

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Hans Martin Tomingas**

**Eesti keeleressurssidel põhinev sõnarägastike generaator**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Sven Aller, MSc

Tartu 2025

## **Eesti keeleressurssidel põhinev sõnarägistike generaator**

### **Lühikokkuvõte:**

Käesoleva praktilise bakalaureusetöö eesmärk oli luua veebipõhine sõnarägistike generaator, mis võimaldab automaatselt koostada eestikeelseid sõnarägistikke sisendteema alusel. Töös kirjeldatakse, kuidas sisendteemaga seotud sõnad leitakse, kuidas nendest sõnadest koostatakse sõnarägistik ning kuidas on üles ehitatud loodud veebirakendus. Valminud rakendus toetab sõnarägistike genereerimist kolmes keeles: eesti, inglise ja saksa keeles. Lisaks saab koostada kakskeelseid sõnarägistikke ning sõnarägistikke, kus peidetud sõnade asemel on vihjeteks sõnade definitsioonid. Tagasiside rakendusele oli valdavalt positiivne ning rakenduse meelelahutuslikku ning keeleõppelist väärtust hinnati kõrgelt.

### **Võtmesõnad:**

keeleteadus, keeletehnoloogia, eesti keel, automaatne genereerimine, veebirakendus, NodeJS, Vue, sõnarägistik

**CERCS:** P175 Informaatika, süsteemiteooria

## **Word Search Generator Based on Estonian Language Resources**

### **Abstract:**

The aim of this practical bachelor's thesis was to develop a web-based word search generator that can automatically create Estonian word search puzzles based on a user-defined topic. The thesis describes the process of finding topic-related words, generating a puzzle grid from those words, and the architecture of the developed web application. The resulting application supports puzzle generation in three languages: Estonian, English, and German. It also allows the creation of bilingual word searches and puzzles where definitions are used as clues instead of the hidden words themselves. Feedback on the application was predominantly positive, and its entertainment and language learning value was highly appreciated.

**Keywords:**

language science, language technology, Estonian language, automatic generation, web application, NodeJS, Vue, word search puzzle

**CERCS:** P175 Informatics, system theory

# Sisukord

<b>Sissejuhatus</b> .....	<b>5</b>
<b>Mõisted ja terminid</b> .....	<b>7</b>
<b>1. Keeleõpe ja keelemängud</b> .....	<b>8</b>
1.1 Keelemängude roll hariduses.....	8
1.2 Sõnarägistikud.....	8
1.2.1 Olemasolevad ressursid ja nende puudused.....	10
<b>2. Sõnarägistike generaatori veebirakendus</b> .....	<b>12</b>
2.1 Wordneti andmete kasutamine.....	12
2.1.1 EstNLTK teek.....	13
2.1.2 Relatsiooniline andmebaas.....	14
2.1.3 Wordneti lugemine operatiivmällu.....	14
2.2 Võimalike teemade kuvamine kasutajale.....	16
2.3 Sõnarägistiku loomine.....	18
2.3.1 Kandidaatsõnade valimine rägistikku.....	18
2.3.2 Rägastiku genereerimine.....	21
2.4 Tagaliides.....	23
2.5 Eesliides.....	25
2.6 Valminud rakendus.....	27
<b>3. Tagasiside rakendusele</b> .....	<b>28</b>
3.1 Testijate taust.....	29
3.2 Tagasiside sõnarägistikele.....	29
3.3 Tagasiside kasutajaliidesele.....	30
3.4 Üldine tagasiside.....	30
<b>4. Edasiarenduse võimalused</b> .....	<b>32</b>
4.1 Testijate soovitusel.....	32
4.2 Rakenduse arendamise käigus tekkinud ideed.....	32
<b>Kokkuvõte</b> .....	<b>34</b>
<b>Viidatud kirjandus</b> .....	<b>35</b>
<b>Lisad</b> .....	<b>38</b>
<b>I. Eesti Wordneti LMF XML-formaadis faili näidis</b> .....	<b>38</b>
<b>II. Eesti keele tähtede jaotus</b> .....	<b>39</b>
<b>III. Genereeritud sõnarägistik teemal „Sport”</b> .....	<b>39</b>
<b>IV. Dokumendina salvestatud sõnarägistik</b> .....	<b>41</b>
<b>V. Eestikeelne rakenduse testimise küsimustik</b> .....	<b>42</b>
<b>Litsents</b> .....	<b>48</b>

## Sissejuhatus

Sõnarägastikud on populaarsed keelemängud, milles mängija otsib peidetud sõnu tähtede ruudustikust. Neid kasutatakse laialdaselt nii meelelahutuses kui ka hariduses, eriti sõnavara ja õigekirja arendamiseks. Keeleõppes aitavad sõnarägastikud tõhusalt tugevdada uute sõnade meeldejätmist ning parandada tähelepanu- ja keskendumisvõimet [1]. Pedagoogilised uuringud on näidanud, et mängulised õppemeetodid, sealhulgas keelemängud nagu sõnarägastikud, suurendavad õppijate motivatsiooni ja toetavad teadmiste kinnistumist [2, 3].

Kuigi internetis leidub mitmeid sõnarägastike generaatoreid, ei arvesta need sageli keeleõppe spetsiifiliste vajadustega. Eelkõige ei toeta olemasolevad tööriistad sõnarägastike automaatset genereerimist kindla teema põhjal. Samuti ei võimalda need tööriistad kasutada vihjeid, mis erinevad peidetud sõnadest, näiteks definitsioonide või tõlgete kujul. Eesti keele kontekstis on täiendavaks piiranguks asjaolu, et paljud tööriistad ei arvesta eesti keele eripärade, sealhulgas täpitähtedega.

Käesoleva bakalaureusetöö eesmärk on luua veebipõhine sõnarägastike generaator, mis kasutab keeleressursse, et automaatselt koostada eestikeelseid teemakeskseid sõnarägastikke. Rakendus suudab leida ja valida sõnu etteantud valdkonnast, määrata neile sobivad vihjed ning peita need tähtede ruudustikku. Lisaks klassikalistele sõnarägastikele, kus vihje ja peidetud sõna kattuvad, toetab loodav süsteem ka kakskeelseid sõnarägastikke ning definitsioonipõhiseid variatsioone. Samuti saab genereerida kolmele erinevale raskuastmele – lihtne, keskmine ja raske – vastavaid sõnarägastikke, mis erinevad rägastiku suuruse, peidetud sõnade suuna ja vihjete tüübi poolest. Töö praktiline väljund on mitmekülgne tööriist, mis ühendab meelelahutusliku ja haridusliku väärtuse ning sobib eriti hästi keeleõppe konteksti.

Bakalaureusetöö on jagatud neljaks osaks. Esimeses peatükis antakse ülevaade sõnarägastike ja teiste keelemängude kasutusest keeleõppes ning hariduses, samuti analüüsitakse olemasolevaid sõnarägastike generaatoreid ja nende piiranguid. Teises

peatükis kirjeldatakse sõnarägistike genereerimise meetodikat ning veebirakenduse arhitektuuri. Kolmas peatükk keskendub loodud rakenduse toimivuse hindamisele kasutajate tagasiside põhjal. Viimases peatükis esitatakse ideid ja võimalusi rakenduse edasiseks arendamiseks.

## Mõisted ja terminid

<p>Eesliides (ingl <i>front end</i>)</p> <p>Veebilehele ilmuv kasutajaliides, mis võimaldab veebisaidi külastajal kahepoolset suhelda saidi dünaamiliste osadega nagu andmebaasid, ostukorviprogrammid ja onlain-ostutöötlustarkvara.<sup>1</sup></p>
<p>Tagaliides (ingl <i>back end</i>)</p> <p>Rakenduse osa, mille ülesandeks on kaudselt toetada eesliidest ning mis on tavaliselt vajalikule ressursile lähemal. Klient/server kontseptsiooni raames võib eesliidest vaadelda kliendina ja tagaliidest serverina.<sup>1</sup></p>
<p>Teek (ingl <i>library</i> või <i>module</i>)</p> <p>Valmiskompileeritud alamprogramm, mida programm saab kasutada. Teegid on eriti kasulikud sagelikasutatavate alamprogrammide salvestamiseks, sest siis pole neid vaja käsilolevasse programmi otse sisse kirjutada.<sup>1</sup></p>
<p>Paisketabel (ingl <i>hash table</i> või <i>hashmap</i>)</p> <p>Andmestruktuur, kus võtmele vastava väärtuse asukoha indeks arvutatakse võtme räsina. Paisketabel kiirendab otsingut lünkliku võtmete jadaga süsteemis.<sup>2</sup></p>
<p>XML (ingl <i>XML</i> ehk <i>Extensible Markup Language</i>)</p> <p>Platvormist sõltumatu märgistuskeel inim- ja masinloetavate vormingute genereerimiseks struktureeritud andmeid sisaldavatele tekstidele.<sup>2</sup></p>
<p>paralleeltöötlus (ingl <i>parallel processing</i>)</p> <p>Mitme protsessi üheaegne töötlemine.<sup>1</sup></p>

<sup>1</sup> <https://www.vallaste.ee>

<sup>2</sup> <https://akit.cyber.ee/>

# 1. Keeleõpe ja keelemängud

Käesolevas peatükis käsitletakse mänguliste meetodite, eelkõige keelemängude ja sõnarägastike, rolli keeleõppes. Tutvustatakse sõnarägastiku kui keelemängu erinevaid tüüpe ning analüüsitakse olemasolevaid sõnarägastike generaatoreid ja tuuakse välja nende piirangud.

## 1.1 Keelemängude roll hariduses

Mängulistel õppemeetoditel on hariduses oluline roll, eriti keeleõppes, kus need aitavad kaasa nii teadmiste omandamisele kui ka õppijate motivatsiooni tõstmisele. Keelemängud, sealhulgas sõnamängud ja mõistatused, on tõhusad vahendid sõnavara õppimiseks, grammatika harjutamiseks ja keelekasutuse kinnistamiseks autentsetes, samas pingevabas kontekstis [3, 4].

Uuringud on näidanud, et keelemängude kasutamine aitab parandada õppijate keskendumisvõimet ja tähelepanu, toetab loovust ning tugevdab sõnade tähenduse ja õigekirja meeldejätmist [2, 3]. Lisaks loovad mängulised tegevused õppijate jaoks positiivse ja kaasava õpikeskkonna, mis on eriti oluline võõrkeeleõppes, kus esialgne ebakindlus võib olla takistav tegur [5, 6]. Sõnarägastikud on üks selline mäng, millel on mitmekesised võimalused õppimise toetamisel ning mida käsitletakse järgnevalt põhjalikumalt.

## 1.2 Sõnarägastikud

JC Franco käsitluse [7] kohaselt on sõnarägastik (ka sõnaotsimispusle) keeleline mõistatus või mäng, mille eesmärk on leida tähtede ruudustikust peidetud sõnad. Klassikaline sõnarägastik koosneb tähtede ruudustikust, kust peab üles otsima nimekirjana etteantud sõnad. Sõnad võivad rägastikus paikneda nii horisontaalselt, vertikaalselt kui ka diagonaalis (joonis 1).



Joonis 1. Lahendatud sõnarägastik The Word Searchi lehel<sup>3</sup>.

Sõnarägastikest on kujunenud mitmeid variatsioone. Mõnes variandis ei anta nimekirjana ette otsitavad sõnad, vaid vihjed või küsimused, millele peab vastuseid rägastikust otsima, sarnanedes nii ristsõnadele. Samuti leidub kakskeelseid sõnarägastikke, kus peidetud sõnad on ühes keeles ja nende vihjed teises. Üks sõnarägastiku variatsioon on selline, kus täpne nimekiri peidetud sõnadest pole nähtav, vaid on teada rägastiku teema ning mängija peab üles leidma kõik teemaga seotud sõnad [7].

Sõnarägastikud ühendavad mängulisuse ja õppimise viisil, mis soodustab uue sõnavara kinnistumist ning toetab tähelepanu ja keskendumise arengut [1]. Kuna sõnarägastikus tuleb sõnu leida erinevatest suundadest, arendab see visuaalset tajut ja aitab õppijatel paremini meelde jätta sõnade kirjalikult [8]. Sõnavara õppimine sisuliselt seotud sõnade rühmade ja definitsioonide kaudu on tõhus viis sõnatähenduste mõistmiseks ja semantiliste seoste loomiseks [9, 10]. Sõnarägastikud toetavad seda lähenemist temaatiliste ja definitsioonipõhiste variatsioonide kujul, mis aitavad õppijatel seostada uusi sõnu olemasolevate teadmistega. Lisaks pakuvad sõnarägastikud vaheldust traditsioonilistele harjutustele, mis aitab hoida õpimotivatsiooni [1].

<sup>3</sup> <https://thewordsearch.com>

### 1.2.1 Olemasolevad ressursid ja nende puudused

Internetiavarustes leidub mitmeid veebipõhiseid tööriistu, mis võimaldavad sõnarägakstikke genereerida. Levinumad näited on The Word Search<sup>3</sup>, Discovery Education<sup>4</sup> ja The Teacher's Corner<sup>5</sup>. Nimetatud veebilehtedel on võimalik luua sõnarägakstikke, andes sisendiks nimekirja sõnadest, mida soovitakse rägakstikku peita.

Sellistel tööriistadel on aga mitmeid piiranguid. Esiteks ei toeta enamik olemasolevaid lahendusi keskse teemaga seotud sõnarägakstike automaatset loomist – kasutaja peab kõik sõnad käsitsi sisestama, mis muudab protsessi ajamahukaks ning vähendab selle hariduslikku väärtust. Kui kasutaja peab kõik sõnad ise valima ja sisestama, siis teab ta neid juba ette, mistõttu ei paku loodud sõnarägakstiku lahendamine talle sisulist õppimiskogemust. Lisaks pakuvad need tööriistad ainult piiratud valikut eeldefineeritud teemasid, näiteks „Kujundid”, „Värvid” või „Loomad”, ning seda peamiselt inglise keeles. Selliseid sõnarägakstikke ei ole võimalik sisuliselt kohandada ega tõlkida.

Teiseks puudub enamikul rakendustest võimalus luua alternatiivseid sõnarägakstikke, kus vihjed on sõnade definitsioonid või vasted teises keeles. Samas on keeleõppes eriti tõhusad just need lähenemised, mis toetuvad sõnade tähendusele ja sisulistele seostele [10]. Sellised sõnarägakstiku variatsioonid võimaldaksid õppijal paremini siduda uusi sõnu nende tähendusega, muutes harjutuse sisukamaks ja toetades sõnavara kinnistumist.

Kolmandaks ei toeta need rakendused tihti eesti tähestikus leiduvaid täpitähti: Ö, Ä, Ö, Ü. Näiteks Dictionary.com<sup>6</sup> veebilehel sõnarägakstikku luues ei hoiatata kasutajat, et täpitähed ei ole toetatud ning tulemusena poolitatakse sõnad täpitähtede kohalt: „känguru” → „k” + „nguru” (joonis 2). The Word Searchi lehel ei lubata täpitähti sisaldavaid sõnu rägakstikku lisada (joonis 3). Tihti on täpitähtedega sõnade lisamine küll lubatud, kuid ruudustiku tühjade väljade (väljad, mis pole osa ühestki peidetud sõnast) täitmisel suvaliste tähtedega kasutatakse inglise tähestikku, mistõttu ei täideta tühjasid välju kunagi täpitähtedega ning ülemäära sagedasti valitakse eesti keele seisukohalt

<sup>4</sup> <https://puzzlemaker.discoveryeducation.com/word-search>

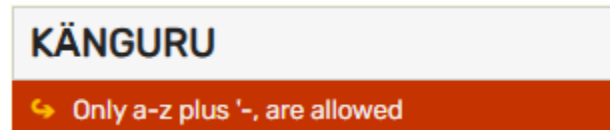
<sup>5</sup> <https://worksheets.theteacherscorner.net/make-your-own/word-search>

<sup>6</sup> <https://www.dictionary.com/e/word-search-maker>

võõrtähti: C, Q, W, X, Y. See teeb eestikeelsete sõnade leidmise rägastikust liiga lihtsaks.



Joonis 2. Dictionary.comis loodud sõnarägastiku sõnade nimekiri, kus sisendsõnad on poolitatud täpitähtede kohalt.



Joonis 3. The Word Search lehel kuvatav hoiatus, kui üritada rägastikku lisada täpitähti sisaldavat sõna.

Käesolevas bakalaureusetöös loodud veebirakendus püüab ületada kõik eelmainitud kitsaskohad, võimaldades automaatselt genereerida eestikeelseid, teemakeskseid ja mitmekesiste vihjetega sõnarägastikke, pakkudes valminud rägastike lahendajatele suuremat väljakutset.

## 2. Sõnarägistike generaatori veebirakendus

Peatükis kirjeldatakse rakenduse loomise protsessi, seejuures antakse ülevaade keeleliste andmete kogumisest ja töötlemisest, sõnarägistiku genereerimisest rakenduse tagaliideses ning veebirakenduse eesliidesest. Lõpuks tutvustatakse valminud veebirakendust tervikuna.

### 2.1 Wordneti andmete kasutamine

Sõnarägistike genereerimise rakenduse keskne funktsioon on leida kasutaja sisestatud teema või märksõna põhjal tähenduselt seotud sõnu. Selleks kasutatakse leksikaalseid andmebaase, mis põhinevad wordneti struktuuril.

Käesoleva töö fookus on Eesti Wordneti (EstWN) kasutamisel. EstWN järgib Princetoni WordNeti ja EuroWordNeti põhimõtteid ning koosneb sünohulkadest – sünonüümsete sõnade kogumitest, mis väljendavad sama mõistet – ning neid ühendavatest semantilistest seostest, nagu ülemmõisted (hüperonüümid) ja alammõisted (hüponüümid) [11]. Selline ülesehitus võimaldab modelleerida mõistete tähenduslikke suhteid ning toetab sõnade temaatilist grupeerimist.

EstWN sisaldab ligikaudu 92 000 lemmat ja üle 200 000 semantilise seose (seisuga detsember 2024) [11]. Rakenduses tuginetakse nendele andmetele, et genereerida temaatilisi sõnarägistikke, mille sõnavara on seotud mõisteliselt või tähenduslikult.

Lisaks EstWN-ile kasutatakse rakenduses ka inglise ja saksa wordneti andmestikke, et toetada kakskeelsete sõnarägistike loomist. Sellistes sõnarägistikes on peidetud sõnad ühes keeles ja vastavad vihjed teises keeles, pakkudes võimalust õppida eesti keelt inglise või saksa keele kaudu – või vastupidi. Keeltevaheline seos luuakse ühise kontseptuaalse identifikaatori ILI (ingl *interlingual index*) abil, mis seob erinevate keelte sünohulgad semantilisel tasandil.

Tõhusaks tööks on oluline, et Wordneti andmetele oleks kiire ja ressursisäästlik ligipääs. Järgnevalt võrreldakse kolme võimalikku lähenemisviisi Wordneti andmestiku

kasutamiseks veebirakenduses: EstNLTK teegi kasutamine, relatsioonilise andmebaasi kasutamine ning andmete laadimine operatiivmällu.

### **2.1.1 EstNLTK teek**

Üks võimalus Eesti Wordneti kasutamiseks rakenduses on EstNLTK teegi vahendusel. EstNLTK<sup>7</sup> on Pythonis kirjutatud keeletehnoloogiline teek, mis sisaldab integreeritud Eesti Wordneti versiooni. Selle abil on võimalik lemmade, sünohulkade ja semantiliste seoste töötlemine ilma vajaduseta luua eraldi andmestruktuure või laadida andmeid käsitsi.

#### **Eelised**

- Lihtne kasutada Pythonis loodud programmides.
- Teek pakub mugavat liidest, mis võimaldab ligipääsu lemmadele ja sünohulkade vahelistele seostele.

#### **Puudused**

- EstNLTK kasutab sisemiselt SQLite-andmebaasifaili, mis ei ole lihtsasti muudetav ega inimloetavas vormingus. See muudab keerukaks wordneti andmete kohandamist, näiteks vigade parandamist või täienduste lisamist.
- Teek ei lae vaikimisi kogu wordneti andmestikku mällu, vaid loeb andmeid kettalt vastavalt vajadusele, mis tähendab sagedasi kettapöördusi. Kuna kettalt lugemine on üldiselt aeglasem kui mälust lugemine, võib see põhjustada jõudlusprobleeme, eriti suuremate andmemahutude puhul [12].
- EstNLTK sisaldab palju funktsionaalsust (nt morfoloogiline analüüs, lausete segmentimine), mida antud töös ei vajata. Samuti sisaldab wordneti andmestik palju sellist infot (nt näitelaused, sõnaliigid), mida rakendus ei kasuta, kuid mille mällu laadimine suurendaks ressursikasutust.

Seetõttu, kuigi EstNLTK pakub mugavat ligipääsu Eesti Wordneti andmetele, ei sobi see hästi jõudluskriitilise veebirakenduse konteksti, kus on oluline kiire andmepöördus ja täielik kontroll kasutatava andmestiku üle.

---

<sup>7</sup> <https://github.com/estnltk/estnltk>

## 2.1.2 Relatsiooniline andmebaas

Teise võimalusena saab wordneti andmed salvestada relatsioonilisse andmebaasi (nt PostgreSQL või SQLite). Wordneti struktuuri on võimalik modelleerida tabelitena, kus eraldi tabelid esindavad lemmade loendit, sünohulki ja nendevahelisi semantilisi seoseid.

### Eelised

- Andmeid hoitakse kettal, mistõttu on rakenduse mälu kasutus madal.
- SQL-päringud võimaldavad keerukate seoste ja filtrite kasutamist, mis võib olla kasulik näiteks täpsemate otsingutingimuste rakendamisel.

### Puudused

- Ligipääs kettal olevatele andmetele on aeglasem kui andmetele mälus, mis võib vähendada rakenduse jõudlust [12].
- JavaScripti-põhises serverikeskkonnas (nt Node.js) tuleb SQL-päringute tulemused teisendada JavaScripti objektideks, mis lisab töötusaega.
- Relatsioonilised andmebaasid pakuvad mehhanisme, mis tagavad andmete terviklikkuse samaaegsel kirjutamisel ja lugemisel (ACID-omadused) [13, 14]. Kuna wordneti andmed on staatilised ja neid ei muudeta, on need mehhanismid käesoleva rakenduse jaoks üleliigsed ning võivad andmepöördusi asjatult aeglustada.

Relatsioonilise andmebaasi kasutamine pakub küll paindlikkust keerukamate päringute koostamisel, kuid ei sobitu hästi sellesse projekti, kus on oluline võimalikult kiire ja vahetu ligipääs staatilistele andmetele.

## 2.1.3 Wordneti lugemine operatiivmällu

Kolmas ja käesolevas rakenduses kasutusele võetud lähenemisviis on wordneti andmete laadimine otse operatiivmällu. Andmed konverteeritakse esmalt sobivasse JavaScripti vormingusse ja laaditakse seejärel mällu, kust neile saab kiiresti ligi.

## Eelised

- Kõige kiirem ligipääs: ei toimu aeglaseid kettapöördusi, kõik vajalikud andmed on mälus.
- Täielik kontroll andmete üle: andmeid saab vajadusel muuta, täiendada või filtreerida.
- Lihtne integreerida Node.js-i rakendusega: kasutatakse JSON-struktuure ja JavaScripti andmeformaate.

## Puudused

- Vajab rohkem operatiivmälu, eriti mitme keele paralleelsel kasutamisel.
- Rakenduse käivitamisel tuleb kogu andmestik esmalt parsida ja laadida mällu, mis võib suurendada alglaadimise aega.

Valminud veebirakenduses kasutatakse kolme keele – eesti<sup>8</sup>, inglise<sup>9</sup> ja saksa<sup>10</sup> – wordneti andmeid. Kasutatakse LMF XML-formaadis<sup>11</sup> (ingl *Lexical Markup Framework compatible XML*) faile, mis sisaldavad järgnevaid märgendeid:

- 1) *<LexicalEntry>*: lemmad ja nende vormid;
- 2) *<Synset>*: sünohulgad unikaalse ID ja definitsiooniga, märgend sisaldab ka atribuuti *ILI*, mis on keelteülene identifikaator sama tähendusega mõistetele;
- 3) *<Sense>*: lemmade seosed sünohulkadega;
- 4) *<SynsetRelation>*: semantilised suhted sünohulkade vahel;
- 5) *<Definition>* või *<ILIDefinition>*: mõistete seletused.

Lisas 1 on välja toodud lõik Eesti Wordneti LMF XML-formaadis failist. Sarnase ülesehitusega XML-failid parsitakse ja teisendatakse järgnevateks JavaScripti objektideks:

- 1) *lemmaToSynsets*: paisketabel, mis seob lemmad vastavate sünohulkade ID-dega;

---

<sup>8</sup> <https://www.cl.ut.ee/ressursid/teksaurus/>

<sup>9</sup> <https://en-word.net/>

<sup>10</sup> <https://github.com/hdaSprachtechnologie/odenet>

<sup>11</sup> <https://globalwordnet.github.io/schemas/>

- 2) *synsetToLemmas*: paisketabel, mis seob sünohulgad neile vastavate lemmadega;
- 3) *synsets*: paisketabel sünohulkade vaheliste seoste (nt hüperonüümid, hüponüümid) hoidmiseks;
- 4) *synsetDefinitions*: paisketabel sünohulkade definitsioonide hoidmiseks;
- 5) *synsetToIli*: paisketabel, mis seob sünohulgad vastava ILI ID-ga;
- 6) *iliToSynsets*: paisketabel, mis võimaldab ILI ID kaudu leida samatähenduslikke sünohulki eri keeltes.

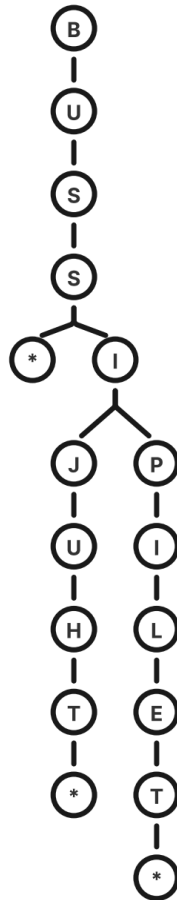
Selline ülesehitus võimaldab tõhusalt genereerida sama teemaga seotud sõnade kogumeid ja seostada vastetega teistes keeltes, toetades seeläbi mitmekeelseid sõnarägistikke.

## 2.2 Võimalike teemade kuvamine kasutajale

Sõnarägistiku loomiseks peab kasutaja andma sisendiks mõne teema – sõna, mis eksisteerib wordnetis. Kuna kasutaja ei pruugi täpselt teada, millised sõnad wordnetis leiduvad, peab rakendus sisendi trükkimisel pakkuma automaatseid soovitusi.

Probleem lahendati prefikspõhise otsinguga: kasutaja sisestatud prefiksi ehk sõne alguse põhjal kuvatakse sobivaid lemmade soovitusi. Näiteks kui kasutaja sisestab „*buss*”, tagastatakse tulemused nagu „*buss*”, „*bussijuht*” ja „*bussipilet*”.

Et selline otsing oleks kiire, loodi spetsiaalne andmestruktuur – prefiksipuu (ingl. *prefix tree* või *trie*). Selles puus esindab iga haru üht tähelist osa ning sõna lõpp tähistatakse spetsiaalse markeriga [15]. Joonisel 4 on kujutatud sõnade „*buss*”, „*bussijuht*” ja „*bussipilet*” paiknemine prefiksipuus. Joonisel tähistab sõna lõppu sümbol „\*”.



Joonis 4. Prefiksipuu sõnadega buss, bussijuht, bussipilet.

Prefiksipuu sobis lemmade hoidmiseks ideaalselt kahel põhjusel. Esiteks toimub prefiksipuus otsing ajas  $O(k)$ , kus  $k$  on sisestatud tähejada pikkus [15]. See tähendab, et sobivate lemmade leidmine toimub etteaimatava ajakuluga, sõltumata sõnade koguarvust, ning peaaegu alati kiiremini kui tavalise massiivi korral. Teiseks kasutab prefiksipuu vähem mälu kui kõiki sõnu sisaldav massiiv, sest prefiksipuu korral salvestatakse sõnade ühised prefiksid ainult ühe korra [15].

Käesolev rakendus ehitab käivitamisel iga wordneti kohta ühe prefiksipuu wordneti kõigist lemmadest ning hoiab seda operatiivmälu, tagades kiire ja sujuva kasutajakogemuse sisendi soovitude andmisel. Kasutajale tagatakse prefiksi põhjal kuni kümme soovitud. Tagastatakse vaid lemmad, mille alusel oleks reaalselt võimalik sõnarägastik luua, ehk lemmad, millele vastavatel sünohulkadel on vähemalt üks kitsamale mõistele viitav semantiline suhe. Tagastatavad lemmad on järjestatud

sünohulkade semantiliste suhete arvu järgi kahanevalt. Sünohulgad, millel on rohkem semantilisi suhteid, tähistavad ilmselt üldisemaid mõisteid ning sobivad seetõttu paremini sõnarägistiku teemaks. Seega kuvatakse nendele sünohulkadele vastavad lemmad soovitude jadas eespool. Selline järjestamise loogika on inspireeritud Google'i otsingumootori algoritmist PageRank, mis järjestab otsingutulemused vastavalt sellele, kui paljud teised lehed viitavad antud lehele [16].

## **2.3 Sõnarägistiku loomine**

Järgnevalt kirjeldatakse, kuidas genereeritakse sõnarägistikke loodud veebirakenduses. Seejuures eristatakse kahte etappi: kandidaatsõnade valimine sõnarägistikku ning kandidaatsõnade põhjal rägastiku genereerimine.

### **2.3.1 Kandidaatsõnade valimine rägastikku**

Sõnade valimisel sisendteema põhjal on oluline, et kõik valitud sõnad oleksid teemaga semantiliselt seotud. Selleks rakendatakse wordneti sünohulkade vahelisi semantilisi seoseid. Käesolevas töös eristatakse kolme tüüpi semantilisi seoseid:

- 1) semantiline seos, mis viitab kitsamale mõistele: hüponüüm, meronüüm, derivatsioon jne;
- 2) semantiline seos, mis viitab üldisemale mõistele: hüperonüüm, holonüüm jne;
- 3) muud nõrgemad semantilised seosed, mille puhul sõltub kontekstist, kumb mõiste on üldisem ja kumb kitsam, nt sarnased mõisted (ingl *similar*) või antonüümid.

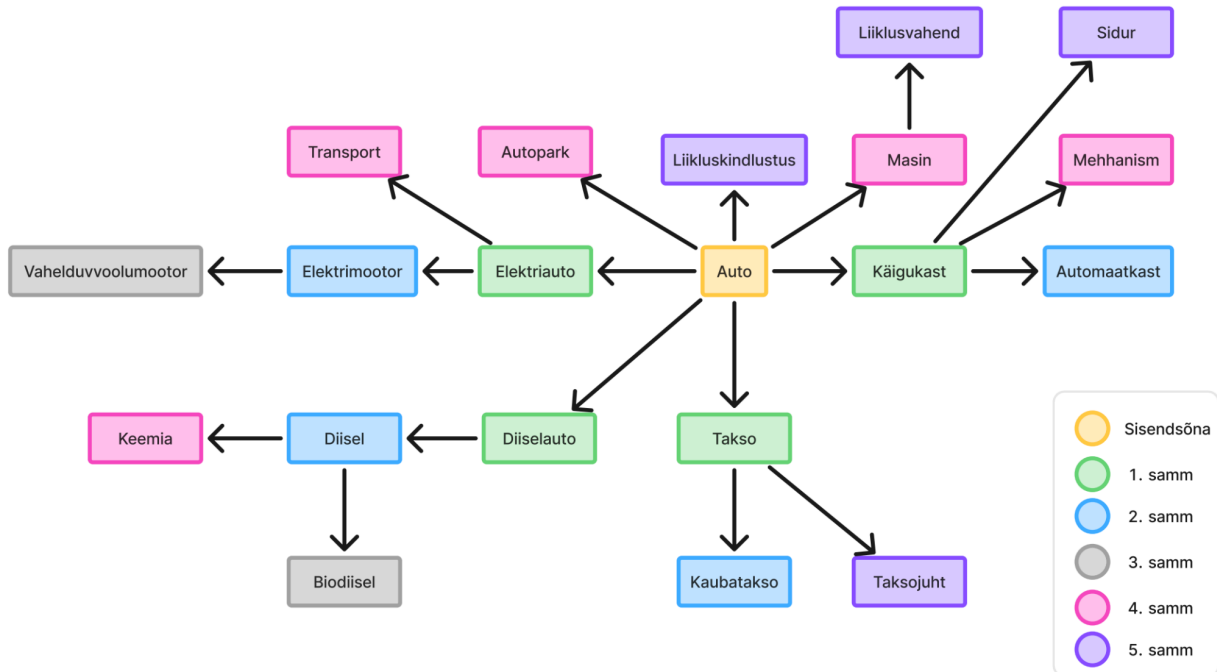
Teemaga seotud sõnade otsimisel uuritakse esmalt seoseid, mis viitavad kitsamatele mõistetele, et vältida otsingupuu liigset hargnemist ning teemasse mittedobivate sõnade sattumist valituks. Alles siis, kui kitsamad mõisted on ammendatud, kuid sõnarägistiku koostamiseks pole leitud piisavalt sõnu, laiendatakse otsingut esmalt üldisematele mõistetele ning lõpuks muude nõrgemate semantiliste seostega ühendatud sünohulkadele.

Kandidaatsõnade valimise algoritm koosneb neljast sammust.

1. Sisendteemale vastav sünohulk (või sünohulgad) märgitakse loetuks ning otsitakse kõik sellega seotud sünohulgad, mis viitavad kitsamale mõistele. Seotud sünohulgad lisatakse järjekorda.
2. Kõik järjekorras olevad sünohulgad märgitakse loetuks ning neile vastavad lemmad lisatakse kandidaatsõnade nimekirja. Iga järjekorras oleva sünohulga puhul otsitakse omakorda sellega seotud sünohulgad, mis viitavad kitsamale mõistele ning need sünohulgad lisatakse järjekorda. Protsessi korratakse kokku kolm korda, et jõuda sügavamate seosteni.
3. Iga seni avastatud sünohulga korral leitakse kõik antud sünohulgaga seotud üldisemale mõistele viitavad sünohulgad ning nendele vastavad lemmad lisatakse kandidaatsõnade nimekirja.
4. Iga seni avastatud sünohulga korral leitakse kõik muude nõrgemate semantiliste seostega ühendatud sünohulgad ning nendele vastavad lemmad lisatakse kandidaatsõnade nimekirja.

Kandidaatsõnad lisatakse nimekirja sõna-vihje paaridena, kus sõnad on rägastikku peidetavad lemmad ning vihjed on sõnarägastiku kõrval kuvatavad sõnad või terminid. Klassikalise sõnarägastiku korral on sõnaks ja vihjeks sama lemma, kuid erineva sisend- ja väljundkeelega sõnarägastiku korral on vihje sisendkeeles ning peidetud sõna väljundkeeles. Erinevates keeltes vastavate sõnade leidmiseks kasutatakse eelnevalt mainitud sünohulga ILI identifikaatorit. Alternatiivina on võimalik luua ka sõnarägastikku, kus vihjeteks on peidetud sõnade definitsioonid. Sel juhul leitakse sünohulgale vastav definitsioon *synsetDefinitions* tabelist.

Iga sammu käigus leitud sünohulgad on eeldatavasti sisendteemaga üha nõrgemalt seotud ning seetõttu lisatakse sõna-vihje paarid nimekirja järjest väiksema kaaluga, muutes nende valimise sõnarägastiku koostamisel ebatõenäolisemaks. Algoritmi tööd sisendteemaga „auto” illustreerib joonis 5.



Joonis 5. Sõnade valimise algoritmi töö sisendsõnaga „auto”.

Algoritm lõpetab töö ning tagastab kandidaatsõnade nimekirja, kui on täidetud üks kahest tingimusest:

- 1) algoritm on jõudnud sammu neli lõppu ning järjekorras pole enam ühtegi sünohulka, mida uurida;
- 2) nimekirja lisatud sõnade pikkuste summa on vähemalt kolm korda suurem sõnarägistiku tähtede arvust, näiteks kui loodava sõnarägistiku suurus on 15 rida × 15 veergu, siis lõpetab algoritm töö, kui nimekirjas olevate sõnade pikkuste summa on vähemalt 675 tähte ( $15 \times 15 \times 3 = 675$ ).

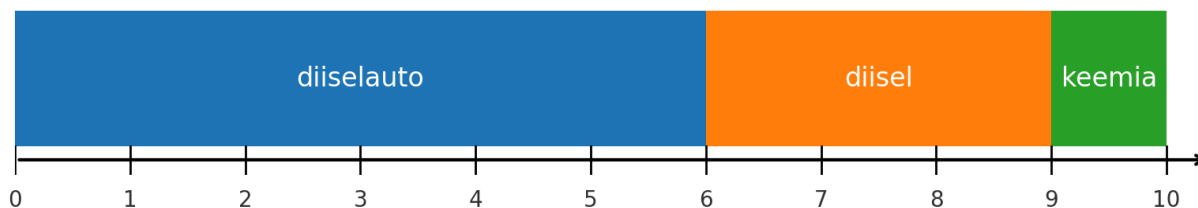
Kandidaatsõnade nimekiri, mille põhjal sõnarägistik järgmises sammus luuakse, sisaldab rohkem sõnu, kui neid rägastikku mahub. Seda kahel põhjusel: esiteks lubab see rägastiku genereerimisel nimekirjast sõnu valida juhuslikult, mis tagab, et sama sisendteemaga genereeritud sõnarägistikud erinevad alati natuke peidetud sõnade loetelu poolest. Teiseks: kui rägastikku sobivaid kandidaatsõnu on rohkem kui vaja, siis võib mõne sõna vahele jätta, kui selle rägastikku paigutamine on liiga keeruline.

### 2.3.2 Rägastiku genereerimine

Peale sisendteema põhjal kandidaatsõnade nimekirja loomist genereeritakse sõnadest rägastik. Käesolevas rakenduses kasutatav sõnarägastiku genereerimise algoritm põhineb Jamis Bucki loodud algoritmil [17], mida on töö raames täiustatud ning tõlgitud Ruby programmeerimiskeelest JavaScripti. Järgnevalt on kirjeldatud kandidaatsõnade nimekirja ning lisaparameetrite (sõnarägastiku keel, soovitud raskusaste) põhjal rägastiku loomise protsessi.

Sõnarägastiku genereerimist alustatakse tühja ruudustiku loomisega, kusjuures ruudustiku suurus sõltub soovitud sõnarägastiku raskusastmest, lihtne sõnarägastik on mõõtmetega  $10 \times 10$ , keskmine  $15 \times 15$  ning raske  $20 \times 20$ . Loodud ruudustikku paigutatakse ühekaupa tsükliliselt sõnad. Seejuures sooritab algoritm järgmised sammud.

1. Valib kandidaatsõnade seast juhuslikult sõna, eelistades suurema kaaluga sõnasid. Põhimõtteliselt paigutab algoritm järjestikuselt kõik kandidaatsõnad arvteljele vastavalt sõna kaalule ehk suurema kaaluga sõna esindab arvteljel suuremat vahemikku. Seda illustreerib joonis 6, kus arvteljele on paigutatud sõnad „diiselauto”, „diisel” ja „keemia”. Kuna sõna „diiselauto” kaal on suurim, katab see arvteljel suurima vahemiku, mistõttu on juhuarvu genereerimisel kõige tõenäolisem, et valituks osutub just see sõna.



Joonis 6. Erineva kaaluga kandidaatsõnad arvteljel.

2. Genereerib juhuarvu vahemikus null kuni kõikide kandidaatsõnade kaalude summa ning valib arvteljelt juhuarvule vastava sõna. Seda sõna proovib algoritm järgnevalt ruudustikku paigutada.

3. Genereerib juhusliku järjestuse kõikidest ruudustiku positsioonidest (rea indeks ning veeru indeks). Antud positsioonid tähistavad kandidaatsõna algustähe võimalikke positsioone.
4. Genereerib juhusliku järjestuse proovitavatest sõna suundadest. Lihtsa raskusastme korral võivad sõnad esineda ainult vasakult paremale või ülevalt alla. Keskmise ja raske taseme sõnarägakstikus võivad sõnad esineda lisaks ka vastupidises suunas (paremalt vasakule või alt üles) ning diagonaalis.
5. Proovib järjest sõna ruudustikku paigutada vastavalt eelnevalt määratud sõna alguspositsioonidele ning sõna suundadele. Esiteks kontrollitakse, et sõna mahuks pikkuse poolest ruudustikku. Seejärel vaadatakse, kas sõna kõikidele tähtedele vastavad ruudustiku positsioonid on sobivad. See tähendab, et igal positsioonil peab kas puuduma juba olemasolev täht või peab olemasolev täht ühtima lisatava sõna vastava tähega. Lihtsa raskusastme korral ei tohi sõnad omavahel kattuda, mistõttu peavad kõik vastavad ruudustiku positsioonid olema täiesti tühjad.
6. Kui sõnale leitakse sobiv alguspositsioon ning suund, siis paigutatakse sõna ruudustikku ning eemaldatakse kandidaatsõnade nimekirjast. Kui proovitakse läbi kõik ruudustiku positsioonid ning sõna suunad, kuid ei õnnestu sõna ruudustikku paigutada, siis eemaldatakse samuti sõna kandidaatsõnade nimekirjast. Jätkatakse järgmiste kandidaatsõnade ruudustikku paigutamist.

Algoritm lõpetab sõnade ruudustikku paigutamise, kui kandidaatsõnade nimekiri on tühi või kui ruudustikku paigutatud sõnade pikkuste summa moodustab vähemalt 90% ruudustiku positsioonide arvust. See kiirendab sõnarägakstiku genereerimist, kuna algoritm ei pea läbi proovima kõiki kandidaatsõnu, vaid lõpetab töö, kui ruudustik on piisavalt täidetud.

Pärast sõnade ruudustikku paigutamist täidab algoritm kõik ülejäänud tühjad positsioonid juhuslike tähtedega. Juhusliku tähe valikul arvestatakse vastava keele tähtede tüüpilist jaotust: mida sagedamini mingi täht keeles esineb, seda suurem on tõenäosus selle valimiseks. Eesti keele puhul põhineb tähtede sageduse mudel Eesti

Keele Instituudi (EKI) tekstikorpuse märgistatistika analüüsil<sup>12</sup>, mille põhjal koostati käesoleva töö jaoks tähtede esinemissagedustabel (vt lisa 2). Inglise keele tähtede jaotuseks on võetud Cornelli Ülikooli 40 000 sõnal põhinev sagedustabel<sup>13</sup>. Saksakeelsete sõnarägistike jaoks kasutatakse Xah Lee koostatud saksa keele tähtede sagedustabelit<sup>14</sup>.

Sõnarägistiku loomiseks leidub ka keerukamaid algoritme, mis suudavad edukamalt etteantud sõnade nimekirja korral kõik sõnad ruudustikku paigutada. Üks alternatiivne meetod on tagurdusmeetod (ingl *backtracking*). Kui käesolevas töös kasutatud algoritm ei eemalda sõnarägistiku genereerimise käigus ruudustikust sõna, kui see on sinna juba paigutatud, siis tagurdusmeetodil põhinev algoritm töötab nii, et kui mingit sõna ei õnnestu ruudustikku paigutada, siis liigub algoritm sammu tagasi ning eemaldab viimasena ruudustikku paigutatud sõna ning proovib seda mingile teisele positsioonile paigutada. See garanteerib, et kui leidub sõnade asetus, mille korral kõik sõnad mahuvad ruudustikku, siis algoritm leiab selle.

Käesolevas rakenduses ei kasutatud tagurdusmeetodil põhinevat algoritmi, sest sõnarägistiku loomisel ei pea kõik kandidaatsõnad ruudustikku mahtuma – eelnevalt kandidaatsõnade nimekirja koostamisel valiti kandidaatsõnasid rohkem, kui neid ruudustikku mahuks, sest siis on võimalik mõni sõna lihtsalt vahele jätta, kui see ruudustikku ei mahu ning ei pea liikuma sammu võrra tagasi ning juba ruudustikku paigutatud sõna eemaldama. Samuti on tagurdusmeetodil põhinevad algoritmid üldiselt väga aeglased, kuna algoritmi ajakulu kasvab eksponentsiaalselt sisendi suuruse kasvul [18]. Seevastu käesolevas töös rakendatud algoritmi ajakulu on lineaarses sõltuvuses kandidaatsõnade nimekirja pikkusest.

## 2.4 Tagaliides

Käesoleva rakenduse tagaliidese eesmärk on hallata sõnarägistike genereerimist ning tagada, et see toimuks kiiresti ja oleks skaleeruv – st et mitu kasutajat saaksid samaaegselt rakendust kasutada ilma, et nad peaksid üksteist ootama. Kuna

---

<sup>12</sup> <https://en.eki.ee/corpus>

<sup>13</sup> <https://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html>

<sup>14</sup> [http://xahlee.info/kbd/german\\_char\\_frequency.html](http://xahlee.info/kbd/german_char_frequency.html)

sõnarägistiku loomine võib sõltuvalt valitud suurusest, raskusastmest ja teemast võtta mitu sekundit, tuleb vältida olukorda, kus ühe kasutaja päring blokeerib teisi. Sellest tulenevalt peab tagaliides toetama paralleeltöötlust.

Tagaliidese loomiseks kasutati Node.js-i<sup>15</sup>. Node.js on avatud lähtekoodiga JavaScripti käituskeskkond (ingl *runtime environment*), mille abil on võimalik luua veebiservereid ja võrgutööriistu. Node.js-i eeliseks on selle võime käsitleda suurt hulka päringuid efektiivselt, ilma iga taotluse jaoks eraldi lõime loomata.

Paralleeltöötlus on realiseeritud kasutades Node.js-i moodulit *worker\_threads*<sup>16</sup>. Antud moodul võimaldab tagaliidesel luua arvutuslikult intensiivsete ülesannete lahendamiseks alamprotsesse, mida serveri protsessor saab paralleelselt põhiprogrammiga jooksutada ning saata ülesande lahendamisel tulemused tagasi põhiprogrammile.

Alamprotsesside loomine sobib ideaalselt sõnarägistike genereerimiseks (algoritm peatükis 2.3.2), kuna protsessile peab sisendiks andma vaid kandidaatsõnade nimekirja ning lisaparameetrid nagu keel ja soovitud raskusaste. Sõnarägistiku genereerimiseks lisatakse iga taotlus järjekorda koos vajalike parameetritega. Tagaliides kontrollib jooksvalt, mitu alamprotsessi on aktiivsed, ning kui neid on vähem kui maksimaalselt lubatud (vaikimisi neli), käivitatakse järgmine järjekorras olev päring. Lubatud protsesside arvu saab seadistada vastavalt serveri jõudlusele, võimaldades süsteemil skaleeruda vastavalt kasutuskoormusele.

Teine arvutuslikult intensiivne ülesanne, mida peab lahendama alamprotsessis, on sisendteema põhjal kandidaatsõnade valimine rägastikku. Kuna selle algoritmi töö eeltingimuseks on wordneti andmestiku lugemine operatiivmällu, mis võtab esiteks aega umbes 20 sekundit ning kasutab teiseks umbes 300MB mälu, siis ei ole mõistlik selle ülesande lahendamiseks iga kord luua uut alamprotsessi. Selle asemel luuakse tagaliidese programmi käivitamisel üks alamprotsess, mis loeb mällu wordneti andmebaasi ning jääb aktiivseks terve programmi elutsükliks. Sarnaselt sõnarägistike genereerimisele, haldab tagaliides ka seda alamprotsessi järjekorra abil.

---

<sup>15</sup> <https://nodejs.org/en/about>

<sup>16</sup> [https://nodejs.org/api/worker\\_threads.html](https://nodejs.org/api/worker_threads.html)

Selliseid alamprotsesse võib luua ka mitu, kuid siis peab iga alamprotsess lugema mällu eraldi koopiat wordneti andmebaasist, kuna alamprotsesside vahel on mäluruumi jagamine väga keeruline. See tähendab, et programmi mälukasutus kasvaks iga uue alamprotsessi kohta 300MB võrra. JavaScriptis on protsesside vahel mälu jagamine võimalik vaid *SharedArrayBuffer*<sup>17</sup> objekti kasutades. Tegemist on lihtsa andmete järjendiga, mis tähendab, et wordneti andmebaasi salvestamiseks kasutatud andmestruktuurid (paisketabel, prefiksipuu) pole vaikumisi toetatud. Paisketabeleid ja puustruktuure on võimalik realiseerida ka üheainsa andmejärjendi – näiteks *SharedArrayBuffer* – abil, nagu on kirjeldatud Jüri Kiho arvutiteaduse õpikus „*Algoritmid ja andmestruktuurid*“ [19]. Selline lahendus on aga väga veaohklik, nõuab keerukaid optimeerimistehnikaid ning käsitsi mäluhaldust. Seetõttu ei kasutatud käesolevas töös *SharedArrayBuffer*-il põhinevat andmete jagamist, vaid piirduti ühe alamprotsessiga, mis haldab wordneti andmebaasi.

Kolmas relatiivselt aeglane ülesanne on suhtlus andmebaasiga, kuna see hõlmab endas kettale kirjutamist ja kettalt lugemist. Node.js toetab asünkroonseid sisend-väljund (ingl *Input-Output, I/O*) operatsioone<sup>18</sup>, mis võimaldab rakenduse tagaliidesel kettale kirjutamise ja kettalt lugemise operatsioonidele vastust oodates samal ajal teisi ülesandeid lahendada. Sellele põhimõttele tugineb PostgreSQL andmebaasiga suhtlemiseks loodud moodul *node-postgres*<sup>19</sup>, mida valminud rakenduses kasutati, aidates tagada, et rakendust saavad kasutada mitmed kasutajad samaaegselt.

## 2.5 Eesliides

Veebirakenduse eesliides loodi JavaScripti raamistiku Vue.js<sup>20</sup> abil – see on avatud lähtekoodiga progressiivne JavaScripti raamistik, mis sobib hästi interaktiivsete üheleheliste rakenduste loomiseks. Vue.js valiti, kuna see on lihtsasti õpitav,

---

<sup>17</sup> [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/SharedArrayBuffer](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/SharedArrayBuffer)

<sup>18</sup> <https://nodejs.org/en/learn/asynchronous-work/overview-of-blocking-vs-non-blocking>

<sup>19</sup> <https://www.npmjs.com/package/pg>

<sup>20</sup> <https://vuejs.org/guide/introduction>

modulaarne ning kasutajasõbralikum ja paindlikum võrreldes teiste tuntud JavaScripti raamistikega nagu Angular või React [20].

Eesliides järgib ühelehelise rakenduse (ingl *Single-Page Application*)<sup>21</sup> põhimõtet: kogu rakenduse HTML-, CSS- ja JavaScript-kood laaditakse veebilehe avamisel korraga ning seejärel muudetakse lehe sisu dünaamiliselt ilma serverilt uusi HTML-dokumente küsimata. See tagab sujuva kasutuskogemuse: kuigi esmane laadimine võib olla aeglasem kui mitmelehelistel rakendustel, toimub navigeerimine edaspidi märgatavalt kiiremini.

Kuna ühe-lehelise rakenduse eesliides koosneb ühest HTML-dokumendist, mille sisu dünaamiliselt muudetakse, siis ei saadeta veebiaadressi muutmisel tagaliidesele päringut uue dokumendi laadimiseks, vaid eesliides peab ise tuvastama veebiaadressi muutumise ning selle põhjal vastava alamlehe kuvama. Samuti ei tuvasta ühelehelise rakenduse korral veebilehitseja eesliidese erinevate alamlehtede vahel navigeerimist ning ei salvesta alamlehti veebilehitseja ajalukku. Navigatsiooni haldamiseks kasutati rakenduses moodulit Vue Router<sup>22</sup>. Rakenduse oleku haldamiseks, näiteks sõnarägastiku lahendamise ajal leitud sõnadel ja veel leidmata sõnadel järje pidamiseks, kasutati moodulit Pinia<sup>23</sup>.

Eesliidese kujundamisel kasutati reageeriva veebidisaini (ingl *responsive web design*) põhimõtet. Reageeriv veebidisain on lähenemisviis, mis seab prioriteediks selle, et veebileht näeks hea välja ning oleks kasutajasõbralik iga suurusega ning igat tüüpi ekraanil (nutitelefon, tahvelarvuti, lauamonitor jne) [21]. Selleks kasutati raamistikku Vuetify<sup>24</sup>. Vuetify on raamistik, mis pakub dünaamilisi Google'i Material Design<sup>25</sup> disaini printsiipidele vastavaid kasutajaliidese komponente. Vuetify võimaldab määrata komponentide käitumise – kas ja kuidas komponenti kuvada – vastavalt ekraani laiusele. Valminud eesliidises kuvatakse nupud suurematel ekraanidel ikooni ja tekstina, kuid väiksematel ekraanidel vaid ikoonina, seejuures kuvatakse selgitav tekst

---

<sup>21</sup> <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

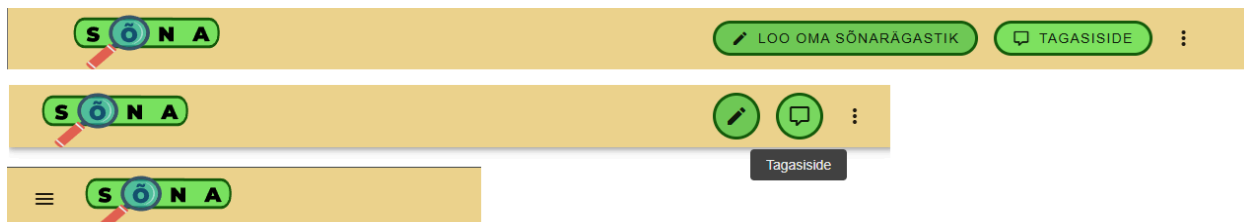
<sup>22</sup> <https://router.vuejs.org/introduction.html>

<sup>23</sup> <https://pinia.vuejs.org/>

<sup>24</sup> <https://vuetifyjs.com/en/>

<sup>25</sup> <https://m3.material.io/>

vaid kursoriga nupu kohale liikudes. Sarnaselt kuvatakse piisavalt laia ekraani korral sõnarägistikku peidetud sõnade nimekiri ruudustikust paremal, kuid kitsamal ekraanil kuvatakse sõnade nimekiri ruudustiku all. Joonisel 7 on näidatud, missugune näeb välja veebilehe navigatsiooniriba suurel, keskmisel ja väikesel ekraanil.



Joonis 7. Navigatsiooniriba erineva suurusega ekraanidel.

Loodud veebirakenduse kasutajaliides on vaikimisi eestikeelne, aga kui veebirakenduse kasutaja veebilehitseja on ingliskeelne, siis määratakse vaikekeeleks inglise keel. Keelt on võimalik muuta navigatsioonimenüüs ning valik salvestatakse veebilehitseja mällu küpsisena. Keelevalik lisatakse ka igale tagaliidesele saadetavale HTTP-päringule Accept-Language päisena ning selle keelevaliku põhjal tagastatakse vigade korral veateated kas eesti või inglise keeles.

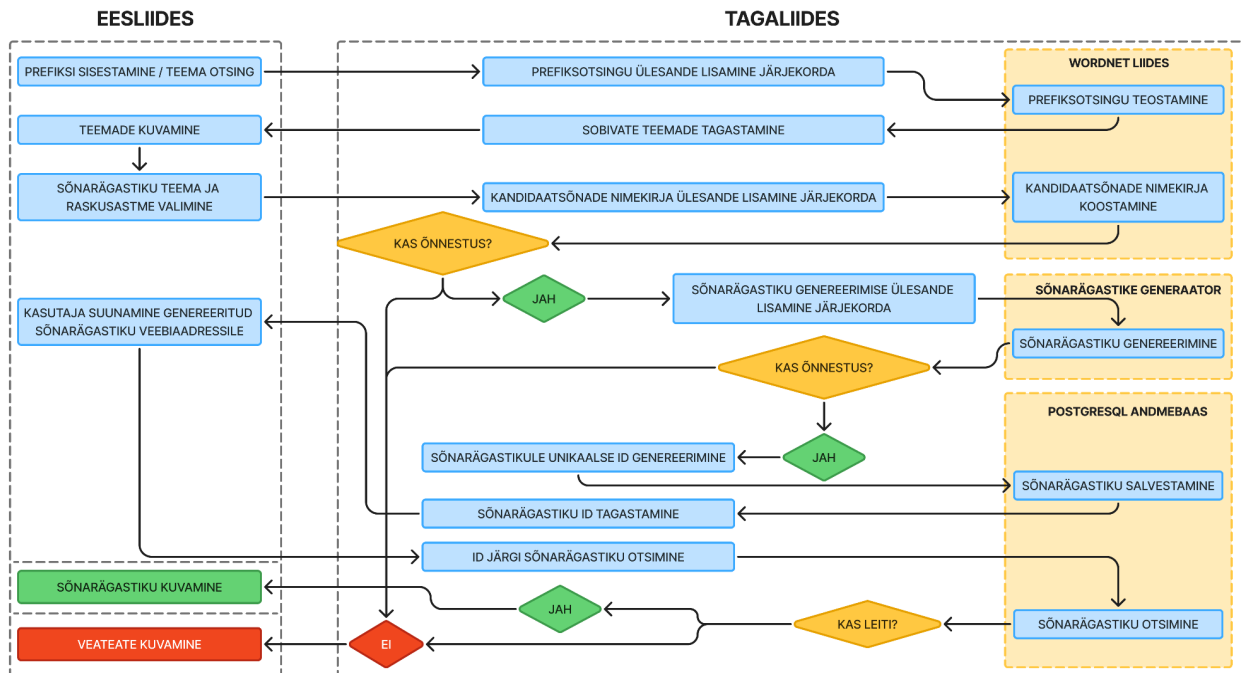
## 2.6 Valminud rakendus

Lõputöö käigus loodud veebirakendus on kättesaadav aadressil <https://www.sonaragastik.ee> ning selle lähtekood on leitav GitHubis: <https://github.com/Tengyyy/word-puzzle>.

Rakenduse avalehel saab automaatselt genereerida sõnarägistikke, sisestades teema ning valides sobiva raskusastme. Lisaks saab määrata sisendkeele ja väljundkeele: sisendkeel määrab, mis keeles kuvatakse vihjed ning väljundkeel määrab, mis keeles on sõnarägistikku peidetud sõnad.

Valminud sõnarägistikku saab veebilehel interaktiivselt lahendada, lohistades sõnadel kas hiirega (arvutis) või sõrmega (puutetundlikul ekraanil). Samuti on võimalik genereeritud sõnarägistik välja printida või salvestada dokumendina. Sisendteemaga

„sport” loodud nädissõnarägistikku on lahendamise vaates näidatud lisan 3 ning dokumendina lisan 4. Rakenduse töövoogu sõnarägistiku loomisel illustreerib joonis 8.



Joonis 8. Sõnarägistiku loomise vooskeem.

Valides veebilehe menüüst „Loo oma sõnarägistik“, avaneb alamleht, kus saab koostada kohandatud sõnarägistikke. Sõnarägistiku raskusastme määramise asemel saab alamlehel täpselt määrata sõnarägistiku laiuse, pikkuse, sõnade kattumise, sõnade suuna ja muu. Lisaks on võimalik sõnarägistikku käsitsi sõnu lisada.

### **3. Tagasiside rakendusele**

Veebirakenduse testimiseks loodi tagasisideküsitlus (vt lisa 5). Testijatel paluti enne testimist lugeda läbi kõik küsimused. Tagasisideküsitlus hõlmas küsimusi rakenduse kasutusmugavuse, kasutajaliidese välimuse ning genereeritud sõnarägistike kvaliteedi kohta.

#### **3.1 Testijate taust**

Tagasisideküsitlusele vastas kümme inimest, kellest seitse olid vanusevahemikus 18-24, kaks vanusevahemikus 35-44 ning üks vastanu oli üle 45 aasta vana. Kõik testijad oskasid inglise keelt ning üheksa neist oskasid ka eesti keelt. Üks testija valdas lisaks saksa keelt. Kõik vastanud olid kas väga tuttavad või mõnevõrra tuttavad sõnarägistiku tüüpi mõistatustega.

#### **3.2 Tagasiside sõnarägistikele**

Sõnarägistike loomise kasulikkust sisendteema alusel hinnati seitsmepunktilisel skaalal järgmiselt: neli testijat andsid 4 punkti, kaks testijat 6 punkti ja neli testijat 7 punkti. Keskmise hinnang oli 6,0 punkti, mis viitab sellele, et veebirakenduse tuumik-funktsionaalsust, seotud sõnade leidmist teema alusel, peeti väga kasulikuks.

Sõnarägistike teemakohasust hindasid testijad samuti skaalal 1–7: üks testija andis 2 punkti, üks 3 punkti, viis testijat 5 punkti ja kolm testijat 7 punkti. Keskmiseks kujunes 5,1 punkti, mis näitab valdavalt positiivset hinnangut, kuid ka mõningast varieeruvust. See näitab, et sõnarägistiku vastavus sisendteemale sõltub suurel määral konkreetse teema olemusest: mõnel sõnal on wordnetis rohkem semantilisi seoseid kui teisel.

Raskusastmete erinevuse arusaadavust hindas üks testija 3 punktiga, üks 4 punktiga, kolm testijat 5 punktiga, üks testija 6 punktiga ning neli testijat 7 punktiga. Keskmise hinnang oli 5,6 punkti, mis näitab, et raskusastmed olid piisavalt eristuvad. Üheksa testijat proovisid lahendada raskemaid sõnarägistikke, kus vihjeteks on otsitavate sõnade asemel nende definitsioonid. Kahe testija sõnul olid need sõnarägistikud väga

põnevad ja väljakutsuvad, kuus testijat pidasid neid keerulisteks, kuid lahendatavateks, ning üks testija leidis, et need olid liiga rasked.

Erineva sisend- ja väljundkeelega sõnarägistikke proovisid kaheksa testijat. Vastavust keelte vahel hindas üks testija 2 punktiga, neli testijat 5 punktiga ja kolm testijat 7 punktiga. Keskmine hinnang oli 5,4 punkti. Üks testija märkis, et saksa keelega ei töötanud sõnarägistiku genereerimine alati ootuspäraselt.

### **3.3 Tagasiside kasutajaliidesele**

Kasutajaliidesele anti valdavalt väga positiivne hinnang. Rakenduse kasutusmugavust hinnati skaalal 1–7: üks testija andis 5 punkti, viis testijat 6 punkti ja neli testijat 7 punkti. Kasutajaliidese välimust (värvivalik, paigutus) hindasid kõik testijat vähemalt 6 punktiga.

Tehnilisi probleeme ei esinenud kuuel testijal, neljal esines väiksemaid raskusi. Ühel testijal oli keeruline väikese ekraaniga seadmel sõnu lohistada, teist häiris, et rakendus ei salvestanud eelnevaid sätteid (nt raskusaste ja keeled), mistõttu tuli need igal korral uuesti sisestada.

Kohandatud sõnarägistike loomist menüüvaliku „Loo oma sõnarägistik” alt proovis kuus testijat, kes leidsid, et see funktsioon on lihtsasti kasutatav. Kõige enam muudeti sõnade suunda ning sõnarägistiku mõõtmeid.

### **3.4 Üldine tagasiside**

Rakenduse meelelahutuslikku väärtust hindas üks testija 4 punktiga, kolm testijat 5 punktiga, kolm testijat 6 punktiga ja kolm testijat 7 punktiga. Keskmine hinnang oli 5,8 punkti 7-st.

Keeleõppelist väärtust hinnati pisut madalamalt: üks testija andis 2 punkti, üks 3 punkti, üks 4 punkti, kolm testijat 5 punkti, kolm testijat 6 punkti ning üks testija 7 punkti. Keskmine tulemus oli 4,9 punkti.

Kümnest testijast üheksa vastasid, et nad kas võib-olla või kindlasti soovitaksid rakendust teistele. Positiivselt toodi enim esile rakenduse lihtsust ja kasutusmugavust ning selle potentsiaali aidata sõnavara laiendada. Tagasiside oli üldjoontes väga positiivne, mistõttu võib eeldada, et rakendusel on potentsiaali kujuneda edukaks keeleõppeliseks mänguks, kui seda edasi arendada ja täiustada.

## **4. Edasiarenduse võimalused**

Käesolevas peatükis on esitatud ideid, kuidas rakendust edasi arendada. Mõtted on jaotatud kaheks: testijate soovitusel ning arendustöö käigus esile kerkinud ideed.

### **4.1 Testijate soovitusel**

Küsimustikule vastajad tõid esile vajaduse suurema seadistatavuse järele kohandatud sõnarägistiku loomise alamlehel. Näiteks üks testija soovis võimalust sõnu käsitsi ümber paigutada. Teine testija soovis, et sõnarägistiku genereerimisel oleks võimalik täpselt määrata, mitu sõna sinna lisatakse. Mõlemad ettepanekud võimaldaksid kasutajatel koostada veel enam soovidele vastavaid sõnarägistikke.

Mitmed testijad pakkusid välja ka ideid rakenduse mängulisemaks muutmiseks. Näiteks soovitati lisada võimalus luua juhusliku teemaga või täiesti teemavabasis sõnarägistikke suvaliste sõnadega. Need pakusid rohkem vaheldust ning võiksid muuta lahendamise lõbusamaks, kuigi keeleõppeline väärtus võib neil olla madalam. Samuti soovitati lisada ajalimiit ja punktiarvestus, kus kasutajad teeniksid punkte sõltuvalt sellest, kui kiiresti nad kõik sõnad leiavad. Selline funktsionaalsus muudaks rakenduse kaasahaaravamaks ja sobiks hästi mängulise õppeviisi toetamiseks.

### **4.2 Rakenduse arendamise käigus tekkinud ideed**

Üheks arenduse käigus välja kujunenud edasiarendusvõimaluseks on sõnade valimine raskusastme alusel. Lõplikus versioonis mõjutab raskusaste küll sõnarägistiku suurust, sõnade suundasid ja vihjetüüpi (sõna või definitsioon), kuid mitte sõnade sagedust ega tuntust. See tähendab, et ka lihtsama raskusastmega sõnarägistik võib sisaldada haruldasi sõnu, mida enamik kasutajaid ei tunne. Sõnade sageduse põhjal liigitamine ning sellega arvestamine raskusastme määramisel võimaldaks pakkuda paremini kasutaja tasemele vastavaid väljakutseid ning suurendaks rakenduse väärtust keeleõppe vahendina.

Teiseks võimaluseks rakenduse haridusliku potentsiaali suurendamiseks on erinevat tüüpi nimede (nt isikute, kohtade, organisatsioonide) kaasamine sõnarägistikesse.

Rakenduses kasutatud wordneti andmebaasid keskenduvad üldjuhul tavakeele sõnavarale, mistõttu sobivad praegused sõnarägastikud eriti hästi keeleõppeks. Samas saab neid kasutada ka meelelahutuseks ja teadmiste avardamiseks.

Kui andmestikku laiendada nimede ja muude spetsiifiliste terminitega ning siduda need semantiliste suhete kaudu olemasolevate sõnadega, oleks võimalik koostada sõnarägastikke, mida saab kasutada näiteks geograafia või ajaloo õppimiseks. Neid nimesid saab automaatselt tuvastada ja siduda, analüüsides Vikipeedia artikleid tehisintellekti abil. Näiteks analüüsides artiklit Microsofti kohta, selgub, et tegemist on tehnoloogiaettevõttega – seega võiks nime „Microsoft“ siduda sõnadega „ettevõte“ ja „tehnoloogia“ ning kaasata see sõnarägastike loomisesse.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua veebirakendus, mis võimaldab automaatselt genereerida erinevat tüüpi sõnarägistikke, kus kõik sõnad on seotud keskse teemaga. Töö käigus uuriti keelemängude kasutamist hariduses ja keeleõppes ning analüüsi olemasolevaid veebipõhiseid sõnarägistike generaatoreid ja nende kitsaskohti.

Töö tulemusena valmis veebirakendus, mis on kättesaadav aadressil <https://www.sonaragastik.ee>. Rakendus võimaldab luua erineva raskusastmega sõnarägistikke, kus kõik peidetud sõnad on seotud valitud teemaga. Lisaks saab koostada alternatiivseid versioone, kus vihjed ei kattu otseselt sõnadega, vaid on näiteks nende definitsioonid või vasted teistes keeltes. Rakendus toetab sõnarägistike loomist eesti, inglise ja saksa keeles, kasutades vastavate keelte wordneti andmebaase. Genereeritud sõnarägistikke saab lahendada veebilehel või printida välja.

Teadaovalt on valminud rakenduse näol tegemist ainsa eesti keelt toetava sõnarägistike generaatoriga, mis võimaldab automaatselt luua teemaga seotud sõnarägistikke. Samuti eristub loodud rakendus teistest sarnastest lahendustest, toetades kakskeelsete sõnarägistike automaatset genereerimist.

Rakendust testis kümme inimest, kes andsid hinnangu selle kasutusmugavusele, sõnarägistike kvaliteedile ning hariduslikule ja meelelahutuslikule väärtusele. Tagasiside oli valdavalt positiivne: meelelahutuslik väärtus sai keskmiseks hindeks 5,8 ning keeleõppeline väärtus 4,9 seitsmepallisel skaalal. Saadud tagasiside põhjal võib järeldada, et töö eesmärgid on täidetud ning loodud rakendus toetab edukalt keeleõpet. Samas pakub rakendus veel mitmeid edasiarendusvõimalusi, näiteks sõnade valimise täiendamine vastavalt raskusastmele.

## Viidatud kirjandus

- [1] Fitria, T. N. The Effectiveness of Word Search Puzzles Game in Improving Student's Vocabulary: A Systematic Literature Review. *Pioneer: Journal of Language and Literature*, 2023, vol 15, 50–67.  
<https://doi.org/10.36841/pioneer.v15i1.2766>
- [2] Kumar, P. K., Vairavan, C. The Impact of Gamification on Motivation and Retention in Language Learning: An Experimental Study Using a Gamified Language Learning Application. *INTI Journal*, 2024, vol 44.  
<http://dx.doi.org/10.61453/INTIj.202444>
- [3] Yip, F. W. M., Kwan, A. C. M. Online vocabulary games as a tool for teaching and learning English vocabulary. *Educational Media International*, 2006, vol 43, 233–249. <https://doi.org/10.1080/09523980600641445>
- [4] Yaccon, N. S., Yunus, M. M. Language Games in Teaching and Learning English Grammar: A Literature Review. *Arab World English Journal*, 2019, vol 10, 209–217. <https://dx.doi.org/10.2139/ssrn.3367576>
- [5] Heidari-Shahreza, M. A. Pedagogy of play: insights from playful learning for language learning. *Discover Education*, 2024, vol 3.  
<https://doi.org/10.1007/s44217-024-00250-3>
- [6] Tutt, P. 5 Playful Strategies That Reduce Language Learning Anxiety. *Edutopia*. 2023.  
<https://www.edutopia.org/article/playful-learning-strategies-that-reduce-language-learning-anxiety/> (15.05.2025).
- [7] Franco, JC. Word Search: What Is It?, Objective, Purpose, And More!.  
<https://www.gamesver.com/word-search-what-is-it-objective-purpose-and-more> (15.12.2024).

- [8] Beelders, T. Visual search patterns for multilingual word search puzzles, a pilot study. *Journal of Eye Movement Research*, 2023, vol 16, 1–12.  
<https://doi.org/10.16910/jemr.16.1.6>
- [9] Gholami, J., Khezrlou, S. Semantic and Thematic List Learning of Second Language Vocabulary. *CATESOL Journal*, 2014, vol 25, 151–162.  
<https://eric.ed.gov/?id=EJ1111893>
- [10] Amirian, S. M. R., Momeni, S. Definition-based Versus Contextualized Vocabulary Learning. *Theory and Practice in Language Studies*, 2012, vol 2, 2302–2307. <https://doi.org/10.4304/tpls.2.11.2302-2307>
- [11] Eesti Wordnet. <https://keeleressursid.ee/et/265-eesti-wordnet> (15.12.2024).
- [12] Bryant, R. E., O'Hallaron, D. R. *Computer Systems: A Programmer's Perspective* (3rd ed.). Pearson, 2016, 617–639.
- [13] Gray, J., Reuter, A. *Transaction processing: Concepts and techniques*. Morgan Kaufmann Publishers Inc., 1992, 159–235.
- [14] PostgreSQL: Documentation: 17: 3.4. Transactions.  
<https://www.postgresql.org/docs/current/tutorial-transactions.html> (15.05.2025).
- [15] Sedgewick, R. Wayne, K. *Algorithms* (4th ed.). Pearson, 2011, 730–757.
- [16] Sullivan, D. *What is Google PageRank? A Guide for Searchers & Webmasters*. 2007.  
<https://searchengineland.com/what-is-google-pagerank-a-guide-for-searchers-webmasters-11068> (12.05.2025).
- [17] Buck, J. *Generating Word Search Puzzles*.  
<https://weblog.jamisbuck.org/2015/9/26/generating-word-search-puzzles.html> (12.05.2025).

- [18] GeeksforGeeks. Introduction to Backtracking.  
<https://www.geeksforgeeks.org/introduction-to-backtracking-2/> (12.05.2025).
- [19] Kiho, J. Algoritmid ja andmestruktuurid. Tartu: TÜ Kirjastus, 2003, 27–48.  
<https://dspace.ut.ee/server/api/core/bitstreams/9b9bf761-f89d-4eed-ae0-ebf0aedaa721/content> (12.05.2025).
- [20] Breedis, R. JavaScripti kasutajaliidese raamistike võrdlus. TÜ arvutiteaduse instituudi bakalaureusetöö. 2021.  
<https://dspace.ut.ee/server/api/core/bitstreams/c655ae35-e7ee-4fcc-a6cd-ea1d5b98fb48/content> (12.05.2025).
- [21] Mozilla Developer Network. Responsive Design.  
[https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Core/CSS\\_Layout/Responsive\\_Design](https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_Layout/Responsive_Design) (12.05.2025).

## Lisad

### I. Eesti Wordneti LMF XML-formaadis faili näidis

```
<LexicalEntry>
  <Lemma writtenForm="mootorsõiduk" />
  <Sense synset="estwn-et-2211-n" />
</LexicalEntry>
<LexicalEntry>
  <Lemma writtenForm="auto" />
  <Sense synset="estwn-et-312-n">
    <Example>Hommikul sõitis kumbki oma autoga tööle.</Example>
  </Sense>
</LexicalEntry>
<Synset id="estwn-et-2211-n" ili="">
  <Definition>ratastega iseliikuv liiklusvahend</Definition>
  <SynsetRelation relType="hyponym" target="estwn-et-312-n" />
</Synset>
<Synset id="estwn-et-312-n" ili="i51496">
  <Definition>mootorveok või -sõiduk veoste või inimeste
transportimiseks maanteedel või tänavail</Definition>
  <SynsetRelation relType="hypernym" target="estwn-et-2211-n" />
</Synset>
```

## II. Eesti keele tähtede jaotus

A: 12,27%	J: 1,67%
E: 10,65%	H: 1,66%
I: 9,91%	Ä: 1,29%
S: 8,75%	Õ: 1,1%
T: 7,35%	B: 0,93%
L: 6,25%	Ü: 0,75%
U: 5,67%	Ö: 0,30%
N: 5,11%	F: 0,20%
K: 4,62%	C: 0,20%
O: 3,95%	Y: 0,06%
D: 3,83%	W: 0,05%
M: 3,69%	Z: 0,04%
R: 3,52%	X: 0,02%
V: 2,41%	Š: 0,01%
P: 1,90%	Q: 0,01%
G: 1,86%	Ž: 0,01%

### III. Genereeritud sõnarägistik teemal „Sport”

**Sport**
PRINDI MÄNG

K	T	D	E	S	S	T	T	O	D	A	V	I	S	E	atleetika	mootorisport
O	D	A	K	L	S	R	A	L	M	O	E	E	S	K	džuudo	näomask
K	A	M	A	P	E	O	L	Ü	O	D	K	N	U	Õ	ekstreemsport	oda
I	T	J	L	A	N	P	I	M	O	U	S	I	L	A	fitness	odavise
T	L	Ä	A	L	T	S	S	P	T	U	T	M	T	J	jääpall	olümpialane
S	E	Ä	S	L	I	A	P	I	O	Ž	R	U	S	S	kalasport	orienteerumine
U	E	P	P	I	F	K	O	A	R	D	E	R	I	K	kelgusport	pallima
Õ	T	A	O	M	K	T	R	L	I	M	E	E	Õ	O	kergejõustik	raadiosport
J	I	L	R	A	I	A	T	A	S	A	M	E	V	O	male	sportmäng
E	K	L	T	L	N	M	A	N	P	L	S	T	E	J	matkasport	talisport
G	A	V	E	D	A	J	A	E	O	E	P	N	M	I	miilijooksja	vedaja
R	A	A	D	I	O	S	P	O	R	T	O	E	T	L	mitmevõistlus	
E	K	S	A	M	O	Ä	N	U	T	G	R	I	I	I		
K	K	S	P	O	R	T	M	Ä	N	G	T	R	M	I		
T	R	O	P	S	U	G	L	E	K	N	J	O	L	M		

## IV. Dokumendina salvestatud sõnarägistik

### Sport

K	T	D	E	S	S	T	T	O	D	A	V	I	S	E
O	D	A	K	L	S	R	A	L	M	O	E	E	S	K
K	A	M	A	P	E	O	L	Ü	O	D	K	N	U	Õ
I	T	J	L	A	N	P	I	M	O	U	S	I	L	A
T	L	Ä	A	L	T	S	S	P	T	U	T	M	T	J
S	E	Ä	S	L	I	A	P	I	O	Ž	R	U	S	S
U	E	P	P	I	F	K	O	A	R	D	E	R	I	K
Õ	T	A	O	M	K	T	R	L	I	M	E	E	Õ	O
J	I	L	R	A	I	A	T	A	S	A	M	E	V	O
E	K	L	T	L	N	M	A	N	P	L	S	T	E	J
G	A	V	E	D	A	J	A	E	O	E	P	N	M	I
R	A	A	D	I	O	S	P	O	R	T	O	E	T	L
E	K	S	A	M	O	Ä	N	U	T	G	R	I	I	I
K	K	S	P	O	R	T	M	Ä	N	G	T	R	M	I
T	R	O	P	S	U	G	L	E	K	N	J	O	L	M

atleetika	kelgusport	mootorisport	pallima
džuudo	kergejõustik	näomask	raadiosport
ekstreemsport	male	oda	sportmäng
fitness	matkasport	odavise	talisport
jääpall	miiljooksja	olümpialane	vedaja
kalasport	mitmevõistlus	orienteerumine	

## V. Eestikeelne rakenduse testimise küsimustik

### Veebirakenduse Sõnarägastike generaator testimise küsitlus

Käesoleva küsimustiku on loonud Hans Martin Tomingas informaatika bakalaureusetöö "Eesti keeleressurssidel põhinev sõnarägastike generaator" testimiseks.

Küsitlus on anonüümne ja vabatahtlik. Täites küsimustiku ning saates ära vastused, annab osaleja nõusoleku kasutada vastuseid eelmainitud töös. Küsitlusele vastamine võtab umbes 5 minutit, millele lisandub aeg veebilehe testimiseks.

Sõnarägastike generaator on veebirakendus, mis võimaldab automaatselt luua sisendteema põhjal erinevates keeltes erineva raskusastmega sõnarägastikke.

Palun lugege enne veebirakenduse testimist üle allolevad küsimused, et teada, millistele veebirakenduse aspektidele tähelepanu pöörata.

Veebirakendus on leitav aadressilt <https://sonaragastik.ee>.

hansmartintomingas@gmail.com [Switch account](#)



Not shared

\* Indicates required question

#### OSA 1: Taustateave

Millisesse vanuserühma sa kuulud? \*

- Alla 18
- 18-24
- 25-34
- 35-44
- 45+



**Kas proovisid lahendada sõnarägistikke kõrgeima raskusastmega, kus vihjeteks \* olid sõnade asemel definitsioonid? Kui jah, siis kuidas hindad seda tüüpi sõnarägistikke?**

- Ei proovinud
- Väga põnevad ja väljakutsuvad
- Väljakutsuvad, kuid lahendatavad
- Liiga rasked
- Segadust tekitavad
- Ei meeldinud

**Kui proovisid sõnarägistikku välja trükkida või PDF-failina salvestada, siis kuidas hindad salvestatud või väljaprintitud sõnarägistiku kvaliteeti (nt paigutus, loetavus, üldine väljanägemine)**

- 1      2      3      4      5      6      7
- Väga halb                                                Väga hea

### **OSA 3: Mitmekeelsed sõnarägistikud**

**Kui proovisid genereerida sõnarägistikke erinevate sisend- ja väljundkeeltega, siis kuidas hindad sõnade vastavust kahe keele vahel?**

- 1      2      3      4      5      6      7
- Väga halb                                                Väga hea

### **OSA 4: Kohandatud sõnarägistike loomine**

Selles osas on küsitud tagasisidet kohandatud sõnarägistike loomise alamlehe kohta (menüüs valik "Loo oma sõnarägistik")

**Kas proovisid luua kohandatud sõnarägistikke menüü valiku "Loo oma sõnarägistik" alt? \***

- Jah
- Ei

**Kui lihtne oli sellel alamleheküljel sõnarägistikke luua?**

- 1      2      3      4      5      6      7
- Väga raske                                Väga lihtne

**Milliseid alljärgnevaid võimalusi kasutasid sõnarägistiku loomisel?**

- Sõnarägistiku mõõtmete muutmine
- Sõnade suuna määramine (diagonaalsuund, vastupidine suund)
- Sõnade kattumise määramine
- Oma sõnaloendi sisestamine

**Milliseid seadistusvõimalusi sooviksid lisaks näha sõnarägistiku loomisel?**

Your answer \_\_\_\_\_

**OSA 5: Kasutuskogemus**

**Kuidas töötas rakendus sinu seadmes? \***

- Täiuslikult
- Väiksemate probleemidega
- Suurte probleemidega
- Ei saanud üldse kasutada

**Kui kogesid rakenduse kasutamisel probleeme, siis milliseid?**

Your answer

---

**Kuidas hindad veebirakenduse kasutajamugavust? \***

	1	2	3	4	5	6	7	
Väga halb	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga hea

**Kuidas hindad veebirakenduse välimust (värvivalik, elementide paigutus jne)? \***

	1	2	3	4	5	6	7	
Väga halb	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga hea

**Kuidas hindad selle rakenduse meelelahutuslikku väärtust? \***

	1	2	3	4	5	6	7	
Väga igav	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga meelelahutuslik

**Kuidas hindad selle rakenduse keeleõppelist väärtust? \***

	1	2	3	4	5	6	7	
Üldse mitte kasulik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Väga kasulik

**Kas soovitaksid seda rakendust teistele? \***

- Jah
- Võib-olla
- Ei

**Mis sulle kõige rohkem meeldis selle rakenduse juures?**

Your answer

---

**Kas sul on ettepanekuid või ideid, kuidas seda rakendust paremaks muuta?**

Your answer

---

# Litsents

## Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Hans Martin Tomingas

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Eesti keeleressurssidel põhinev sõnarägastike generaator**, mille juhendaja on Sven Aller, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Hans Martin Tomingas*

**15.05.2025**