kasutajatel juhtida seadmeid kohapeal või eemalt, kasutades erinevaid

meetodeid, nagu veebiliides, HTTP-päringud ja MQTT [14].

3.3.2 Programmeerimiskeele valik

Rakenduse arendamiseks tööriistu valides on tarvis analüüsida populaarsemaid programmeerimiskeeli, hinnata oma oskusi ja valida eesmärgi saavutamiseks sobivaim.

Java on laialdaselt kasutatav programmeerimiskeel veebirakenduste arendamiseks. See on olnud arendajate seas populaarne valik juba üle kahe aastakümne. Tänapäeval on kasutusel miljoneid Java-rakendusi. Java on mitmeplatvormiline, objektorienteeritud ja võrgukeskne keel, mida saab kasutada platvormina. See on kiire, turvaline ja usaldusväärne programmeerimiskeel, mis võimaldab arendada kõike alates mobiilirakendustest ja ettevõtetarkvarast kuni suurte andmerakenduste ja serveripoolsete tehnoloogiateni välja [15].

Tabel 1. Java programmeerimiskeele plussid ja miinused

Plussid [16]	Miinused [16]
Java loodi nii, et seda oleks lihtne kasutada, kirjutada, kompileerida, siluda ja õppida.	See on oluliselt aeglasem ja mälumahukam kui natiivselt kompileeritud keeled, nagu C või C++.
Java programmeerimine keskendub objektide loomisele, objektidega manipuleerimisele ja objektide koos töötamisele. Seega aitab Java luua modulaarseid programme ja korduvkasutatavat koodi.	Graafilise kasutajaliidese rakenduste vaikimisi välimus ja tunnetus, mis on kirjutatud Java-vormingus swing toolkit abil, erineb oluliselt omarakendustest.
Java on esimene programmeerimiskeel, mis sisaldab disaini lahutamatu osana turvalisust.	
Java pakub rakenduste arendamiseks erinevaid API-sid.	

Python on programmeerimiskeel, mida kasutatakse laialdaselt veebirakendustes, tarkvaraarenduses, andmeteaduses ja masinõppes. Arendajad kasutavad Pythonit, kuna see on tõhus ja kergesti õpitav ning võib töötada paljudel erinevatel platvormidel. Pythoni tarkvara on tasuta allalaaditav, integreerub hästi igat tüüpi süsteemidega ja suurendab arenduskiirust [17].

Tabel 2. Python programmeerimiskeele plussid ja miinused

Plussid [16]	Miinused [16]
See on avatud lähtekoodiga ja tasuta.	Koodi aeglane täitmine.
Lihne lugeda ja lihtne õppida.	Pythoni andmebaasi juurdepääsukiht on primitiivne ja vähearenenud.
See pakub suuri standardseid teeke.	Pythoni programmeerimiskeel kasutab palju mälu.
Sellel on võimsad juhtimisvõimalused, kuna seda kutsutakse otse C, C++ kaudu või Java Jythoni kaudu.	

PHP on avatud lähtekoodiga tasuta serveripoolne skriptikeel, mida kasutatakse laialdaselt veebiarenduses. See käivitatakse serveris, mis seejärel transleeritakse HTML-koodi abil kliendi poolele. PHP on saadaval kõigis suuremates operatsioonisüsteemides, nagu Linux, Microsoft Windows ja macOS. Enamik veebiservereid, sealhulgas Apache ja IIS, toetavad ka PHP-d [18].

Tabel 3. PHP programmeerimiskeele plussid ja miinused

Plussid [16]	Miinused [16]
PHP kõige olulisem eelis on see, et see on avatud lähtekoodiga ja tasuta, seda saab alla laadida kõikjalt ja see on hõlpsasti kättesaadav veebirakenduste arendamiseks.	See pole nii turvaline, kuna see on avatud lähtekoodiga, mis on kergesti kättesaadav.
PHP raamistike pakutavad tööriistad, funktsioonid ja koodilõigud aitavad arendajatel kiirendada kohandatud veebirakenduste arendamist.	PHP-l on vigade ja hoiatuste leidmiseks suhteliselt vähe silumistööriistu.
PHP-raamistikud lihtsustavad veebirakenduste arendamist ja hooldust, toetades mudelivaate kontrolleri arhitektuuri, et jagada veebirakendus mudeliteks, vaadetekks ja kontrolleriteks.	PHP raamistiku ja tööriistade rohkemate funktsioonide kasutamine viib veebirakenduste halva jõudluseni.
Sellel on võimas raamatukogu tugi, et kasutada andmete esitamiseks erinevaid funktsioonimoduleid.	

C# on kaasaegne üldotstarbeline objektorienteeritud programmeerimiskeel, mida saab kasutada paljude ülesannete jaoks, mis hõlmavad paljusid elukutseid. C# kasutatakse

peamiselt Windows .NET keskkonnas, kuigi seda saab rakendada ka avatud lähtekoodiga platvormile [19].

Tabel 4. C# programmeerimiskeele plussid ja miinused

Plussid [16]	Miinused [16]
C#-i on lihtne õppida ja see toetab kaasaegseid funktsioone.	C# põhineb täielikult Microsoft .Neti raamistikul. Rakenduse käivitamiseks tuleb arvutisse installida .NET-raamistik.
C# keel on objektorienteeritud programmeerimiskeel, nii et oops muudab arendamise ja hoolduse protseduurile orienteeritud programmeerimiskeelega võrreldes lihtsamaks.	
C# on väga kiire. Selle kompileerimise ja täitmise aeg on kiire.	
See toetab automaatset mäluksortimist.	

Arvestades iga programmeerimiskeele eeliseid ja puudusi, on vaja hinnata autori isiklikku kogemust ja välja selgitada, millises programmeerimiskeeles oleks autoril eesmärki kõige mugavam realiseerida. Tabelis 5 on hinnatud autori oskusi ja raskusi programmeerimiskeelte õppimisel.

Tabel 5. Programmeerimiskeelte võrdlus

Programmeerimiskeel	Oskus	Õppimiskeerukus
Java	Hea	Lihtne
C#	Rahuldav	Keskmine
PHP	Hea	Väga lihtne
Python	Hea	Lihtne

PHP on tuntud kui üldotstarbeline programmeerimiskeel. Seda kasutatakse serveripoolse skriptikeelena, mis on mõeldud peamiselt veebisaitide arendamiseks. Võrreldes teiste programmeerimiskeeltega on PHP-d lihtne õppida. Peale selle, et algajatele on saadaval ulatuslik dokumentatsioon, on PHP süntaks elementaarne. See on paindlik ja kombineerub hästi teiste veebiarendustehnoloogiatega ning on ka kiirem kui teised programmeerimiskeeled, näiteks Python [20].

Eesmärgi saavutamiseks valis autor PHP keele, kuna see programmeerimiskeel sobib suurepäraselt eesmärgi saavutamiseks ning lisaks võimaldab see autoril end täiendada PHP veebiarenduses. Tulevikus plaanib autor selles programmeerimiskeeles aktiivselt oma oskusi veebiarenduse vallas arendada.

Töövõimaluste osas on PHP nõutav avatud lähtekoodiga programmeerimiskeel, hoolimata sellest, et see on üks vanimaid selles valdkonnas. PHP programmeerijad saavad seda keelt kasutada nii käsurea skriptide kui ka serveripoolsete skriptide kirjutamiseks ja töölaarakenduste arendamiseks.

Kuna PHP on avatud lähtekoodiga platvorm, on sellel tohtu kogukond, kes aitab igas keerulises olukorras ja vastab kõigile küsimustele. Internetis on selle programmeerimiskeele jaoks palju juhendeid ja õpetusi, mis teeb PHP õppimise lihtsaks ja kiireks.

3.3.3 Teenusepoolse raamistiku valik

Raamistik on tarkvarakeskkond, mis lihtsustab ja kiirendab tarkvara loomist. Raamistik määratleb rakenduse arhitektuuri ja pakub selle komponentide vahelist suhtlust. See sisaldab erinevaid teekes ja kasutab neid rakenduse arhitektuuri koostamiseks. Raamistikuga töötamisel toimub juhtimise ümberpööramine [21].

Raamistikud pakuvad arendajatele palju võimalusi, sealhulgas [21]:

- kõrvaldada vajadus koodi iga kord uuesti kirjutada;
- kiirendada arendusprotsessi;
- lihtsustada rakendusega töötamist;
- keskenduda rakendusepõhisele loogikale;
- eraldada loogika esitlusest;
- rakendada veebitoote põhifunktsioone.

Arvestades valitud programmeerimiskeelt, on vaja rakenduse arendamiseks valida sobiv raamistik. Järgmised on mõned parimad tänapäeval kasutatavad PHP raamistikud [22]:

- Laravel;
- Symfony;

- CodeIgniter;
- Zend Framework / Laminas Project;
- Yii (Framework);
- CakePHP;
- Slim;
- Phalcon;
- FuelPHP;
- Fat-Free Framework.

JetBrains'i andmetel eelistab 67% programmeerijatest Laravelit ja 6% Yii-d. Laravel on üks paindlikumaid ja kiiremaid raamistikke, mis pakub ulatuslikku arendustööriistade komplekti tõhusaks veebilehe arendamiseks [23].

Kõigist tuntud PHP raamistikest on Laravel ja Yii kaks avatud lähtekoodiga PHP raamistikku, mis töötavad Model View Controller (MVC) raamistikus. Mõlemad keskkonnad kiirendavad arendusprotsessi vaatamata sellele, et neil on arendusprotsessides vähe erinevusi. Lisas 4 analüüsitakse Yii ja Laraveli peamisi erinevusi ja otsustatakse, milline raamistik oma eesmärgi saavutamiseks valida [23].

Yii põhijooned, mida veebirakenduste arendamiseks PHP raamistiku valimisel arvestada [23]:

- **Hästi kirjutatud dokumentatsioon:** Yii pakub veebiarendajatele suurepärase tuge võimaldades ligipääsu arvukatele veebiõpetustele. Samuti on selles raamistikus arendajatele lai kogukonna toetus;
- **Projekti seadistamine:** Laraveli ja Yii projekti seadistuste võrdlemisel saavutab Yii rohkem punkte kui Laravel. Kuigi mõlemad need raamistikud järgivad lihtsat installiprotsessi;
- **Piiratud arendusaeg:** projektidel kulub korduvate ülesannete täitmiseks teatud aeg. Yii pakub mitmeid tööriistu, mis aitavad nendele asjadele vähem aega kulutada. Kokkuvõttes aitab see lühendada arendusele kuluvat aega;
- **Suurepärase turvalisus:** turvafunktsioon on üks peamisi põhjuseid, miks valida Yii muude raamistike asemel. See kiire ja tugev raamistik aitab luua turvalisema veebirakenduse ja hoiab ära rakenduse küberrünnakute eest;

- **Lihne seadistada:** Yii installimine on lihtne, kuna *composer* juhib installiprotsessi. See aitab aega kokku hoida, pakkudes kiiremaid installi- ja konfigureerimisprotseduure;
- **Laiendatavad ja kaasaegsed tehnoloogiad:** Yii on tuntud oma võime poolest kohanduda kasutaja nõudmistele ja pakub raamistiku iga komponendi piiramatut laiendatavust. Isegi see haldab OOP-raamistikku ja kasutab mõnda PHP keerukat funktsiooni, nagu anonüümsed funktsioonid, SPL-klassid ja liidesed ning staatiline sidumine.

Laravel põhijooned, mida veebirakenduste arendamiseks PHP raamistiku valimisel arvestada [23]:

- **Mallid:** Laravel on populaarne oma sisseehitatud kergete mallide poolest, mida kasutatakse dünaamilise sisukülvi kaudu intuiitivsete paigutuste loomiseks. Need mallid on loodud Laraveli arendajate arendusprotsessi lihtsustamiseks;
- **ORM (Object-Relational Mapping):** Laraveli teine omadus on ORM, mis võimaldab arendajatel hõlpsasti andmebaasipäringuid kirjutada. Laravel on õige valik, kui on vaja kiiremat veebiarendust;
- **Suurepärane üksuse testimiseks:** testimine mängib olulist rolli enne rakenduse juurutamist, et tagada selle vigade puudumine. Laravel on õige valik veebiarenduste üksuste testimiseks. See raamistik võib käivitada mitme üksuse teste, et tagada veebirakenduste veavabadus;
- **Eelinstallitud teigid ja modulaarne:** kõigi PHP raamistike hulgas on Laravelil eelinstallitud teigid. See pakub ka palju muid objektorienteeritud teeke.

Olles uurinud raamistike omadusi ja eeliseid, otsustas käesoleva lõputöö autor proovida töötada Yii raamistikuga. Yii2 funktsioonide hulgas [24]:

- Täielik tugi jQueryle, esiotsa teegile, mis võimaldab suhelda lehe HTML-märgistuse ja selle JavaScripti koodi vahel;
- Tugi Bootstrapile, raamistikule, mis võimaldab kiiresti luua mallide veebiliidese elemente. Need võivad olla vormid, karussellid jne;
- Kaitse veebisaitide SQL-i süstide ja XSS-i rünnakute eest;
- Oskus töötada andmebaasiga ja teostada selle migratsiooni - sisu versioonikontroll;

- Töötamine mallimootoriga Smarty ja Twig – programmid, mis muudavad PHP koodi HTML-leheks;
- AJAX-tehnoloogia tugi dünaamilise sisu laadimise jaoks ja programmeerimisstiil REST, mis kirjeldab komponentide koostoime funktsioone;
- Mallikoodi genereerimine sisseehitatud Gii tööriista abil;
- Silumine natiivse siluriga Yii2-debug;
- Automatiseeritud testimine, vigade käsitlemine ja vormide valideerimine;

3.3.4 Kliendipoolne tehnoloogia

Veebisaidi seda osa, millega kasutaja vahetult suhtleb, nimetatakse kasutajaliideseks. Seda nimetatakse ka rakenduse kliendipoolseks. See sisaldab kõike, mida kasutajad vahetult kogevad: tekstivärvid ja stiilid, pildid, graafika ja tabelid, nupud, värvid ja navigeerimismenüüd.

Allpool on kõige populaarsemad tööriistad ja tehnoloogiad esiotsa arendamiseks:

- **AngularJS** – seda peetakse üheks olulisemaks esiotsa veebiarendustööriistaks. Lisaks on see avatud lähtekoodiga veebirakenduse raamistik, mis aitab HTML-i süntaksit hõlpsalt laiendada. Üldiselt pakub AngularJS loetavat, juurdepääsetavat ja väljendusrikast keskkonda, mis lihtsustab esiotsa arendusprotsessi [25];
- **jQuery** – üks populaarsemaid JavaScripti teekes, mida kasutatakse laialdaselt esiotsa arendamiseks. jQuery teek aitab hõlpsasti juurde pääseda mis tahes DOM-i elemendile, pääseda juurde DOM-i elemendi atribuutidele ja sisule ning nendega manipuleerida. Samuti pakub jQuery teek mugavat API-d AJAX-iga töötamiseks [25];
- **Npm** – Node.js-is sisalduv paketihooldur. NPM aitab leida tuhandete arendajate pakette ja taaskasutada koodi ning neid võimsal viisil koguda. NPM-i hallatakse käsurealt ning see on lihtne ja kiire viis veebirakenduste loomiseks. NPM-skriptidega saab teha palju asju, sealhulgas uute pakettide tõmbamine, vanade värskendamine ja palju muud [25];
- **Bootstrap** – üks populaarsemaid esiotsa veebiarendustööriistu, mida kasutatakse laialdaselt kogu maailmas. Bootstrap on mainekas raamistik, mida arendajad kasutavad kliendipoolse kodeerimise loomiseks ja ahvatlevate esiosade loomiseks. Kasutajaliidese raamistikud nagu Bootstrap pakuvad tavaliste

arenduselementide abstraktsiooni korduvkasutatavateks mooduliteks, mis tähendab, et arendajad saavad rakenduste elemente kiiresti ja lihtsalt luua. Lisaks võib see lühendada rakenduste loomiseks kuluvat aega ja koodi [25];

- **VueJS** – kerge JavaScripti raamistik reageerivate veebikasutajaliideste loomiseks. Vue laiendab standardset HTML-i ja CSS-i, et luua võimsate tööriistade komplekt interaktiivsete veebirakenduste esiosa loomiseks [26].

Lähtudes tehnilistest nõuetest ja loodava lahenduse eesmärgist valis autor frontendi arendamiseks Bootstrapi ja JQuery. Kuna reaalajas on vaja kuvada elektri tarbimise, ostu ja müügi statistikat ning päikeseinverterilt saadud andmeid, on Ajaxi kasutamine vajalik.

Ajax võimaldab veebilehtedel JavaScriptiga serverisse tagaplaanil päringuid teha ja andmeid vastu võtta, segamata avatud lehe kuvamist ja olekut. Andmete saamiseks kasutatakse *XMLHttpRequest* objekti. JavaScript ja *XMLHttpRequest* võimaldavad teavet serveri ja brauseri vahel asünkroonselt vahetada, et vältida lehe uuestilaadimist [27].

3.3.5 Andmebaasi valik

Andmebaasihaldussüsteem (DBMS) on tarkvara- ja keeletööriistade komplekt, mis võimaldab luua andmebaase ja hallata andmeid. Teisisõnu, DBMS on programmide kogum, mis võimaldab andmebaase korraldada, juhtida ja hallata [28]. Enamik veebilehti ei saa ilma andmebaasita toimida, seega kasutatakse DBMS-i peaaegu kõikjal.

Järgnevalt on välja toodud mõned populaarsemad andmebaasid:

- **MySQL** – peetakse üheks levinumaks DBMS-iks. MySQL on avatud lähtekoodiga relatsiooniline DBMS, mille peamisteks eelisteks on kiirus ja paindlikkus, mida toetab suur hulk erinevat tüüpi tabeleid. Lisaks on see usaldusväärne tasuta süsteem, millel on lihtne liides ja võimalus sünkroonida teiste andmebaasidega. Need tegurid koos võimaldavad MySQL-i kasutada nii suurkorporatsioonidel kui ka väikeettevõtetel [28];
- **Microsoft SQL Server** – Microsofti poolt välja töötatud patenteeritud DBMS. Optimaalne kasutamiseks Windowsi perekonna operatsioonisüsteemides, kuid see võib töötada ka Linuxiga. Süsteem võimaldab sünkroonida teiste Microsofti tarkvaratoodetega ning pakub ka usaldusväärset andmekaitset ja lihtsat liidest,

kuid seda iseloomustab kõrge litsentsikulu ja suurenenud ressursikulu. Üldiselt säilitab see siiski oma populaarsuse, suures osas tänu sellele, et Microsofti tooteid kasutavad paljud ettevõtted [28];

- **PostgreSQL** – sisaldab arvukalt funktsioone, nagu pärimine ja funktsioonide laadimine, mis muudavad selle üheks parimaks saadaolevaks andmebaasiks. Neid funktsioone kasutatakse laiendatavuse, töökindluse ja andmete terviklikkuse suurendamiseks andmete töötlemisel. Sellel on erinevad andmetüübid ja töökindlad funktsioonikomplektid, mis võimaldavad ettevõtetel oma andmetöötluseesmärke saavutada [28];
- **MariaDB** – on populaarne relatsiooniandmebaasi haldussüsteem, mis ühildub MySQL-i protokolliga ja klientidega. MariaDB toetab veergude salvestamist ja sellel on massiliselt paralleelselt hajutatud andmearhitektuur. MariaDB-l on teie päringute arutamiseks ja lahendamiseks suur kasutajate kogukond [28];
- **Oracle** – tööstusharudes laialdaselt kasutatav relatsiooniandmebaasi haldussüsteem (RDBMS). See on üks populaarsemaid andmebaase, kuna seda peetakse kulusid optimeerivaks ja suure jõudlusega lahenduseks. See on ainus konvergeeritud ettevõtte relatsiooniandmebaas, mis on selles valdkonnas olemas. Oracle Database toetab andmebaasiga suhtlemiseks struktureeritud päringukeelt (SQL) [28].

MySQL on PHP arendajate esimene valik. SQL-keelt kasutava avatud lähtekoodiga relatsiooniandmebaasi haldussüsteemina (RDBMS) aitab MySQL andmebaas automatiseerida andmete otsimist ja pakub suurepärase tuge PHP MySQL veebirakenduste arendamisel [29]. MySQL automatiseerib kõige sagedasemad toimingud, mis on seotud unikaalse kasutajateabe hankimise ja salvestamisega esitatud kriteeriumide alusel. MySQL on avatud lähtekoodiga relatsioonilise andmebaasi haldussüsteem ja PHP serveripoolseks programmeerimiskeeleks. Kui mõlemad on kombineeritud, võib see pakkuda väga eksklusiivseid lahendusi [30]. Mõned PHP ja MySQL kasutamise peamised eelised on loetletud allpool [29]:

- Kulusäästlikud isikupärastatud PHP rakenduste lahendused;
- Täielik avatud lähtekoodiga platvorm;
- Värskendatakse sageli (need värskendused on saadaval ka tasuta);
- Erinevate lisandmoodulite ja pistikprogrammide saadavus;

- Suurepärase veebisaidi jõudlus;
- Suur hulk andmebaasi liideseid;
- PHP kontrollib erinevaid turvasemeid;
- Integreerimisvõimalus mitmesuguste muude avatud lähtekoodiga platvormidega;
- Integreerib teisi tarkvarakeeli nagu C, C++, JAVA/AJAX jne;
- HTML-koodi saab manustada PHP-koodi;
- Platvormidevaheline võime;
- Serveripoolne veebiprogrammeerimiskeel;
- Suur kasutajaskond.

Kõige eelneva põhjal otsustas autor kasutada nii MySQL-i andmebaasi kui ka phpMyAdminit – PHP-s kirjutatud rakendust, mis pakub brauseri kaudu täisväärtuslikku, sealhulgas kaugtööd MySQL-i andmebaasidega. See on kõige populaarsem MySQL-i andmebaasihaldusrakendus. Selle tarkvara abil saab luua, värskendada, muuta, kustutada, importida ja eksportida MySQL andmebaasi tabeleid [31].

3.4 Analüüsi kokkuvõte

Loodava prototüübi analüüsi käigus toodi välja rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded, ning analüüsiti, kuidas saavutada süsteemi maksimaalset efektiivsust kasutaja isiklike vajadusi ja soove eiramata. Samuti koostati süsteemi tööalgoritmi plokk skeem.

Pärast erinevate programmeerimiskeelte eeliste ja puuduste analüüsi ning autori isiklikku kogemust valiti lahenduse väljatöötamiseks PHP programmeerimiskeel, kuna sellel on lihtne ja arusaadav süntaks ning selle uurimiseks on palju dokumentatsiooni. Arenduseks valiti Yii raamistik. PHP raamistike seas paistab Yii silma väga hea jõudluse, turvalisuse ja suhtelise õppimise lihtsuse poolest. Esiotsa osaks valiti Bootstrap, JQuery ja Ajax, kuna need tehnoloogiad on juba Yii raamistikus kaasatud, sobivad kõige paremini interaktiivse kasutajaliidese loomiseks ja annavad võimaluse uuendada veebilehe üksikuid komponente ilma seda täielikult värskendamata.

Autor otsustas kasutada kõige populaarsemat MySQL andmebaasi, et salvestada seadistusi, andmeid elektritarbimise ja börsihindade kohta, andmeid päikeseinverterilt. Andmebaasi haldamiseks valiti mugav tööriist - phpMyAdmin.

4 Lahenduse loomine

Prototüüp on loodud virtuaalses VPS-serveris, kasutades Linux'i operatsioonisüsteemi. VPS on virtuaalse pühendatud serveri rentimise teenus. Operatsioonisüsteemi haldamise osas vastab see enamasti füüsilisele spetsiaalsele serverile. Eelkõige on olemas juurjuurdepääs, privaatsed IP-aadressid, pordid, filtreerimisreeglid ja marsruutimistabelid. Lisas 5 käsitletakse täpsemalt, kuidas klient, VPS-server ja Sonoffiga ühendatud koduseade omavahel suhtlevad.

Järgmisena installiti serverisse apache, kasutades käsku *sudo apt-get install apache2*. Composer on installeeritud käsuga *sudo apt install composer*. Composerit kasutatakse tavaliselt populaarsetel PHP raamistikel põhinevate uute projektide käivitamiseks. Kui Composer sai installitud, installiti Yii rakenduse mall *composer create-project --prefer-dist yiisoft/yii2-app-advanced yii-application* käsu abil.

Failis `frontend/web/index.php` luuakse Yii rakenduse objekt, määratakse konfiguratsioonid ja käivitatakse rakenduse põhifunktsioonid. Kui sisestusskript loob rakenduse, laadib see konfiguratsiooni ja rakendab seda järgmiselt (Joonis 5):

```
<?php

defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');

require __DIR__ . '/../vendor/autoload.php';
require __DIR__ . '/../vendor/yiisoft/yii2/Yii.php';
require __DIR__ . '/../common/config/bootstrap.php';
require __DIR__ . '/../config/bootstrap.php';

$config = yii\helpers\ArrayHelper::merge(
    require __DIR__ . '/../common/config/main.php',
    require __DIR__ . '/../common/config/main-local.php',
    require __DIR__ . '/../config/main.php',
    require __DIR__ . '/../config/main-local.php'
);

(new yii\web\Application($config))->run();
```

Joonis 5. Yii rakendusobjekti loomise programmikoodi näide

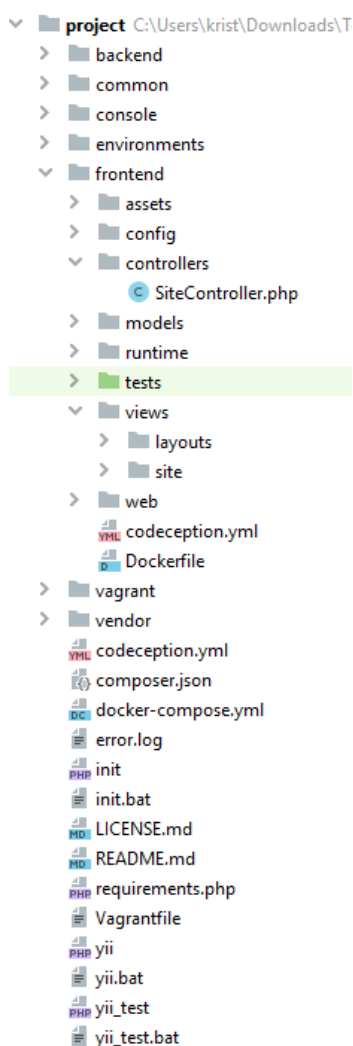
Kaust *vendor* määratleb Composer'i juhtkataloogi. See sisaldab kõiki rakenduse kasutatavaid kolmandate osapoolte teeke, sealhulgas raamistikku.

Mudelid, vaated ja kontrollid on loodud Gii tööriista abil. Gii on laiendus, mis pakub veebipõhist koodigeneraatorit mudelite, vormide, moodulite, CRUD-i jne loomiseks. Kaust *frontend/models* sisaldab mudeleid. Kontrollid asuvad failis *frontend/controllers/SiteController.php* ja on lingiks mudeli ja vaate vahel. Kontrollid tegeleb nii nende interaktsioonidega kui ka sisendiga. Vaade on veebirakenduse visuaalne osa, mida saidi kasutaja näeb, ehk teisisõnu need on html-paigutuse failid, mis on ühendatud kontrolleri meetoditega. Fail *frontend/views/layouts/main.php* sisaldab *header* ja *footer* malle ning ka *Vaade*. Nagu on näidatud joonisel 6, sisaldab muutuja *\$content* vaate sisu või selle faili renderdamise tulemust, mille nimi anti kontrolleri meetodile:

```
<main role="main" class="flex-shrink-0">
  <div class="container">
    <?= Breadcrumbs::widget([
      'links' => isset($this->params['breadcrumbs']) ? $this-
>params['breadcrumbs'] : [],
    ]) ?>
    <?= Alert::widget() ?>
    <?= $content ?>
  </div>
</main>
```

Joonis 6. Muutuja, mis sisaldab Vaade sisu programmikoodis

Joonisel 7 on esitatud projekti struktuur:



Joonis 7. Projekti struktuur

4.1 Börsihindade saamine

Iga 10 sekundi järel saadab Fronius serverisse faili *fronius_data.php*. Fail sisaldab *cURL*, mis käivitab kontrolleri, milles lisaks päikeseinverterilt saadud andmete dekrüpteerimisele käivitatakse ka meetod *getTodayData()*. Joonisel 8 näidatud funktsiooni sees on päring saidile, kus hinnad on JSON formaadis ja kui serveri andmebaasis selle päeva kohta andmeid pole, siis kirjutatakse sinna uus rida.

```

public function getTodayData()
{
    $timestamp = new \DateTime(date("Y-m-d", strtotime("-1 day")), new
    \DateTimeZone("Europe/Tallinn"));
    $start=$timestamp->format('Y-m-d').'T'.$timestamp->format('00:00:00').'999Z';
    $timestamp->modify('+1 day');
    $end=$timestamp->format('Y-m-d') . 'T' . $timestamp->format('23:59:59').'999Z';
    $url='https://dashboard.elering.ee/api/nps/price?start='.$start.'&end='.$end;

    $spotData = SpotData::find()->where(['pvm' => date('Y-m-d')])->one();
    if(!isset($spotData->id))
    {
        $getData= file_get_contents($url);

        $spotData = new SpotData;
        $spotData->pvm = date('Y-m-d');
        $spotData->json = $getData;
        if(!$spotData->save())
        {
            print_r($spotData->getErrors());
            exit;
        }
    }

    return $spotData;
}

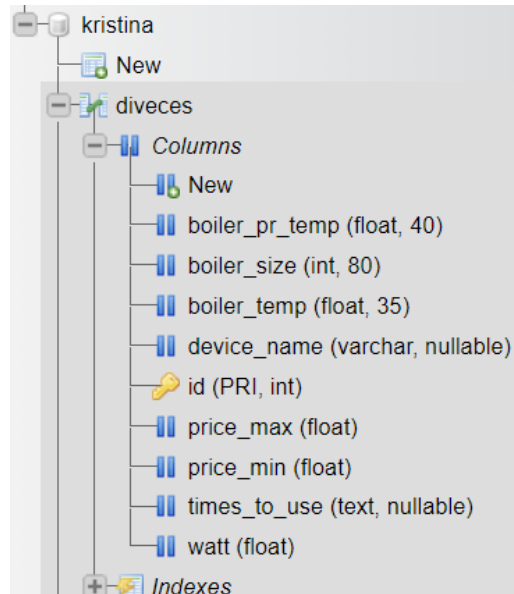
```

Joonis 8. Börsihidade saamine programmikoodis

4.2 Andmebaas

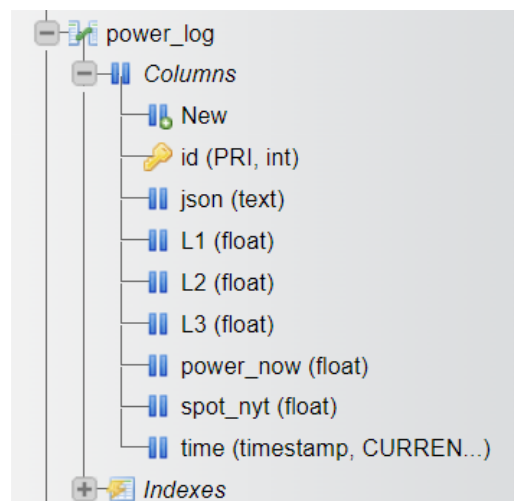
MySQL installiti `sudo apt-get install mysql-server -y` käsu abil. Järgmisena installiti phpMyAdmin käsuga `sudo apt-get install phpmyadmin -y`. Fail `common/config/main.local.php` sisaldab andmebaasiühendust.

Seadme andmed, antud juhul boileri andmed, salvestatakse joonisel 9 näidatud tabelisse. Sellesse salvestatakse järgmised väärtused, mille määrab veebilehe kasutaja: seadme nimi, hetketemperatuur, soovitud temperatuur, võimsus, maksimaalsed ja minimaalsed hinnapiirangud ning seadme kasutamise aeg.



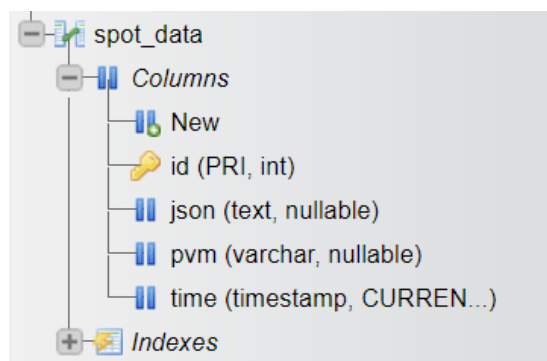
Joonis 9. Seadme konfiguratsiooni tabel

Tabelis *power_log* (Joonis 10) salvestatakse Froniuse päikeseinverterilt pärinevad andmed. Nagu kirjeldatud punktis 4.1, saadab Fronius iga 10 sekundi järel serverisse faili *fronius_data.php*. Lisa 6 näitab, et see fail kirjutab andmed tekstifaili *fronius_data.txt* ja see sisaldab ka cURL-i, mis käivitab meetodi *actionFroniusdata()* failis *frontend/controllers/SiteController.php*, kus tekstifaili andmed dekrüpteeritakse ja seejärel kirjutatakse andmebaasi.



Joonis 10. Päikeseinverterilt tulevate andmete tabel

Börsihinnad salvestatakse tabelis *spot_data* (Joonis 11).



Joonis 11. Börsihindade tabel

4.3 Valminud lahenduse tulemused

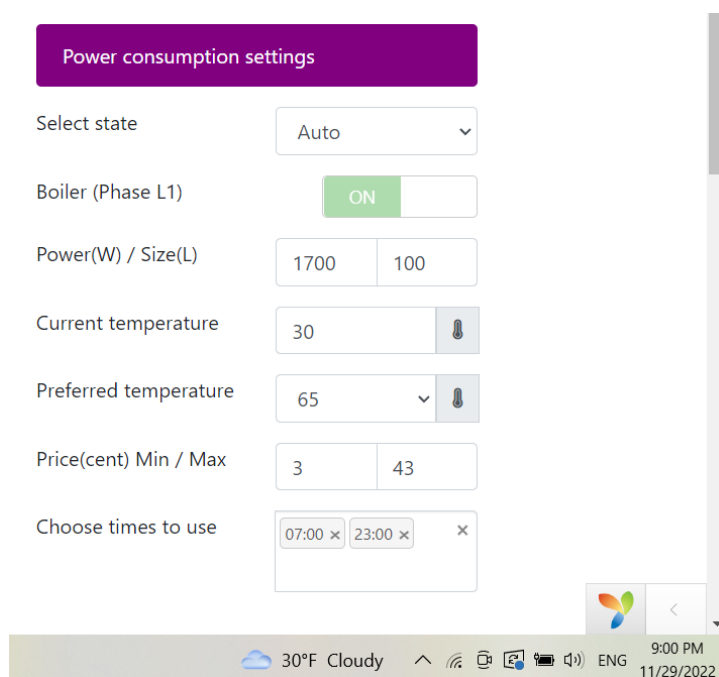
Süsteemi tulemuse kontrollimiseks on vaja määrata seadeid ja jälgida, kas boiler lülitub teatud ajal sisse ja välja. Näiteks on kasutaja valinud boileri kasutamiseks kellaajad 07.00 ja 23.00. Seejärel skaneerib süsteem kellaega 00.00-07.00 ja 07.00-23.00 ning leiab madalaima hinnaga tunnid, millal boiler sisse lülitada. Kui hind on alla maksimumhinna piiri, siis peaks boiler sisse lülituma. Süsteem arvutab boileri töötundide arvu seadistuste ja veetemperatuuri alusel. Näiteks boileri võimsus – 1700 vatti, boileri maht – 100 liitrit, hetketemperatuur – 30 kraadi Celsiuse järgi, soovitud temperatuur – 65 kraadi Celsiuse järgi. Seetõttu peab boiler soovitud temperatuuri saavutamiseks töötama 2 tundi ja 24 minutit¹.

Lisas 7 on börsihinnad kuni 07.00 ja kuni 23.00 29.11.2022. Lähtudes sellest, et boiler peab töötama 2 tundi 24 minutit on vaja leida 3 tundi kõige madalamate hindadega:

- Tunnid madalaimate börsihindadega kuni 07.00 – 00.00, 03.00, 05.00;
- Tunnid madalaimate börsihindadega kuni 23.00 – 07.00, 21.00, 22.00.

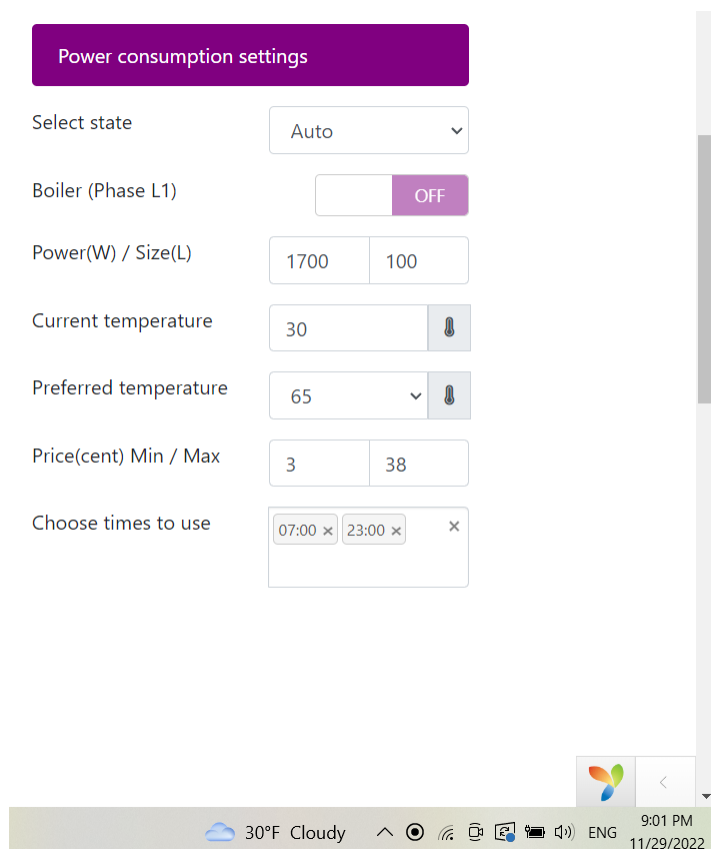
¹ $t = 4.2 \cdot 100 \cdot (65-30) / 3600 / 1.7$

Joonis 12 esitab boileri sisselülitamist kell 21.00 nagu peab.



Joonis 12. Boileri sisselülitamine

Joonisel 13 on kujutatud boileri väljalülitamist, kui maksimumhinna piir on hetkel hinnast madalam.



Joonis 13. Boileri väljalülitamine

4.4 Valminud lahenduse võimalikud edasiarendused

Autori väljatöötatud lahendust võiks täiendada lisafunktsioonide ja tehnoloogiatega. See lõputöö on mudel, mille funktsionaalsuse põhjal saab välja töötada lõpptoote. Edasiseks arendamiseks on võimalikud järgmised täiendused:

- **Koduserveri kasutamine:** kindlasti tasub paigaldada koduserver ja kasutada väljatöötatud prototüüpi kohtvõrgus;
- **Akude kasutamine:** kui päikeseenergiat on palju ja börsihind madal, võiks akud laadida ning kõrge hinna korral päikeseenergiat müüa;
- **Kasutada päikeseenergiat maksimaalselt:** arvestada päikeseenergiaga ning esmalt kasutada seda. Osta tarnijalt energiat ainult siis, kui päikeseenergiast ei piisa vajaduste rahuldamiseks;
- **Börsipaketid:** valida sobiv börsipakett;
- **Teadete saatmine:** saata teateid süsteemi muudatustest kasutaja meilile või telefonile;
- **Erinevate seadmete ühendamine süsteemiga:** võimaldada lisada erinevaid seadmeid.

5 Kokkuvõte

Käesoleva lõputöö eesmärk oli luua prototüüplahendus, mis võimaldab efektiivsemalt tarbida elektrit, automatiseerida koduseadmete tööd muutuva börsihinna alusel ning vastavalt sellele vähendada elektritarbijate poolt börsihinna alusel makstavaid kulusid. Algselt kirjeldas töö autor elektri tootmise ja tarbimisega seotud probleeme ning esitas olemasolevad lahendused. Analüüsi käigus said määratletud funktsionaalsed ja mittefunktsionaalsed nõuded prototüübi loomiseks. Edasi analüüsiti, kuidas saavutada süsteemi efektiivsus kasutaja isiklike vajadusi eiramata. Süsteemi loogika esitati plokkskeemi kujul. Seejärel valiti sobivad tehnoloogiad probleemi lahendamiseks võrdlustabelite põhjal. Arenduseks valis autor LAMP *stack*-i – nelja erineva tarkvaratehnoloogia komplekti: Linux, Apache, MySQL, PHP. Projekt loodi Yii2 raamistiku abil. Frontend töötati välja Yii2 raamistikku sisseehitatud tehnoloogiate abil: Bootstrap, jQuery ja Ajax. Tänu Ajaxile ei laadita veebilehte andmete uuendamisel täielikult uuesti.

Töö alguses püstitatud eesmärk on täidetud. Arendustöö tulemusena loodi süsteem, mis on võimeline lülitama rakendatava elektriseadme voolu börsihinna ja kasutaja poolt määratud parameetrite alusel. See võimaldab elektrit ratsionaalsemalt tarbida, vähendada elektritarbimise teenuste eest tasumise kulusid ja vastavalt sellele ka probleemi lahendada. Loodud lahenduse või lõpptoote massiline kasutamine pärast edasiarendust aitab tagada stabiilsuse elektri nõudluse ja pakkumise vahel ning vähendada keskkonnareostust.

Põhiline päikeseenergiat arvestav lahenduse kasuteguri testimine ja tulemuste analüüs on plaanis läbi viia kevadel, kuna prototüübi loomise hetkel algas talvehooaeg, mil päikeseenergia on ebapiisav või puudub üldse.

Pärast prototüübi edasiarendamist ja testimisest saavad väiketootjad seda kodus kasutada. Kui autor arendusega soovitud tulemuse saavutab, plaanib ta selle lahenduse oma kodus ellu viia.

Kasutatud allikate loetelu

- [1] Save On Energy, „How Does Saving Energy Help The Environment“ 2022. [Võrgumaterjal]. Saadaval: <https://www.saveonenergy.com/green-energy/save-energy-go-green/>. [Kasutatud: 28.10.2022]
- [2] Eesti Energia AS, „Electricity market“ 2022. [Võrgumaterjal]. Saadaval: <https://www.energia.ee/en/era/elekter/elektriturg>. [Kasutatud: 28.10.2022]
- [3] L. Wadowsky, „The best energy-efficient smart home devices for reducing those hefty energy bills“ 2022. [Võrgumaterjal]. Saadaval: <https://thegadgetflow.com/blog/best-energy-efficient-smart-home-devices-reducing-those-hefty-energy-bills>. [Kasutatud: 28.10.2022]
- [4] Constellation, „Top Energy Saving Inventions and Innovations“ 2022. [Võrgumaterjal]. Saadaval: <https://blog.constellation.com/2021/10/14/top-energy-saving-inventions-and-innovations>. [Kasutatud: 28.10.2022]
- [5] E. Heath, „What Are Peak Hours for Electricity?“ 2022. [Võrgumaterjal]. Saadaval: <https://www.familyhandyman.com/article/electricity-peak-hours/#what-are-peak-electricity-hours>. [Kasutatud: 28.10.2022]
- [6] BSLBATT, „Kuidas kasutusaja muud töötab“ 2019. [Võrgumaterjal]. Saadaval: <https://www.lithium-battery-factory.com/et/use-rates>. [Kasutatud: 28.10.2022]
- [7] Elering. [Võrgumaterjal]. Saadaval: <https://dashboard.elering.ee>. [Kasutatud: 01.11.2022]
- [8] E4U, „Water Heater Calculator, Formula for Time Required to Heat“. [Võrgumaterjal]. Saadaval: <https://www.electrical4u.net/energy-calculation/water-heater-calculator-time-required-to-heat-water>. [Kasutatud: 01.11.2022]
- [9] P. Abel, „Aasta päikesepaneelidega – oli küll hea otsus“ 2022. [Võrgumaterjal]. Saadaval: <https://www.tera.ee/aasta-kogemust-paikesepaneelidega>. [Kasutatud: 01.11.2022]
- [10] R. Perälä, „Päikeseelekter“ 2018. [Kasutatud: 01.11.2022]
- [11] Fronius Symo, „Fronius Symo 6.0-3-M“ 2019. [Võrgumaterjal]. Saadaval: https://pood.soojuskeskus.ee/wp-content/uploads/2021/02/Fronius_Inverter_SYMO_web_1.jpg. [Kasutatud: 01.11.2022]
- [12] Fronius International GmbH, „Fronius Symo 6.0-3-M“ 2022. [Võrgumaterjal]. Saadaval: <https://www.fronius.com/en-gb/uk/solar-energy/installers-partners/technical-data/all-products/inverters/fronius-symo/fronius-symo-6-0-3-m>. [Kasutatud: 05.11.2022]
- [13] Sonoff, „Sonoff TH-16“ 2016. [Võrgumaterjal]. Saadaval: <https://www.megateh.eu/files/products/00/51/39/th16sonoff.jpg>. [Kasutatud: 05.11.2022]

- [14] C. Gomes, „Flashing Tasmota firmware onto Sonoff devices over-the-air: “How-to” using just a terminal and web-browser“ 2021. [Võrgumaterjal]. Saadaval: <https://cgomesu.com/blog/ota-tasmota-sonoff>. [Kasutatud: 05.11.2022]
- [15] Amazon Web Services, „What is Java?“ 2022. [Võrgumaterjal]. Saadaval: <https://aws.amazon.com/what-is/java>. [Kasutatud: 05.11.2022]
- [16] S. Bonello, „Pros and cons of programming languages“ 2021. [Võrgumaterjal]. Saadaval: <https://www.chubbydeveloper.com/pros-and-cons-of-programming-languages>. [Kasutatud: 05.11.2022]
- [17] Amazon Web Services, „What is Python?“ 2022. [Võrgumaterjal]. Saadaval: <https://aws.amazon.com/what-is/python>. [Kasutatud: 05.11.2022]
- [18] S. Astari, „What Is PHP? Learning All About the Scripting Language“ 2022. [Võrgumaterjal]. Saadaval: <https://www.hostinger.com/tutorials/what-is-php>. [Kasutatud: 05.11.2022]
- [19] Pluralsight, „Everything you need to know about C#“ 2019. [Võrgumaterjal]. Saadaval: <https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c->. [Kasutatud: 05.11.2022]
- [20] A. Roznovsky, „Why use PHP? Main advantages and disadvantages“ 2021. [Võrgumaterjal]. Saadaval: <https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages>. [Kasutatud: 05.11.2022]
- [21] А. Денисенко, „Фреймворки в веб-разработке — что это, какие существуют и для чего нужны“ 2022. [Võrgumaterjal]. Saadaval: <https://highload.today/frejmvorki-v-veb-razrabotke/#:~:text=Фреймворк%20определяет%20архитектуру%20приложения%20и,с%20фреймворком%20происходит%20инверсия%20управления>. [Kasutatud: 05.11.2022]
- [22] C. Brotherton, „The Most Popular PHP Frameworks to Use in 2022“ 2022. [Võrgumaterjal]. Saadaval: <https://kinsta.com/blog/php-frameworks/#what-are-the-best-php-frameworks-in-2022>. [Kasutatud: 05.11.2022]
- [23] J. Patel, „Yii vs Laravel: Which Web App Framework to Choose?“ 2022. [Võrgumaterjal]. Saadaval: <https://www.monocubed.com/blog/yii-vs-laravel>. [Kasutatud: 05.11.2022]
- [24] SkillFactory, „Yii2“ 2022. [Võrgumaterjal]. Saadaval: <https://blog.skillfactory.ru/glossary/yii2>. [Kasutatud: 05.11.2022]
- [25] CSSChopper, „Top 10 Front End Development Tools to Choose In 2020“ 2020. [Võrgumaterjal]. Saadaval: <https://www.csschopper.com/blog/top-front-end-development-tools>. [Kasutatud: 05.11.2022]
- [26] R. Winter, „What Is Vue.js? (Uses + 5 Website Examples)“ 2022. [Võrgumaterjal]. Saadaval: <https://blog.hubspot.com/website/vue-js>. [Kasutatud: 05.11.2022]
- [27] W3Schools, „XML Ajax“. [Võrgumaterjal]. Saadaval: https://www.w3schools.com/xml/ajax_intro.asp. [Kasutatud: 05.11.2022]

- [28] D. Kumar, „6 Best Databases To Use In 2022“. 2022. [Võrgumaterjal]. Saadaval: <https://hevodata.com/learn/best-database/#types>. [Kasutatud: 10.11.2022]
- [29] Webxloo, „PHP / MySQL Development“. 2022 [Võrgumaterjal]. Saadaval: <https://webxloo.com/technology/php-mysql.html>. [Kasutatud: 10.11.2022]
- [30] OmkarSoft, „Top Benefits of Using PHP Mysql for Developing Dynamic Websites“ 2017. [Võrgumaterjal]. Saadaval: <https://www.omkarsoft.com/blog/top-benefits-using-php-mysql-developing-dynamic-websites/#:~:text=PHP%20is%20a%20server-side,genuinely%20dynamic%20and%20flexible%20content>. [Kasutatud: 10.11.2022]
- [31] javaTpoint, „phpMyAdmin“. [Võrgumaterjal]. Saadaval: <https://www.javatpoint.com/phpmyadmin#:~:text=The%20main%20purpose%20of%20phpMyAdmin,tables%20by%20using%20this%20software>. [Kasutatud: 10.11.2022]

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kristina Sizova

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebirakendus elektriseadmete haldamise ja elektrimüügi automatiseerimiseks”, mille juhendaja on Antoi Meelis
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

22.11.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut omaühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 - Fronius Symo 6.0-3-M



Joonis 14. Fronius Symo 6.0-3-M

Lisa 3 – Sonoff TH-16



Joonis 15. Sonoff TH-16

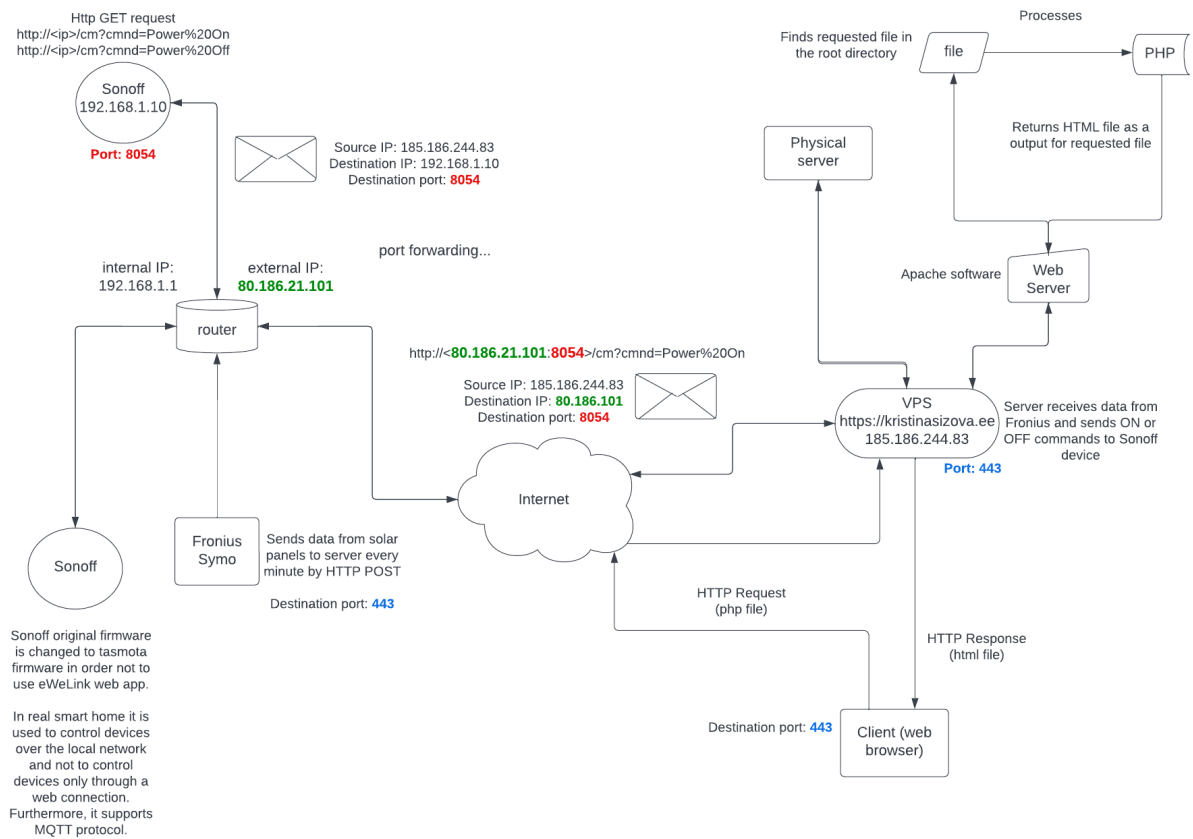
Lisa 4 – Yii ja Laravel raamistike võrdlus

Tabel 6. Yii ja Laravel raamistike võrdlus

Võrdlustegurid	Yii	Laravel
Jõudlus	Yii on veebiarenduse jõudluse osas kiireim raamistik. See raamistik edestab jõudluse poolest Laravelit.	Laravel on Yii-ga võrreldes aeglasem. Toimivuse kiirendamiseks kasutatakse vähe lähenemisviise, nagu klassikaardi optimeerimine ja marsruutimine.
Marsruutimine	Yii pakub automaatset vaikemarsruutimist. Pole vaja iga kontrolleri toimingu jaoks marsruuti deklareerida.	Laraveli puhul on vaja kontrolleri iga toimingu jaoks marsruuti deklareerida ja kasutada ressursi marsruutimist, et deklareerida mitu marsruuti üheks.
Migratsioon	Yii raamistik pakub täielikku tööriista tõhusaks andmete migreerimiseks.	Laravel pakub külviseadet kriitiliste andmete sujuvaks migreerimiseks ja nende andmete testimiseks moodulite kaudu.
Silumine	See raamistik pakub integreeritud silumiskonsooli selliste ülesannete käsitlemiseks nagu andmebaasipäringud, andmebaasipäringute logimine jne.	Laravel pakub kõikehõlmavaid silumistööriistu. Saadaval on tagasihelistamispinuga silumiskonsool.
Õppimiskõver	Yii on üsna lihtne õppida ja mõista. Iga arendaja saab seda raamistikku Yii	Laraveli õppimiskõver on üsna järsk, kuna algaja jaoks on seda

	dokumentide kaudu õppida.	raske õppida. Selle raamistiku kiirus on andmete kogumise ja edastamise osas aeglane.
CRUD põlvkond	Yii-1 on hõlpsasti kasutatav Gii tööriist, mis võimaldab prototüüpi luua, genereerides kasutatud koodifragmente või täielikke CRUD-kontrollereid. Arendajad saavad veebiliidese kaudu hõlpsasti vajalikke juppe genereerida. Samuti on käsurea liides saadaval ka arendajatele, kes eelistavad konsooli kaudu töötada.	Laravel ei sisalda karbist välja võetud CRUD GUI-d, arendajad nõuavad Laraveli kolmanda osapoole CRUD-generaatorite kasutamist, nagu CRUDBoosters, Craftable ja LaravelCRUD Generator.
Kogukonna tugi	Kuna Yii on Laravelist vanem, on selle raamistikul rohkem veebiarendajaid üle kogu maailma, muutes Yii Laraveliga võrreldes suuremaks kogukonnaks.	Laravelil on arvukalt veebirakenduste spetsialiste, kes pakuvad kvaliteetseid tulemusi. Sellel avatud lähtekoodiga raamistikul on kogu maailmas tohutult toetust.
Turvalisus	Yii-1 on sisseehitatud rollipõhine juurdepääsukontroll, mis pakub autentimist, paroolikaitset ja palju muud.	Kuna veebiarendajad peavad tööriistadele juurdepääsuks kasutama laiendusi, pole see turvafunktsioone arvestades õige valik.

Lisa 5 – Marsruutimine Sonoffi, kliendi ja VPS-i vahel



Joonis 16. Marsruutimine süsteemis

Lisa 6 – Päikeseivnerterilt saadud fail

```
→ ~ cat /var/www/html/fronius_data.php
<?php
$filename = 'fronius_data.txt';
if (!file_exists($filename)) {
    file_put_contents($filename, '');
}

if(isset($_POST))
{
    file_put_contents($filename, $_POST);
    exec("curl https://kristinasizova.ee/site/froniusdata");
    exit;
}
?>
```

Joonis 17. Programmikood päikeseivnerterilt failis

Lisa 7 – Elektri börsihinnad 29.11.2022

Tabel 7. Börsihinnad 29.11.2022 00.00–07.00

Kell	00:00	01:00	02:00	03:00	04:00	05:00	06:00
Hind (c/kWh)	24.79	41.79	26.64	23.32	24.99	24.13	28.04

Tabel 8. Börsihinnad 29.11.2022 07.00–23.00

Kell	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00
Hind (c/kWh)	34.82	39.62	44.36	44.53	44.59	43.56	41.81	42.43
Kell	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00
Hind (c/kWh)	44.64	45.07	46	50.11	48.69	45.27	39.04	33.49