

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Dmitri Kurtukov 232993IAIB

Ruslan Melnikov 233253IAIB

# **Adaptiivse õppeplatvormi prototüübi arendus**

Bakalaureusetöö

Juhendaja: Siim Rebane

BSc

Tallinn 2026

## **Autorideklaratsioon**

Kinnitame, et oleme 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.

Autorid: Dmitri Kurtukov ja Ruslan Melnikov

31.05.2026

## Lühikokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua funktsionaalne prototüüp adaptiivsest matemaatika õppeplatvormist, mis võimaldab õppijal lahendada harjutusülesandeid, kontrollib vastuseid automaatselt ning kohandab ülesannete raskusastet õppija seniste tulemuste põhjal. Lisaks pidi lahendus toetama õpetaja tööd, võimaldades hallata õppegruppe, aktiveerida õppeteemasid ja jälgida õppijate tulemusi.

Töö käigus anti ülevaade e-õppe platvormidest, personaliseeritud õppesüsteemidest ja adaptiivse õppimise algoritmidest. Prototüübi jaoks määratleti nõuded, kavandati süsteemi arhitektuur, andmebaasi mudel, API ja adaptiivne algoritm ning realiseeriti veebirakendus õppija ja õpetaja rollidele.

Prototüübi taustsüsteem loodi FastAPI raamistikuga, kasutajaliides Reacti ja TypeScriptiga ning andmete talletamiseks kasutati PostgreSQL-i. Adaptiivne mehhanism põhineb Elo-laadsel reitingul ja reeglipõhisel raskusastme muutmisel. Lahendust testiti tehniliselt ning katsetati kolmes 11. klassi matemaatikatunnis. Kasutatavuse hindamisele vastas 42 õppijat ja 3 õpetajat.

Testimise tulemused näitasid, et prototüübi põhifunktsionaalsus toimib ning kasutajad hindasid platvormi üldist arusaadavust positiivselt. Enamik õppijaid leidis, et ülesannete raskusaste muutus loogiliselt või pigem loogiliselt. Töö tulemusena valmis toimiv prototüüp, mis demonstreerib adaptiivse matemaatika harjutamise põhivoogu ning loob aluse süsteemi edasiseks arendamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 8 peatükki, 3 joonist, 9 tabelit.

## **Abstract**

### **Development of an Adaptive Learning Platform Prototype**

The aim of this bachelor's thesis was to create a functional prototype of an adaptive mathematics learning platform that enables students to solve practice tasks, checks answers automatically, and adjusts task difficulty based on the student's previous performance. In addition, the solution was intended to support teachers by allowing them to manage study groups, activate learning topics, and monitor student progress.

The thesis gives an overview of e-learning platforms, personalized learning, adaptive learning systems, and algorithms used in adaptive learning. Based on this background, the requirements of the prototype were defined, and the system architecture, database model, API, and adaptive algorithm were designed. The practical result of the thesis is a web application with separate student and teacher views.

The backend of the prototype was developed with FastAPI, the frontend with React and TypeScript, and PostgreSQL was used for data storage. The adaptive mechanism is based on an Elo-like rating and rule-based difficulty adjustment. The prototype was technically tested and evaluated in three 11th grade mathematics lessons. The usability evaluation included responses from 42 students and 3 teachers.

The results showed that the main functionality of the prototype works as intended and that users evaluated the overall clarity of the platform positively. Most students stated that the change in task difficulty was logical or rather logical. As a result of the thesis, a working prototype was created that demonstrates the basic flow of adaptive mathematics practice and provides a foundation for further development.

The thesis is in Estonian and contains 46 pages of text, 8 chapters, 3 figures, 9 tables.

## Lühendite ja mõistete sõnastik

API	Rakendusliides ( <i>Application Programming Interface</i> ), mille kaudu suhtlevad kasutajaliides ja taustsüsteem.
REST	Arhitektuuriline lähenemine veebipõhiste API-de loomiseks, kus andmeid ja toiminguid käsitletakse ressurssidena.
Backend	Taustsüsteem ehk rakenduse serveripoolne osa, mis töötleb päringuid, suhtleb andmebaasiga ja rakendab ärioloogikat.
Frontend	Kasutajaliides ehk rakenduse brauseris töötav osa, millega kasutaja vahetult suhtleb.
FastAPI	Pythoni veebiraamistik, mida kasutati prototüübi taustsüsteemi loomiseks.
React	JavaScripti teek komponentpõhise kasutajaliidese loomiseks.
TypeScript	JavaScriptil põhinev programmeerimiskeel, mis lisab staatilise tüübikontrolli.
PostgreSQL	Relatsiooniline andmebaasisüsteem, mida kasutati prototüübi andmete talletamiseks.
JSONB	PostgreSQL-i andmetüüp, mis võimaldab salvestada JSON-struktuuris andmeid binaarselt optimeeritud kujul.
Docker	Konteineriseerimise platvorm, mida kasutati rakenduse osade ühtseks käivitamiseks.
Docker Compose	Tööriist mitmest konteinerist koosneva rakenduse kirjeldamiseks ja käivitamiseks.
Elo-laadne reiting	Reitingupõhine lähenemine õppija hinnangulise taseme uuendamiseks vastavalt õigetele ja valedele vastustele.
IRT	Ülesande vastusteooria ( <i>Item Response Theory</i> ), mille abil saab modelleerida õppija võimekuse ja ülesande raskuse seost.
Adaptiivne õpe	Õppimisviis, mille puhul süsteem kohandab õppesisu või ülesannete raskusastet õppija tegevuse ja tulemuste põhjal.
Õppija mudel	Süsteemis hoitav info õppija tulemuste, taseme ja edenemise kohta, mille põhjal tehakse otsuseid järgmiste ülesannete valimisel.
RBAC	Rollipõhine ligipääsukontroll ( <i>Role-Based Access Control</i> ), mille puhul kasutaja õigused sõltuvad tema rollist süsteemis.
HTTP-only küpsis	Küpsis, millele brauseri JavaScript ei pääse ligi ning mida saab kasutada tundlikumate autentimisandmete kaitsmiseks.

Värskendustoken      Token, mille abil saab kasutaja sessiooni pikendada ja väljastada uue ligipääsutokeni.

## Sisukord

1	Sissejuhatus.....	11
2	Valdkonna ülevaade .....	14
2.1	E-õppe platvormid ja personaliseeritud õpe .....	14
2.2	Adaptiivsed õppesüsteemid .....	15
2.3	Adaptiivse õppimise algoritmid .....	16
2.4	Olemasolevate lahenduste võrdlus.....	18
3	Prototüübi lähtetingimused, nõuded ja metoodika .....	20
3.1	Lähtetingimused ja sihtrühm .....	20
3.2	Kasutajarollid.....	21
3.3	Funktsionaalsed nõuded.....	22
3.4	Mittefunktsionaalsed nõuded.....	23
3.5	Arendus- ja valideerimismetoodika .....	24
3.6	Tehnoloogiliste valikute põhjendus.....	25
4	Süsteemi kavandamine .....	27
4.1	Süsteemi üldarhitektuur .....	27
4.2	Kasutusvood.....	28
4.3	Andmebaasi mudel.....	29
4.4	API kavandamine .....	32
4.5	Adaptiivse algoritmi kavandamine .....	34
5	Prototüübi realiseerimine .....	37
5.1	Taustsüsteemi realiseerimine .....	37
5.2	Kasutajaliidese realiseerimine .....	38
5.3	Õppija vaate realiseerimine .....	39
5.4	Õpetaja haldusvaate realiseerimine.....	40
5.5	Adaptiivse mootori realiseerimine.....	40
5.6	Rakenduse käivitamine ja konteineriseerimine.....	41
6	Testimine ja tulemuste analüüs .....	43

6.1	Testimise eesmärk ja metoodika .....	43
6.2	Funktsionaalne testimine.....	44
6.3	API testimine.....	44
6.4	Adaptiivse algoritmi testimine .....	45
6.5	Kasutatavuse hindamine .....	46
6.6	Tulemuste analüüs.....	49
6.7	Lahenduse piirangud .....	50
7	Edasiarenduse võimalused.....	52
8	Kokkuvõte.....	55
	Kasutatud kirjandus .....	57
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	59
	Lisa 2 – Õppija kasutatavuse küsimustik .....	60
	Lisa 3 – Õpetaja kasutatavuse küsimustik .....	62

## **Jooniste loetelu**

Joonis 1. Õppija ülesande lahendamise voog. ....	29
Joonis 2. Andmebaasi skeem. ....	30
Joonis 3. Adaptiivse taseme uuendamise voog. ....	35

## Tabelite loetelu

Tabel 1. Olemasolevate lahenduste võrdlus loodava prototüübiga.....	18
Tabel 2. Prototüübi funktsionaalsed nõuded. ....	22
Tabel 3. Prototüübi mittefunktsionaalsed nõuded. ....	23
Tabel 4. Andmebaasi põhilised olemid.....	31
Tabel 5. API peamised rühmad. ....	33
Tabel 6. Adaptiivse mehhanismi testimisel kontrollitud olukorrad. ....	45
Tabel 7. Õppijate hinnangud prototüübile.....	47
Tabel 8. Õppijate hinnang raskusastme muutumise loogilisusele.....	48
Tabel 9. Õpetajate hinnangute koondtulemused.....	49

# 1 Sissejuhatus

E-õppe keskkonnad on muutunud õppimise tavaliseks osaks. Euroopa Liidu digihariduse tegevuskava aastateks 2021–2027 rõhutab, et digitaalsed lahendused peavad toetama kvaliteetset ja kättesaadavat haridust [1]. Samas ei lahenda õppe viimine digitaalsesse keskkonda automaatselt üht keskset probleemi: õppijad ei omanda teemasid sama kiirusega ega vaja alati sama tüüpi harjutamist. Kui keskkond pakub kõigile õppijatele samu ülesandeid samas järjekorras, jääb osa digitaalse õppe võimalustest kasutamata.

Õppimisel on sobiva raskusastmega harjutusülesannetel oluline roll, sest õppija areng sõltub järjepidevast harjutamisest, tagasisidest ja võimalusest keskenduda just neile teemadele, mis vajavad täiendavat kinnistamist. Kui ülesanded on õppija jaoks liiga lihtsad, võib motivatsioon väheneda; kui need on liiga keerulised, võib õppija jääda teemast maha ega pruugi mõista, milliseid oskusi tuleb parandada. Seetõttu on aktuaalne arendada õpisüsteeme, mis koguvad õppija soorituse kohta andmeid ning kasutavad neid järgmiste ülesannete raskusastme kohandamiseks.

Adaptiivne õppeplatvorm ei käsitle õppijat staatilise kasutajana, vaid uuendab tema taseme hinnangut õppimise käigus. OECD digihariduse ülevaates tuuakse esile, et digitaalsete õppevahendite puhul on oluline nende kasulikkus, usaldusväärsus ja sobivus tegelikku õppeprotsessi [2]. Varasemad uuringud on näidanud, et veebipõhine diagnostiline hindamine ja adaptiivne harjutamine võivad toetada personaliseeritud õpet ning anda õpetajale täpsemat tagasisidet õppijate edenemise kohta [3, 4].

Bakalaureusetöös käsitletav probleem seisneb selles, kuidas kavandada ja realiseerida veebipõhine adaptiivse õppeplatvormi prototüüp, mis sobib matemaatika harjutusülesannete lahendamiseks. Lahendus peab võimaldama õppijal lahendada süsteemi pakutavaid ülesandeid, kontrollima vastuseid automaatselt ning muutma ülesannete raskusastet õppija seniste tulemuste põhjal. Lisaks peab süsteem toetama õpetaja tööd, võimaldades hallata õppegrupe, aktiveerida teemasid ja jälgida õppijate tulemusi.

Töö eesmärk on luua funktsionaalne prototüüp adaptiivsest matemaatika õppeplatvormist. Prototüüp keskendub gümnaasiumiastme matemaatikale ning toetab töö praeguses mahus trigonomeetria ning kombinatoorika ja tõenäosuse teemasid. Õppija vaates on põhirõhk ülesannete lahendamisel ja oma edenemise jälgimisel. Õpetaja vaates on põhirõhk gruppide, õpeteemade ja tulemuste ülevaate haldamisel.

Eesmärgi saavutamiseks püstitati järgmised alamülesanded:

1. Anda ülevaade e-õppe platvormidest, personaliseeritud õppes ja adaptiivsetest õppesüsteemidest.
2. Kirjeldada adaptiivses õppes kasutatavaid algoritmilisi lähenemisi ning võrrelda olemasolevaid lahendusi.
3. Määratleda prototüübi sihtrühm, kasutajarollid ning funktsionaalsed ja mittefunktsionaalsed nõuded.
4. Kavandada süsteemi arhitektuur, andmebaasi mudel, API ja adaptiivse algoritmi tööpõhimõte.
5. Realiseerida õppija ja õpetaja rollidele vastav veebirakendus.
6. Rakendada ülesannete valimine, vastuste kontrollimine ja õppija taseme uuendamine.
7. Testida prototüübi põhifunktsionaalsust ning analüüsida loodud lahenduse tulemusi ja piiranguid.

Prototüübi taustsüsteem on arendatud Pythonis FastAPI raamistikuga. Kasutajaliides on loodud Reacti ja TypeScriptiga ning andmete talletamiseks kasutatakse PostgreSQL-i. Rakenduse käivitamiseks kasutatakse Dockerit, et lihtsustada arenduskeskkonna seadistamist ja süsteemi osade koos töötamist. Adaptiivne mehhanism põhineb õppija soorituse jälgimisel ja Elo-laadsel reitingu uuendamisel, mille abil hinnatakse õppija hetketaset ning valitakse sobiv ülesannete raskusaste.

Töö koosneb kolmest suuremast osast. Esimene osa annab ülevaate probleemi teoreetilisest poolest: e-õppe platvormidest, personaliseeritud õppes, adaptiivsetest õppesüsteemidest, algoritmide ja olemasolevatest lahendustest. Teine osa kirjeldab õppeplatvormi prototüübi arendamist – see hõlmab nii nõuete analüüsi, meetodika valikut kui ka süsteemi kavandamist, sealhulgas arhitektuuri, kasutusvooge, andmebaasi, API-t ja adaptiivset algoritmi. Samuti kirjeldab see prototüübi realiseerimist. Viimane osa keskendub tulemuslikkuse testimise

metoodikale, saadud tulemuste analüüsile ning toob välja lahenduse piirangud. Enne töö kokkuvõtet kirjeldatakse ka prototüübi edasiarenduse võimalusi.

## **2 Valdkonna ülevaade**

Adaptiivse õppeplatvormi arendamisel ei piisa ainult tehnilise lahenduse kavandamisest. Enne süsteemi nõuete ja arhitektuuri määratlemist tuleb mõista, millist õppimist platvorm toetab, kuidas digitaalsed õpikeskkonnad õppija tegevust suunavad ning millistel alustel saab ülesannete raskusastet kohandada. Käesoleva töö kontekstis on see oluline lähtekoht, sest prototüüp keskendub matemaatika harjutusülesannetele, kus sobiv raskusaste, järjepidev harjutamine ja õigeaegne tagasiside mõjutavad otseselt õppija edenemist.

### **2.1 E-õppe platvormid ja personaliseeritud õpe**

E-õppe platvormide all mõistetakse digitaalseid keskkondi, mis toetavad õppematerjalide jagamist, ülesannete lahendamist, suhtlust, hindamist ja õppetegevuse jälgimist. Sellised keskkonnad võivad olla erineva ulatusega: osa neist toimib peamiselt materjalide ja ülesannete haldamise vahendina, teised sisaldavad interaktiivseid harjutusi, automaatset tagasisidet või õppija tegevuse analüütikat. Euroopa Liidu digihariduse tegevuskava rõhutab, et digitaalsed lahendused peaksid aitama kaasa kvaliteetsele ja kättesaadavale haridusele, mitte olema üksnes tavapärase õppe elektrooniline koopia [1].

OECD digihariduse ülevaates käsitletakse erinevaid digitaalseid õppevahendeid, alates õppematerjalide ja õppehaldussüsteemide kasutamisest kuni nutikamate digivahenditeni, mis võimaldavad interaktiivsust, andmete kasutamist ja õppetegevuse kohandamist [2]. Seega ei seisne e-õppe platvormi väärtus ainult selles, et õppematerjal on veebis kättesaadav, vaid ka selles, kuidas keskkond toetab õppija tegevust, tagasiside saamist ja õpetaja otsuste tegemist.

Personaliseeritud õppe eesmärk on arvestada õppija teadmiste, tempo, vajaduste ja varasemate tulemustega. Matemaatika õppimisel avaldub see selgelt, sest sama teema võib erinevatele õppijatele valmistada erineval määral raskusi. Samal ajal kui üks õpilane on valmis liikuma keerukamate ülesannete juurde, vajab teine teema kordamist ja rohkem

harjutamist. Kui e-õppe platvorm ei arvesta õppija sooritust, jääb õpetajal suur osa kohandamisest käsitsi teha ning süsteem ise ei kasuta ära õppetegevuse käigus tekkivaid andmeid.

Veebipõhine diagnostiline hindamine võib personaliseeritud õpet toetada, sest see võimaldab koguda andmeid õppija teadmiste ja oskuste kohta ning kasutada neid edasise õppe suunamiseks. Csapó ja Molnár kirjeldavad eDia süsteemi näitel, kuidas tehnoloogiapõhine hindamine saab anda õpetajale täpsemat infot õppija teadmiste taseme kohta ning toetada õppimise individualiseerimist [3]. See toetab ka käesoleva töö lähtekohta, kuna loodava prototüübi üks eesmärke on kasutada õppija varasemaid vastuseid järgmiste ülesannete raskusastme valimisel.

Samas ei taga tehnoloogia kasutamine iseenesest paremaid õpitulemusi. UNESCO 2023. aasta haridustehnoloogia raport rõhutab, et tehnoloogia mõju sõltub sellest, kas see lahendab tegelikku hariduslikku probleemi ja kas seda kasutatakse sobivas õppeolukorras [5]. Seetõttu peab e-õppe platvormi kavandamisel lähtuma mitte ainult tehnilisest teostatavusest, vaid ka sellest, millist õppetegevust süsteem toetab. Sellest tulenevalt ei piisa käesoleva töö kontekstis ainult tavapärase e-õppe platvormi käsitlemisest. Kuna prototüübi eesmärk on kasutada õppija vastuseid järgmiste ülesannete raskusastme valimiseks, tuleb eraldi käsitleda adaptiivseid õppesüsteeme.

## **2.2 Adaptiivsed õppesüsteemid**

Adaptiivne õppesüsteem erineb tavapärasest e-õppe platvormist selle poolest, et süsteem ei paku kõigile õppijatele sama järjestusega õppesisu, vaid muudab õppematerjali või ülesannete valikut õppija tegevuse põhjal. Kohandamine võib toimuda näiteks õppija vastuste õigsuse, lahendamise ajaloo, hinnangulise teadmiste taseme või konkreetsete raskuskohtade alusel.

Adaptiivse õppesüsteemi keskne osa on õppija mudel. Selle abil hoiab süsteem infot õppija seniste tulemuste kohta ning otsustab järgmise õppetegevuse üle. Matemaatika harjutuskeskkonnas võib selline otsus tähendada näiteks seda, kas õppijale kuvatakse lihtsam, keskmine või keerulisem ülesanne. Kui õppija vastab järjest õigesti, võib süsteem tõsta raskusastet; kui õppija teeb mitu viga, võib süsteem pakkuda lihtsamaid ülesandeid.

Adaptiivsust võib rakendada erineva keerukusega. Lihtsamal juhul kasutatakse reeglipõhist lähenemist, kus raskusaste muutub kindlate tingimuste alusel. Keerukamad süsteemid võivad kasutada statistilisi või masinõppel põhinevaid mudeleid, mis hindavad õppija teadmiste taset ja ülesannete raskust tõenäosuslikult. Wauters, Desmet ja Van den Noortgate käsitlevad adaptiivseid ülesandepõhiseid õpikeskkondi ning toovad välja, et õppija teadmiste taseme ja ülesande raskuse sobitamine on üks oluline võimalus muuta õppimist tõhusamaks [6].

Käesolevas töös loodav prototüüp kasutab adaptiivsust piiratud, kuid praktiliselt rakendataval kujul. Süsteem ei püüa katta kogu matemaatika õppekava ega luua keerukat õppija teadmiste mudelit kõigi alamteemade lõikes. Selle asemel keskendub prototüüp teemade kaupa harjutamisele, automaatsele vastuste kontrollimisele ja õppija hinnangulise taseme uuendamisele seniste tulemuste põhjal. Selline lähenemine sobib prototüübi eesmärgiga, sest võimaldab näidata adaptiivse õppe põhimõtet ilma liigse tehnilise keerukuseta.

Adaptiivse õppesüsteemi praktiline toimimine sõltub sellest, kuidas süsteem õppija taset hindab ja kuidas selle põhjal järgmisi ülesandeid valib. Seetõttu tuleb eraldi käsitleda algoritmilisi lähenemisi, mille abil saab õppija taset hinnata ja ülesandeid valida.

### **2.3 Adaptiivse õppimise algoritmid**

Adaptiivse õppimise algoritmide eesmärk on hinnata õppija taset ning valida selle põhjal sobiv järgmine õppetegevus. Algoritmi valik sõltub sellest, milliseid andmeid süsteem kogub, kui palju ülesandeid on olemas, kui täpset õppija mudelit soovitakse luua ning kui hästi peab süsteem olema seletatav õpetajale ja õppijale.

Üks tuntud lähenemine on teadmused jälgimine, mille puhul hinnatakse, kas õppija on konkreetse oskuse omandanud või mitte. Pelánek annab ülevaate *Bayesian Knowledge Tracing* ja logistiliste mudelite kasutamisest õppija teadmiste jälgimisel ning rõhutab, et mudeli valik sõltub konkreetsest kasutuskontekstist [7]. Sellised mudelid võivad olla kasulikud siis, kui süsteemis on palju alamteemasid ja iga ülesanne on seotud kindlate oskustega.

Teine levinud lähenemine on *Item Response Theory* ehk IRT, mille puhul seostatakse õppija võimekus ja ülesande raskus tõenäosusliku mudeli abil. IRT-d kasutatakse sageli

adaptiivses testimises, kuid seda on uuritud ka õppimisele suunatud ülesandepõhistes keskkondades [6]. IRT eeldab üldjuhul piisavat hulka ülesandeid ja andmeid, et hinnata usaldusväärselt nii õppija taset kui ka ülesannete raskust.

Klinkenberg, Straatemeier ja van der Maas on uurinud arvutipõhist adaptiivset matemaatika harjutamist ning käsitletud õppija võimekuse ja ülesande raskuse jooksva hindamise võimalust õppeprotsessi käigus [4]. See on käesoleva töö jaoks oluline lähtekoht, sest ka loodav prototüüp peab siduma õppija senise soorituse järgmiste ülesannete raskusastmega.

IRT ja Elo-laadne lähenemine sobivad erineva mahuga lahendustele. IRT võimaldab modelleerida õppija võimekuse ja ülesande raskuse seost täpsemalt, kuid eeldab tavaliselt suuremat ülesannete hulka ning piisavalt vastuste andmeid, et mudeli parameetreid usaldusväärselt hinnata. Käesoleva töö prototüübis on teemade ja ülesandetüüpide arv piiratud ning eesmärk on näidata adaptiivse raskusastme valimise põhivoogu. Seetõttu on prototüübi jaoks sobivam Elo-laadne lähenemine, mis võimaldab õppija hinnangulist taset uuendada iga vastuse järel ning on lihtsamini seletatav.

Elo reitingusüsteem on algselt tuntud malemängijate taseme hindamisest, kuid seda saab kasutada ka adaptiivsetes õppesüsteemides, tõlgendades õppija vastust kui kohtumist õppija ja ülesande vahel. Kui õppija lahendab ülesande õigesti, tõuseb tema hinnanguline tase; vale vastuse korral see langeb. Pelánek kirjeldab Elo reitingusüsteemi kasutamist adaptiivsetes õppesüsteemides ning toob välja, et selline lähenemine võimaldab dünaamiliselt hinnata õppija oskust ja ülesannete raskust [8].

Prototüübis kasutatakse Elo-laadset mehhanismi koos lihtsate reeglitega, mis määravad õppijale sobiva raskusastme. Selline lahendus on prototüübi jaoks põhjendatud kolmel põhjusel. Esiteks on see arusaadav ja tehniliselt teostatav väiksema andmemahuga. Teiseks sobib see süsteemile, kus ülesanded on jaotatud raskusastmeteks. Kolmandaks on algoritmi otsuseid lihtsam selgitada, mis on oluline õpetaja vaate ja süsteemi usaldusväärsuse seisukohalt. Kuna adaptiivset ja personaliseeritud õpet rakendatakse erinevates õppeplatvormides erineval kujul, on vajalik võrrelda ka olemasolevaid lahendusi ning määratleda, millist rolli täidab käesolevas töös loodav prototüüp.

## 2.4 Olemasolevate lahenduste võrdlus

Olemasolevate lahenduste võrdlemisel tuleb eristada üldotstarbelisi õppehaldussüsteeme ja matemaatikale keskendunud harjutuskeskkondi. Võrdlus ei hinda nende lahenduste pedagoogilist tulemuslikkust, vaid keskendub avalikult kirjeldatud funktsionaalsustele: õppesisu haldamine, automaatne tagasiside, adaptiivsus, õpetajale kuvatav analüütika ja sobivus käesoleva töö eesmärgiga.

Võrdluseks valiti Moodle, Khan Academy, IXL ja Matific, sest need esindavad erinevaid lähenemisi e-õppe ja matemaatika harjutamise toetamisele. Tabelis 1 võrreldakse valitud lahendusi tunnuste alusel, mis on käesoleva töö eesmärgi seisukohalt olulised.

Tabel 1. Olemasolevate lahenduste võrdlus loodava prototüübiga.

Lahendus	Sisu haldus	Tagasiside	Adaptiivsus	Õpetaja analüütika
Moodle	Jah	Osaliselt	Osaliselt	Jah
Khan Academy	Piiratud	Jah	Osaliselt	Osaliselt
IXL	Piiratud	Jah	Jah	Jah
Matific	Piiratud	Jah	Jah	Jah
Loodav prototüüp	Piiratud	Jah	Jah	Jah

Loodava prototüübi puhul tähendab tabelis märgitud piiratud sisu haldus seda, et õpetaja saab hallata grupe ja aktiveerida teemasid, kuid ei loo süsteemis vabalt uusi ülesandetüüpe.

Moodle on avatud lähtekoodiga õppehaldussüsteem, mida saab kasutada kursuste, õppe-materjalide, ülesannete ja hindamise haldamiseks. Moodle'i tugevus on paindlikkus ja laiendatavus, sest platvormi saab kohandada erinevate õppeasutuste vajadustele [9]. Käesoleva töö seisukohalt on Moodle aga pigem üldine õppehalduskeskkond kui spetsiaalselt matemaatika harjutusülesannete adaptiivseks valimiseks loodud prototüüp.

Khan Academy pakub õppijatele õppematerjale, harjutusi ja soovitusi. Khan Academy dokumentatsioonis kirjeldatakse MAP Recommended Practice lahendust, mis võimaldab siduda õppijale soovitatud matemaatikaharjutusi MAP Growth testi tulemustega [10]. See näitab, et personaliseeritud harjutamine võib põhineda õppija taseme kohta kogutud andmetel. Samas on tegemist valmisplatvormiga, mille sisemisi algoritme ja andmemudeleid

ei saa lõputöö raames samal viisil kavandada, muuta ja analüüsida nagu enda loodud prototüübis.

IXL on personaliseeritud õppeplatvorm, mis pakub matemaatika ja teiste ainete harjutusi, kohest tagasisidet, soovitusi ja õpetajale mõeldud analüütikat [11]. IXL-i puhul on oluline, et õppija harjutuste raskusastet ja soovitusi saaks siduda tema edenemisega. Käesoleva töö prototüüp on sellest kitsam, kuid keskendub sarnase põhimõtte tehnilisele realiseerimisele: õppija soorituse põhjal valitakse sobiv raskusaste ja õpetajale kuvatakse tulemuste ülevaade.

Matific on matemaatikale keskendunud õppeplatvorm, mis kirjeldab oma lahenduses adaptiivset algoritmi, individuaalseid õpiradu, kohest tagasisidet ja õpetajale mõeldud reaalses analüütikat [12]. Matific on käesoleva tööga sisuliselt lähedasem kui üldised õppehaldussüsteemid, sest mõlemad keskenduvad matemaatika harjutamisele ja õppija edenemise jälgimisele. Erinevus seisneb selles, et käesoleva töö eesmärk ei ole luua valmis kommertslahendusega võrreldavat süsteemi, vaid arendada ja analüüsida prototüüpi, mille arhitektuur, andmemudel ja adaptiivne mehhanism on töö autorite poolt kavandatud ja realiseeritud.

Olemasolevate lahenduste võrdlus näitab, et personaliseeritud harjutamine, automaatne tagasiside ja õpetajale kuvatav analüütika on tänapäevastes õppeplatvormides olulised funktsioonid. Valdkonna ülevaatest järeldeb, et e-õppe platvormi väärtus ei seisne ainult õppematerjalide veebipõhises kättesaadavuses, vaid ka õppija tegevuse toetamises, tagasiside andmises ja õpetajale otsustamiseks vajaliku info kuvamises. Käesoleva töö prototüübi eripära on kitsam fookus: süsteem keskendub gümnaasiumiastme matemaatika harjutusülesannete valimisele, vastuste automaatsele kontrollimisele, õppija hinnangulise taseme uuendamisele ja õpetajale tulemuste ülevaate kuvamisele.

### **3 Prototüübi lähtetingimused, nõuded ja metoodika**

Adaptiivse õppeplatvormi prototüübi arendamine eeldab enne tehnilise lahenduse kavandamist selgete lähtetingimuste ja nõuete määratlemist. Kuna töö eesmärk ei ole valmis kommertstoote loomine, vaid toimiva prototüübi arendamine ja analüüsimine, tuleb nõuded siduda töö mahu, sihtrühma, kasutajarollide ja valitud õppeülesandega. Nende aspektide määratlemine loob aluse järgmises peatükis 4 kirjeldatavale süsteemi kavandamisele.

#### **3.1 Lähtetingimused ja sihtrühm**

Käesoleva töö lähtetingimuseks on luua veebipõhine prototüüp, mis toetab matemaatika harjutusülesannete lahendamist ja võimaldab ülesannete raskusastet õppija senise soorituse põhjal kohandada. Prototüüp ei käsitle kogu matemaatika õppekava, vaid keskendub piiratud arvule teemadele, et adaptiivse õppe põhimõtet oleks võimalik töö mahu piires realiseerida, katsetada ja analüüsida. Töö praeguses mahus toetab süsteem trigonomeetria, kombinatoorika ja tõenäosuse teemasid.

Trigonomeetria ning kombinatoorika ja tõenäosus valiti prototüübi esimesteks teemadeks, sest need võimaldavad koostada eri raskusastmega harjutusülesandeid ning kontrollida vastuseid automaatselt. Samuti erinevad need teemad ülesandetüüpide ja vastusevormingute poolest, mis aitab prototüübi raames kontrollida, kas süsteemi ülesannete genereerimise ja vastuste valideerimise loogika on piisavalt paindlik.

Prototüübi sihtrühmaks on gümnaasiumiastme matemaatika õppijad ja õpetajad. Õppija jaoks peab süsteem pakkuma võimalust lahendada teemade kaupa harjutusülesandeid, saada vastuse õigsuse kohta tagasisidet ning jälgida oma edenemist. Õpetaja jaoks peab süsteem pakkuma võimalust luua õppegrupe, hallata gruppide liikmeid, aktiveerida õppeteemasid ning vaadata õppijate tulemuste ülevaadet.

Oluline lähtetingimus on ka see, et loodav lahendus on prototüüp. See tähendab, et süsteem peab demonstreerima adaptiivse matemaatika harjutuskeskkonna põhifunktsionaalsust, kuid

ei pea sisaldama kõiki võimalusi, mida võiks oodata täismahus õppeplatvormilt. Näiteks ei ole töö keskmis vabalt uute ülesandetüüpide loomine õpetaja poolt ega kogu matemaatika õppekava katmine. Ülesannete genereerimine toimub teemaspetsiifiliste generaatorite abil ning adaptiivne mehhanism kasutab õppija seniseid vastuseid raskusastme valimiseks.

### 3.2 Kasutajarollid

Süsteemis eristatakse kahte peamist kasutajarolli: õppija ja õpetaja. Rollide eristamine on vajalik, sest nende tegevused ja ligipääsuõigused on erinevad. Õppija keskendub ülesannete lahendamisele ja enda edenemise jälgimisele, õpetaja aga õppegruppide ja õppeteemade haldamisele ning õppijate tulemuste analüüsimisele.

Õppija rolli kuuluvad järgmised pädevused:

- kasutajakonto loomine ja süsteemi sisselogimine;
- õpetaja loodud grupiga liitumine kutsekoodi abil;
- aktiivsete õppeteemade vaatamine oma gruppides;
- süsteemi pakutud harjutusülesannete lahendamine;
- vastuse esitamine ja automaatse tagasiside saamine;
- oma edenemise, täpsuse ja lahendamiskatsete ülevaate nägemine.

Õpetaja rolli kuuluvad järgmised pädevused:

- õpetaja konto loomine registreerimiskoodi abil;
- õppegruppide loomine ja muutmine;
- grupi kutsekoodi haldamine;
- õppijate liitumise võimaldamine kutsekoodi abil ning vajadusel õppijate grupist eemaldamine;
- õppeteemade aktiveerimine või deaktiveerimine grupi jaoks;
- grupi, teema ja üksikute õppijate tulemuste analüüsimine.

Rollipõhine jaotus vähendab süsteemi keerukust kasutaja jaoks ning toetab andmete turvalisemat käsitlemist. Õppija ei pea nägema õpetaja haldusfunktsioone ning õpetaja saab vaadata ainult nende gruppide andmeid, mille haldaja ta on.

### 3.3 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad tegevusi, mida süsteem peab kasutajatele võimaldama. Nõuete määratlemisel lähtuti töö eesmärgist, valdkonna taustast ning prototüübi praktilisest ulatusest. Tabelis 2 on esitatud süsteemi peamised funktsionaalsed nõuded.

Tabel 2. Prototüübi funktsionaalsed nõuded.

Tähis	Nõue
FN-01	Süsteem peab võimaldama kasutajal registreeruda õppija või õpetajana.
FN-02	Süsteem peab võimaldama kasutajal sisse ja välja logida.
FN-03	Süsteem peab rakendama rollipõhist ligipääsu õppija ja õpetaja funktsioonidele.
FN-04	Õpetaja peab saama luua, muuta ja hallata õppegruppe.
FN-05	Õpetaja peab saama hallata grupi kutsekoodi ja selle aktiivsust.
FN-06	Õppija peab saama liituda grupiga kutsekoodi abil.
FN-07	Õpetaja peab saama aktiveerida ja deaktiveerida õppeteemasid oma grupis.
FN-08	Õppija peab nägema ainult neid teemasid, mis on tema grupis aktiivsed.
FN-09	Süsteem peab genereerima õppijale valitud teema ja tema hinnangulise taseme põhjal harjutusülesande.
FN-10	Süsteem peab võimaldama õppijal esitada vastuse genereeritud ülesandele.
FN-11	Süsteem peab kontrollima õppija vastust automaatselt.
FN-12	Süsteem peab pärast vastuse esitamist uuendama õppija teemakohast edenemist.
FN-13	Süsteem peab kohandama õppijale pakutavate ülesannete raskusastet tema seniste tulemuste põhjal.
FN-14	Õppija peab saama vaadata oma teemakohast edenemist ja tulemusi.
FN-15	Õpetaja peab saama vaadata grupi, teema ja õppija tulemuste analüütikat.

Funktsionaalsete nõuete koostamisel arvestati, et prototüübi põhieesmärk on näidata terviklikku adaptiivse harjutamise voogu. Seetõttu on nõuete keskmes õppija ülesande lahendamise tsükkel: ülesande küsimine, vastuse esitamine, vastuse kontrollimine, tulemuse

salvestamine ja õppija taseme uuendamine.

### 3.4 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded kirjeldavad süsteemi omadusi, mis ei ole seotud üksiku kasutajategevusega, kuid mõjutavad lahenduse kvaliteeti, kasutatavust ja edasiarendatavust. Prototüübi puhul on olulised eelkõige kasutatavus, töökindlus, turvalisus, laiendatavus ja käivitatavus.

Tabel 3. Prototüübi mittefunktsionaalsed nõuded.

Tähis	Nõue
MFN-01	Kasutajaliides peab olema arusaadav nii õppija kui ka õpetaja rollis kasutajale.
MFN-02	Kasutaja peab pääsema ligi ainult nendele andmetele ja vaadetele, mis vastavad tema rollile ja seotud õppegruppidele.
MFN-03	Kasutaja paroolid peavad olema salvestatud räsitud kujul.
MFN-04	Süsteem peab säilitama õppija vastused ja tulemused viisil, mis võimaldab hiljem edenemist ja analüütikat kuvada.
MFN-05	Süsteemi ülesehitus peab võimaldama laiendada ülesannete genereerimise loogikat, sealhulgas lisada uusi teemasid või kasutada tulevikus teistsugust genereerimismehhanismi.
MFN-06	Rakendus peab olema käivitatav konteinerite abil.
MFN-07	Taustsüsteemi API peab olema dokumenteeritav ja testitav.
MFN-08	Süsteemi põhifunktsionaalsus peab olema kontrollitav automaatsete testidega.
MFN-09	Prototüüpi peab olema võimalik hinnata sihtrühma esindajatega kasutatavuse testimise kaudu.

Turvalisuse seisukohalt on oluline, et õppija ja õpetaja ligipääsuõigused oleksid eraldatud. Õppija ei tohi pääseda ligi õpetaja haldusvaadetele ega teiste õppijate analüütikale. Õpetaja peab nägema ainult nende gruppide andmeid, mida ta ise haldab. Kuna süsteem käsitleb kasutajakontosid ja õppijate tulemusi, tuleb autentimise ja andmete talletamise juures vältida lahendusi, mis võiksid võimaldada volitamata ligipääsu.

Laiendatavuse seisukohalt peab prototüübi ülesehitus võimaldama tulevikus täiendada ülesannete genereerimise loogikat. See ei tähenda ainult uute matemaatikateemade lisamist, vaid ka võimalust asendada või täiendada teemaspetsiifilisi generaatoreid teistsuguse genereerimismehhanismiga. Käesolevas töös kasutatakse deterministlikke generaatoreid, kuid süsteemi kavandamisel tuleb arvestada, et edasises arenduses võiks ülesannete loomist toetada ka tehisintellektil põhinev lahendus. Sellisel juhul peab siiski säilima võimalus kontrollida ülesannete korrektsust ja vastuste valideeritavust.

### **3.5 Arendus- ja valideerimismetoodika**

Töö praktiline osa viidi läbi iteratiivse prototüüpimise põhimõttel. See sobib olukorda, kus eesmärk on luua töötav tarkvaraline prototüüp, mille funktsionaalsust saab arenduse käigus järk-järgult täpsustada. Iteratiivne lähenemine võimaldas esmalt realiseerida süsteemi põhivoo ning seejärel täiendada seda gruppide, teemade, adaptiivse mehhanismi ja analüütikaga. Metoodika sobis töö eesmärgiga, sest adaptiivse mehhanismi ja kasutajaliidese lahendusi oli võimalik täpsustada pärast põhifunktsionaalsuse esmast realiseerimist.

Arendus algas süsteemi põhiobjektide ja kasutajarollide määratlemisest. Selle põhjal kavandati andmemudel, mis toetab kasutajaid, grupe, õppeteemasid, ülesandeid, vastuseid ja õppija edenemist. Seejärel realiseeriti autentimine ja rollipõhine ligipääs, sest need on vajalikud peaaegu kõigi teiste funktsioonide turvaliseks kasutamiseks.

Pärast kasutajate ja gruppide põhifunktsionaalsuse loomist keskenduti õppeteemadele ja harjutusülesannetele. Süsteemi lisati võimalus aktiveerida teemasid gruppide kaupa, genereerida õppijale järgmine ülesanne ning kontrollida esitatud vastust. Seejärel lisati adaptiivne mehhanism, mis uuendab õppija hinnangulist taset pärast iga vastust. Viimases etapis täiendati õpetaja ja õppija vaateid analüütika ning edenemise kuvamisega.

Metoodika valikul arvestati, et tegemist on kahe autoriga praktilise tarkvaraarendusprojektiga. Töö jagunes loogiliselt taustsüsteemi, andmemudeli, adaptiivse mehhanismi, kasutajaliidese ja testimise vahel. Arendusprotsessi käigus kontrolliti loodud funktsionaalsust automaatsete testide ja käsitsi katsetamise abil.

Lisaks tehnilisele testimisele kavandati prototüübi tulevane katsetamine reaalses õppekeskkonnas. Katsetamise eesmärk on kontrollida, kas õppija ja õpetaja vaated on arusaadavad

ning kas ülesannete lahendamise põhivoog toimib tegelikus kasutusolukorras. Kasutatavuse hindamise tulemusi käsitletakse töö testimise ja tulemuste analüüsi peatükis 6.

### **3.6 Tehnoloogiliste valikute põhjendus**

Prototüübi tehnoloogilised valikud peavad toetama kiiret arendust, selget arhitektuuri, laiendatavust ja sobivust adaptiivse õppeplatvormi loomiseks. Lahendus koosneb kolmest peamisest osast: taustsüsteemist, kasutajaliidesest ja andmebaasist. Lisaks kasutatakse konteineriseerimist, et lihtsustada rakenduse käivitamist ja arenduskeskkonna ühtlustamist.

Taustsüsteemi arendamiseks valiti Python ja FastAPI. Python sobib hästi prototüübi arendamiseks, sest selle süntaks on loetav ning ökosüsteem toetab hästi andmetöötlust ja algoritmilist loogikat. FastAPI valiti seetõttu, et see toetab tüübivihjeid, andmete valideerimist ja automaatselt genereeritavat OpenAPI dokumentatsiooni [13]. See on oluline REST API põhise rakenduse puhul, kus kasutajaliides suhtleb taustsüsteemiga selgelt määratletud päringute kaudu.

Kasutajaliidese arendamiseks valiti React ja TypeScript. React sobib komponentpõhise kasutajaliidese loomiseks ning võimaldab jagada vaated väiksemateks taaskasutatavateks osadeks [14]. TypeScript lisab JavaScriptile staatilise tüübikontrolli, mis aitab vähendada arendusvigu ja muudab suurema kasutajaliidese koodi paremini hallatavaks [15]. Käesoleva prototüübi puhul on see oluline, sest kasutajaliides sisaldab mitut rollipõhist vaadet, vorme, teenusepäringuid ja analüütika kuvamist.

Andmebaasina valiti PostgreSQL, sest tegemist on töökindla relatsioonilise andmebaasisüsteemiga, mis sobib struktureeritud andmete hoidmiseks. Prototüüp vajab seoseid kasutajate, gruppide, teemade, ülesannete ja vastuste vahel, mistõttu relatsiooniline andmemudel on sobiv valik. Lisaks toetab PostgreSQL JSONB andmetüüpi, mis võimaldab salvestada genereeritud ülesannete sisu paindlikul kujul [16]. See on kasulik, sest erinevate teemade ülesanded võivad sisaldada erineva struktuuriga andmeid.

Rakenduse käivitamiseks ja arenduskeskkonna ühtlustamiseks kasutatakse Dockerit ja Docker Compose'i. Docker Compose võimaldab kirjeldada mitmest teenusest koosneva rakenduse, näiteks taustsüsteemi, kasutajaliidese ja andmebaasi, ning käivitada need ühtse konfiguratsiooni alusel [17]. See vähendab käsitsi seadistamise vajadust ja lihtsustab

projekti käivitamist erinevates arenduskeskkondades.

Valitud tehnoloogiad määravad süsteemi kavandamise lähtekohad. Kuna lahendus põhineb eraldatud kasutajaliidesel, taustsüsteemil ja andmebaasil, tuleb edasises kavandamises eraldi käsitleda üldarhitektuuri, kasutusvooge, andmemudelit, API-t ja adaptiivse algoritmi tööpõhimõtet.

## 4 Süsteemi kavandamine

Süsteemi kavandamisel lähtuti eelmises peatükis 3 määratletud nõuetest ja prototüübi praktilisest ulatusest. Kavandatav lahendus peab toetama õppija ja õpetaja rolle, grupipõhist õppimist, teemade aktiveerimist, ülesannete moodustamist teemaspetsiifilise loogika põhjal, automaatset vastuste kontrollimist ning õppija taseme uuendamist. Kuna tegemist on veebipõhise prototüübiga, jaotati süsteem kasutajaliideseks, taustsüsteemiks ja andmekihiks.

### 4.1 Süsteemi üldarhitektuur

Süsteem kavandati kolmekihilise veebirakendusena. Kasutajaliides töötab brauseris ning suhtleb taustsüsteemiga HTTP-päringute kaudu. Taustsüsteem pakub REST API-t, kontrollib kasutaja õigusi, haldab ärioloogikat, moodustab teemaspetsiifiliste generaatorite abil ülesandeid, valideerib vastuseid ning uuendab õppija edenemist. Andmekiht põhineb PostgreSQL andmebaasil, kus talletatakse kasutajad, grupid, teemad, moodustatud ülesanded, vastused ja õppija adaptiivne seisund.

Selline jaotus võimaldab süsteemi osasid arendada ja testida eraldi. Kasutajaliides ei pea teadma ülesannete moodustamise ega adaptiivse algoritmi sisemist loogikat, vaid kasutab selleks taustsüsteemi API-t. Taustsüsteem omakorda eraldab kasutajaliidese andmebaasi struktuurist ning tagab, et andmetele pääseb ligi ainult kontrollitud päringute kaudu.

Kasutajaliidese vaates on süsteem jagatud rollipõhisteks vaadeteks. Õppija näeb oma gruppe, aktiivseid teemasid, harjutusülesandeid ja edenemist. Õpetaja näeb enda loodud gruppe, grupi liikmeid, õppeteemasid ja õppijate tulemuste analüütikat. Rollipõhine jaotus kajastub ka taustsüsteemis, kus API päringute täitmisel kontrollitakse kasutaja rolli ja seost vastava grupiga.

Taustsüsteemi kavandamisel eraldati mitu vastutusala. Autentimise osa tegeleb registreerimise, sisselogimise, ligipääsutokenite ja värskendustokenitega. Gruppide osa haldab õppegrupe, kutsekoode ja liikmesust. Teemade osa määrab, millised õppeteemad on plat-

vormil olemas ja millised neist on konkreetsetes gruppis aktiivsed. Ülesannete ja adaptiivse õppimise osa vastutab ülesannete moodustamise, vastuste kontrollimise ja õppija taseme uuendamise eest.

Pärast süsteemi põhikomponentide ja vastutusvaldade määratlemist saab kirjeldada kasutusvooge, mille kaudu õppija ja õpetaja süsteemi põhifunktsionaalsust kasutavad.

## **4.2 Kasutusvood**

Süsteemi kavandamisel määratleti peamised kasutusvood, mis katavad prototüübi keskse funktsionaalsuse. Õpetaja töövoog algab konto loomise ja sisselogimisega. Õpetaja loob õppegrupi, millele süsteem loob kutsekoodi. Selle kutsekoodi abil saavad õppijad grupiga liituda. Õpetaja saab hiljem kutsekoodi aktiivsust muuta või koodi uuendada.

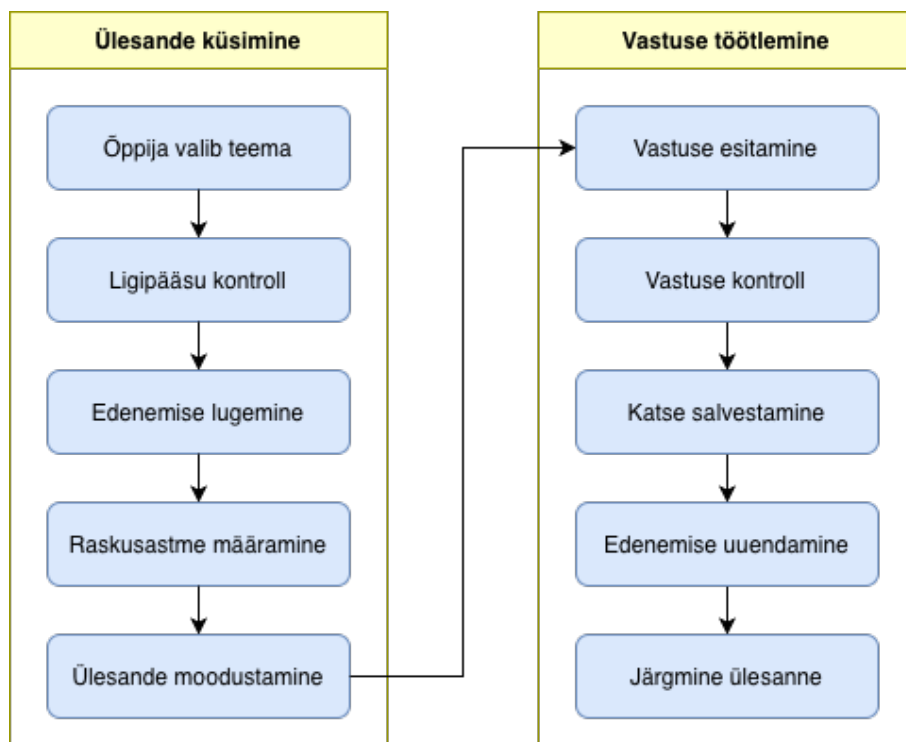
Õppija töövoog algab konto loomise ja sisselogimisega. Pärast sisselogimist saab õppija liituda grupiga õpetajalt saadud kutsekoodi abil. Kui õppija kuulub gruppi, näeb ta selles grupis aktiivseid õppeteemasid. Õppija ei saa ligi teemadele, mida õpetaja ei ole vastavas grupis aktiveerinud.

Õppeteema kasutusvoog algab õpetaja tegevusest. Õpetaja valib grupi ning aktiveerib seal üht või mitu platvormi teemakataloogis olevat teemat. Pärast teema aktiveerimist muutub see õppijatele nähtavaks. Kui teema deaktiveeritakse, ei saa õppijad seda enam grupis harjutamiseks kasutada.

Ülesande lahendamise töövoog on prototüübi keskne kasutusvoog. Õppija valib aktiivse teema ja küsib süsteemilt järgmise ülesande. Taustsüsteem kontrollib, kas õppija kuulub vastavasse gruppi ja kas teema on aktiivne. Seejärel leitakse õppija teemakohane edenemine, määratakse sobiv raskusaste ja moodustatakse ülesanne teemaspetsiifilise loogika põhjal. Moodustatud ülesanne salvestatakse andmebaasi eraldi ülesande eksemplarina.

Pärast ülesande kuvamist esitab õppija vastuse. Taustsüsteem kontrollib, kas ülesanne kuulub samale õppijale, grupile ja teemale ning kas sellele ülesandele ei ole juba vastatud. Seejärel valideeritakse õppija vastus, salvestatakse katse ja uuendatakse õppija teemakohast edenemist. Uuendamise käigus muutuvad õppija katsete arv, õigete vastuste arv, täpsus, järjekuste õigete või valede vastuste arv, adaptiivne reiting ja vajadusel ka raskusaste.

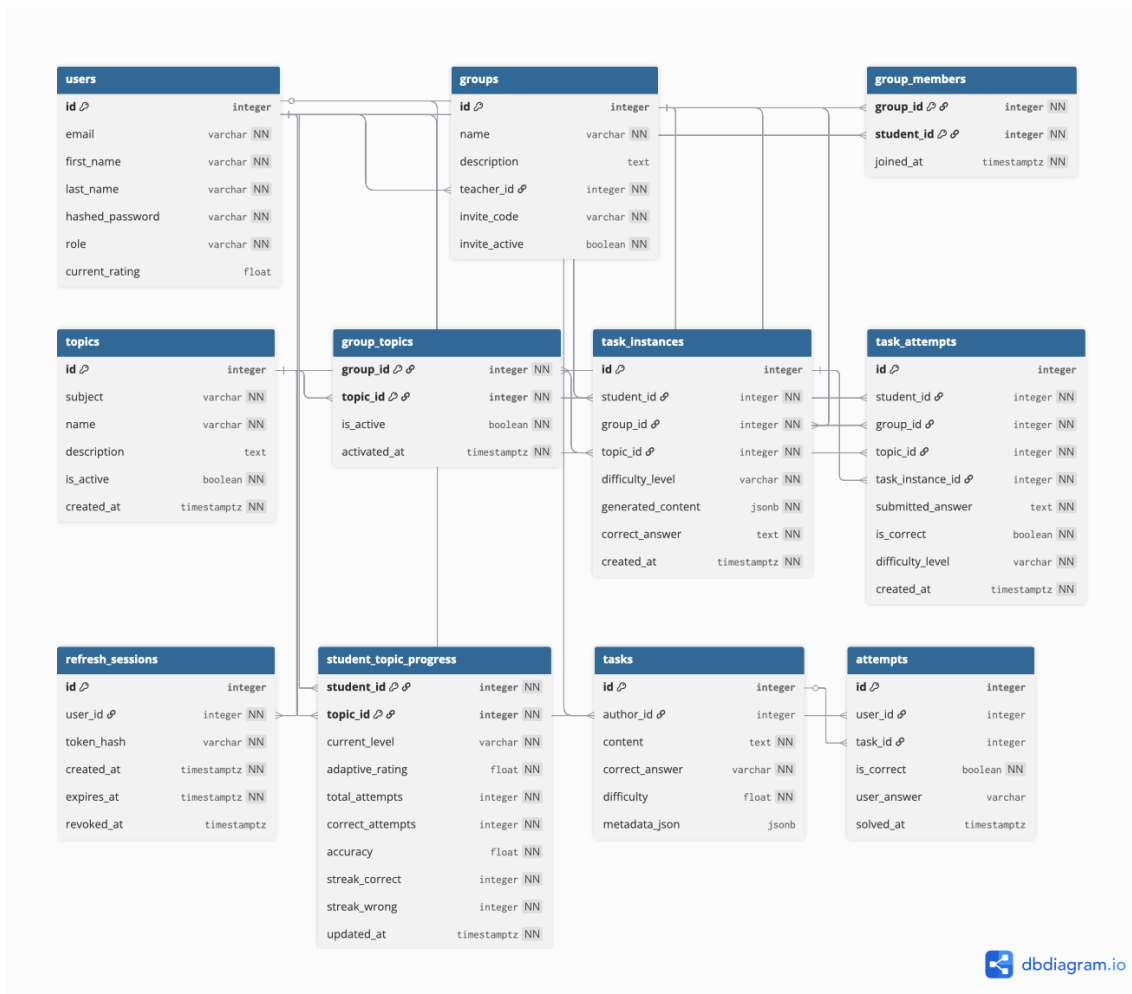
Joonisel 1 on esitatud õppija ülesande lahendamise voog alates teema valimisest kuni järgmise ülesande valimiseni.



Joonis 1. Õppija ülesande lahendamise voog.

### 4.3 Andmebaasi mudel

Andmebaasi mudel kavandati relatsioonilisena, sest süsteemis on selged seosed kasutajate, gruppide, teemade, ülesannete ja vastuste vahel. Samal ajal kasutatakse ülesannete sisu hoidmiseks JSONB andmetüüpi, kuna eri teemade ülesanded võivad sisaldada erineva struktuuriga andmeid. Joonisel 2 on esitatud prototüübi andmebaasi skeem.



Joonis 2. Andmebaasi skeem.

Tabelis 4 on esitatud prototüübi põhilised andmebaasi olemid ja nende roll süsteemis.

Tabel 4. Andmebaasi põhilised olemid.

Olem	Kirjeldus
users	Hoiab kasutajate kontosid, rolle ja autentimiseks vajalikke andmeid.
refresh_sessions	Hoiab räsitud värskendustokenite sessioone, aegumisae- ga ja tühistamise infot.
groups	Hoiab õpetajate loodud õppegrupe ja grupiga seotud kutsekoodi.
group_members	Seob õppijad gruppidega ning talletab grupiga liitumise aja.
topics	Hoiab platvormi tasemel toetatud õppeteemade kataloo- gi.
group_topics	Määrab, millised teemad on konkreetses grupis aktiivsed.
task_instances	Hoiab õppijale moodustatud ülesande eksemplare koos raskusastme ja õige vastusega.
task_attempts	Hoiab õppija esitatud vastuseid ja nende õigsust.
student_topic_progress	Hoiab õppija teemakohast adaptiivset seisundit ja koond- statistikat.

Kasutajate ja gruppide seos on lahendatud tabeliga `group_members`, sest üks õppija võib kuuluda mitmesse gruppi ning ühes grupis võib olla mitu õppijat. Igal grupil on üks omanik ehk õpetaja, kellele viitab `groups.teacher_id`. Selline mudel võimaldab õpetajal hallata ainult enda grupe ning õppijal näha ainult neid grupe, kuhu ta kuulub.

Õppeteemad on eraldatud kaheks tasemeks. Tabel `topics` kirjeldab platvormil toetatud teemasid, näiteks trigonomeetriat ning kombinatoorikat ja tõenäosust. Tabel `group_topics` seob teemad konkreetsete gruppidega ning võimaldab õpetajal teemasid grupipõhiselt aktiveerida või deaktiveerida. Seetõttu ei pea iga grupp omama eraldi teemakirjeldusi, kuid õpetaja saab määrata, millised teemad on tema grupis kasutatavad.

Ülesannete lahendamise andmed on jagatud kaheks. Tabel `task_instances` hoiab süsteemi poolt moodustatud ülesannet kindlas õppija, grupi ja teema kontekstis. Tabel `task_`

`attempts` hoiab õppija esitatud vastust. Selline eraldus võimaldab hiljem näha, milline ülesanne õppijale anti ja mida ta sellele vastas. Samuti võimaldab see vältida sama ülesande korduvat esitamist, sest `task_attempts.task_instance_id` on unikaalne.

Õppija adaptiivne seisund salvestatakse tabelis `student_topic_progress`. See tabel kasutab primaarvõtmena õppija ja teema kombinatsiooni, sest õppija tase võib erineda teemade kaupa. Näiteks võib õppija olla trigonomeetrias kõrgemal tasemel kui kombinaatorikas. Tabelis hoitakse praegust raskusastet, adaptiivset reitingut, katsete arvu, õigete vastuste arvu, täpsust ja järjestikuste vastuste infot.

Füüsilises andmebaasis leiduvad ka varasemast arendusetapist pärit tabelid `tasks` ja `attempts`. Need ei kuulu prototüübi põhilisse grupi- ja teemakesksesse õppimisvoogu ning neid ei kasutata õppija praeguse ülesannete lahendamise voo osana. Tabelid jäeti migratsioonides alles, et säilitada skeemi arendusajalugu ja vältida varasemate migratsioonide ümberkirjutamist.

#### **4.4 API kavandamine**

Taustsüsteemi API kavandati REST-põhisena. API eesmärk on pakkuda kasutajaliidesele selgelt eraldatud ligipääsu autentimisele, kasutajaandmetele, gruppidele, teemadele, ülesannetele ja analüütikale. Päringute juures kontrollitakse kasutaja autentimist, rolli ja seost vastava grupiga.

API on jaotatud loogilisteks rühmadeks. Tabelis 5 on toodud API peamised rühmad ja nende ülesanded.

Tabel 5. API peamised rühmad.

Rühm	Ülesanne
Autentimine	Kasutaja registreerimine, sisselogimine, ligipääsutokeni uuendamine ja väljalogimine.
Kasutaja Grupid	Kasutaja profiili ja parooli muutmisega seotud toimingud. Õppegruppide loomine, muutmine, vaatamine, kutsekoodi haldamine ja liikmesuse haldamine.
Teemad	Platvormi teemade kuvamine ning teemade aktiveerimine või deaktiveerimine grupis.
Ülesanded	Järgmise ülesande moodustamine ja õppija vastuse esitamine.
Analüütika	Grupi, teema ja õppija tulemuste ülevaadete kuvamine õpetajale ning edenemise kuvamine õppijale.

Autentimise kavandamisel kasutatakse lühikese elueaga ligipääsutokenit ja HTTP-only küpsises hoitavat värskendustokenit. Andmebaasis ei salvestata värskendustokenit avatekstina, vaid selle räsi. Värskendustokeni uuendamisel tühistatakse vana sessioon ja luuakse uus. Selline lahendus võimaldab toetada väljalogimist, sessiooni aegumist ja tokeni rotatsiooni.

Gruppide API puhul on oluline, et õpetaja ja õppija tegevused oleksid eraldatud. Õpetaja saab luua gruppe, muuta nende andmeid, hallata kutsekoodi, vaadata liikmeid ja eemaldada õppijaid. Õppija saab liituda grupiga kutsekoodi abil ning lahkuda grupist. Iga grupiga seotud päringu puhul kontrollitakse, kas õpetaja on selle grupi omanik või kas õppija kuulub sellesse gruppi.

Ülesannete API keskne osa on järgmise ülesande küsimine ja vastuse esitamine. Järgmise ülesande küsimisel ei saadeta kasutajaliidesest raskusastet käsitsi, vaid taustsüsteem määrab selle õppija teemakohase edenemise põhjal. Vastuse esitamisel kontrollitakse, kas ülesanne kuulub samale õppijale ja kas sellele ei ole varem vastatud. Alles seejärel salvestatakse vastus ja uuendatakse õppija edenemine.

Analüütika API kavandati õpetaja töö toetamiseks. Õpetaja saab vaadata grupi üldist edenemist, konkreetse teema tulemusi ja üksiku õppija tulemusi. Õppija vaates kasutatakse API-t tema enda edenemise kuvamiseks, kuid õppija ei pääse ligi teiste õppijate tulemustele.

## 4.5 Adaptiivse algoritmi kavandamine

Adaptiivse algoritmi eesmärk on valida õppijale sobiva raskusastmega ülesanne ning uuendada õppija taset pärast iga vastust. Prototüübis kasutatakse kolme raskusastet: *easy*, *medium* ja *hard*. Igal õppijal on iga teema kohta eraldi edenemise kirje, sest õppija oskused võivad teemade lõikes erineda.

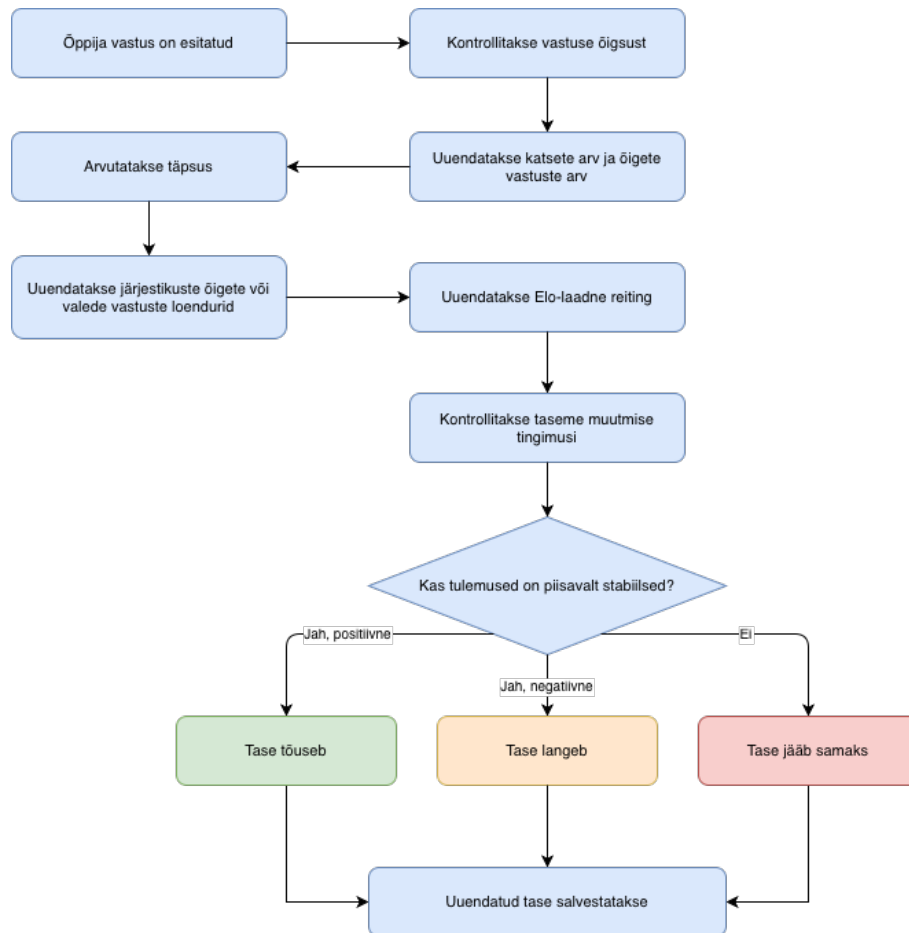
Adaptiivse voo alguses otsitakse õppija ja valitud teema põhjal kirje tabelist `student_topic_progress`. Kui seda veel ei ole, alustatakse vaikimisi lihtsaimast tasemest ja algsest adaptiivsest reitingust. Selle põhjal moodustatakse õppijale ülesanne valitud teema ja praeguse raskusastme järgi. Moodustatud ülesanne salvestatakse tabelisse `task_instances`.

Ülesande esitamisel võrreldakse õppija vastust ülesande õige vastusega. Vastuse kontrollimisel arvestatakse ülesande oodatud vormingut, sest eri teemade ülesanded võivad eeldada erinevat vastusekuju. Näiteks võib vastus olla arv, nurk, nurkade loend või muu struktureeritud väärtus. Kontrolli tulemus salvestatakse tabelisse `task_attempts`.

Pärast vastuse kontrollimist uuendatakse õppija teemakohast edenemist. Süsteem suurendab katsete arvu, uuendab õigete vastuste arvu, arvutab täpsuse ning muudab järjestikuste õigete või valede vastuste loendureid. Lisaks uuendatakse Elo-laadset adaptiivset reitingut, kus õppija vastuse tulemus mõjutab tema hinnangulist taset.

Raskusastme muutmisel ei kasutata ainult reitingut. Prototüübis kombineeritakse Elo-laadset reitingut lihtsate reeglitega, mis arvestavad katsete arvu, täpsust ja järjestikuseid õigeid või valesid vastuseid. Selline lahendus vähendab olukorda, kus üks juhuslik õige või vale vastus muudaks õppija taset liiga järsult. Samuti muudab see algoritmi käitumise arusaadavamaks ja paremini kontrollitavaks.

Joonisel 3 on esitatud adaptiivse taseme uuendamise voog pärast õppija vastuse esitamist.



Joonis 3. Adaptiivse taseme uuendamise voog.

Adaptiivne voog on kokkuvõtlikult järgmine:

1. õppija valib aktiivse teema ja küsib järgmise ülesande;
2. süsteem kontrollib õppija ligipääsu grupile ja teemale;
3. süsteem loeb õppija teemakohase edenemise;
4. süsteem valib praeguse raskusastme põhjal ülesande;
5. moodustatud ülesanne salvestatakse andmebaasi;
6. õppija esitab vastuse;
7. süsteem kontrollib vastust ja salvestab katse;
8. süsteem uuendab õppija statistikat, adaptiivset reitingut ja vajadusel raskusastet.

Sellise kavandiga toetab süsteem prototüübi põhieesmärki: õppija ei saa juhuslikku või kõigile ühesugust ülesannet, vaid ülesande raskusaste sõltub tema senisest sooritusest konkreetses teemas. Kavandatud adaptiivne voog seob ülesande moodustamise, vastuse

kontrollimise ja õppija edenemise uuendamise üheks terviklikuks protsessiks. Selle kavandi põhjal realiseeriti prototüübis nii tehniline arhitektuur, andmemudel, API kui ka adaptiivne mehhanism.

## 5 Prototüübi realiseerimine

Prototüübi realiseerimisel lähtuti eelmises peatükis 4 kirjeldatud kavandist. Realiseerimise eesmärk oli luua terviklik veebirakendus, kus õppija saab lahendada teemakohaseid harjutusülesandeid ning õpetaja saab hallata gruppe, õppeteemasid ja tulemuste ülevaateid. Süsteem realiseeriti eraldatud taustsüsteemi, kasutajaliidese ja andmebaasiga, et eri vastutusosalad oleksid selgelt lahutatud ning lahendust oleks võimalik hiljem laiendada.

### 5.1 Taustsüsteemi realiseerimine

Taustsüsteem on realiseeritud Pythonis FastAPI raamistikuga. Backend'i kirjeldamisel on oluline eristada nelja vastutusala: API päringute käsitlemine, andmekihi kasutamine, autentimine ja rollipõhine ligipääs. Need osad töötavad koos, kuid täidavad erinevaid ülesandeid süsteemi terviklikus töövoos.

Rakendus on jaotatud ruuteriteks, skeemideks, andmemudeliteks, andmebaasioperatsioonideks ja tuumikloogikaks. Selline jaotus aitab hoida API päringute käsitlemise, andmete valideerimise, andmebaasiga suhtlemise ja adaptiivse loogika eraldi kihtides. FastAPI rakenduse käivituspunktis registreeritakse autentimise, kasutajate, gruppide ja teemade ruuterid. Lisaks seadistatakse CORS, et arenduskeskkonnas saaks kasutajaliides taustsüsteemiga suhelda.

API päringud kasutavad Pydantic skeeme, mille kaudu määratakse sisend- ja väljundandmete kuju. See vähendab olukordi, kus kasutajaliides ja taustsüsteem tõlgendavad samu andmeid erinevalt. Andmebaasiga suhtlemiseks kasutatakse SQLAlchemy mudeleid ja asünkroonseid andmebaasisessioone. Mudelid kirjeldavad kasutajaid, gruppe, teemasid, ülesande eksemplare, vastuseid, õppija edenemist ja värskendustokeni sessioone. Andmebaasi struktuuri muutmiseks kasutatakse Alembic migratsioone, mis võimaldavad skeemi arenduse käigus järk-järgult täiendada.

Autentimine realiseeriti ligipääsutokenite ja värskendustokenite abil. Kasutaja sisselogi-

misel väljastatakse lühikese elueaga ligipääsutoken ning värskendustoken salvestatakse kliendis HTTP-only küpsises. Andmebaasis hoitakse ainult värskendustokeni räsi. Tokeni uuendamisel tühistatakse vana sessioon ja luuakse uus, mis võimaldab toetada tokeni rotatsiooni ja väljalogimist.

Rollipõhine ligipääs realiseeriti taustsüsteemi päringute tasemel. Õpetaja õigused kontrollitakse näiteks gruppide loomisel, muutmisel, liikmete vaatamisel ja teemade aktiveerimisel. Õppija õigused kontrollitakse grupiga liitumisel, aktiivsete teemade vaatamisel ja ülesannete lahendamisel. Grupiga seotud päringute puhul kontrollitakse lisaks rollile ka seda, kas õpetaja on grupi omanik või kas õppija kuulub vastavasse gruppi.

## 5.2 Kasutajaliidese realiseerimine

Kasutajaliides realiseeriti Reacti ja TypeScriptiga. Rakenduse marsruutimine põhineb React Routeril ning vaated on jaotatud avalikeks, kaitstud ja rollipõhisteks lehtedeks. Avalikud vaated on sisselogimine ja registreerimine. Kaitstud vaated eeldavad autentimist ning rollipõhised vaated piiravad ligipääsu õppija või õpetaja rolli alusel.

Rakenduse keskne autentimisolek on koondatud autentimiskonteksti. Selle abil saab kasutajaliides teada, kas kasutaja on sisse logitud, milline on tema roll ning millisele töölauale ta suunata. Kui kasutaja liigub üldisele töölaua aadressile, suunatakse ta rolli põhjal õppija või õpetaja vaatesse.

Taustsüsteemiga suhtlemiseks kasutatakse teenuseklasse ja ühist API klienti. API klient lisab päringutele ligipääsutokeni ning käsitleb olukorda, kus token on aegunud. Sellisel juhul proovib kasutajaliides värskendustokeni abil saada uue ligipääsutokeni ja kordab esialgset päringut. See vähendab vajadust kasutajat katkestada iga kord, kui ligipääsutoken aegub.

Kasutajaliidese loogika on jaotatud lehtedeks, komponentideks, teenusteks ja kohandatud *hook*'ideks. Lehed vastavad suurematele kasutusvaadetele, näiteks õppija töölaud, õpetaja töölaud, grupi detailvaade, teema vaade ja õppija analüütika. Komponentid kuvavad korduvkasutatavaid osi, näiteks gruppe, profiili, kutsekoodi, teemade haldust, edenemist ja analüütikat. Kohandatud Reacti *hook*'id koondavad andmete laadimise, vormide esitamise ja kasutajatoimingutega seotud oleku.

Kasutajaliidese vormides kasutatakse skeemipõhist valideerimist. See aitab kontrollida sisendi kuju juba enne päringu saatmist taustsüsteemi. Samas jääb lõplik valideerimine taustsüsteemi ülesandeks, sest ainult server saab usaldusväärselt kontrollida kasutaja õigusi, parooli korrektsust, kutsekoodi kehtivust või ülesande kuulumist konkreetsele õppijale.

### **5.3 Õppija vaate realiseerimine**

Õppija vaade keskendub gruppidega liitumisele, aktiivsete teemade vaatamisele, ülesannete lahendamisele ja oma edenemise jälgimisele. Pärast sisselogimist näeb õppija töölauda, kus kuvatakse tema grupid. Uue grupiga liitumiseks sisestab õppija õpetajalt saadud kutsekoodi. Kui kutsekood on kehtiv ja aktiivne, lisatakse õppija vastava grupi liikmeks.

Grupi detailvaates näeb õppija selle grupi aktiivseid õppeteemasid. Teemad kuvatakse ainult siis, kui õpetaja on need grupis aktiveerinud. See realiseerib peatükis 3 kirjeldatud grupipõhise teemade ligipääsu: õppija ei vali kogu platvormi teemakataloogist, vaid näeb ainult oma õppegrupi jaoks avatud teemasid.

Teema vaates saab õppija küsida järgmist ülesannet. Kasutajaliides ei määra ise ülesande raskusastet, vaid saadab taustsüsteemile päringu järgmise ülesande saamiseks. Taustsüsteem määrab õppija praeguse teemakohase taseme, moodustab sobiva ülesande ja tagastab selle kasutajaliidesele. Ülesande juures kuvatakse küsimus, raskusaste ja vajadusel vihje oodatud vastusevormingu kohta.

Vastuse esitamisel saadab kasutajaliides õppija sisestatud vastuse taustsüsteemile. Pärast vastuse kontrollimist kuvatakse õppijale, kas vastus oli õige, ning vajadusel ka korrektne vastus. Seejärel uuendatakse õppija edenemise andmed, et kasutaja näeks oma katsete arvu, õigete vastuste arvu, täpsust ja praegust taset. Selline voog toetab kiiret tagasisidet ja võimaldab õppijal jätkata järgmise ülesandega.

Õppija edenemist kuvatakse teemakohaselt. See on oluline, sest adaptiivne tase ei ole kasutaja üldine omadus, vaid sõltub konkreetsest teemast. Sama õppija võib olla ühes teemas lihtsal tasemel ja teises teemas keskmisel või raskel tasemel.

## 5.4 Õpetaja haldusvaate realiseerimine

Õpetaja vaade keskendub gruppide, liikmete, õppeteemade ja tulemuste haldamisele. Õpetaja töölaud kuvab õpetaja loodud grupe ning võimaldab luua uusi grupe. Grupi loomisel määratakse nimi ja soovi korral kirjeldus. Süsteem loob grupile kutsekoodi, mida õpetaja saab õppijatega jagada.

Grupi detailvaates saab õpetaja näha grupi liikmeid ja kutsekoodi. Kutsekoodi saab aktiveerida või deaktiveerida ning vajadusel uuendada. See võimaldab õpetajal kontrollida, millal õppijad saavad grupiga liituda. Kui õppija ei peaks enam gruppi kuuluma, saab õpetaja ta grupist eemaldada.

Õppeteemade haldus realiseeriti grupipõhiselt. Õpetaja saab näha platvormil olemasolevaid teemasid ning määrata, millised neist on konkreetses grupis aktiivsed. Teema aktiveerimisel tekib või uuendatakse seos grupi ja teema vahel. Deaktiveerimisel jääb teema platvormi kataloogi alles, kuid õppijad ei saa seda selles grupis harjutamiseks kasutada.

Õpetaja analüütikavaated annavad ülevaate grupi, teema ja üksiku õppija tulemustest. Grupi tasemel saab õpetaja näha koondnäitajaid, näiteks katsete arvu ja keskmist täpsust. Teema tasemel saab võrrelda õppijate edenemist konkreetses teemas. Üksiku õppija vaates kuvatakse õppija teemade kaupa tulemused ja lahendamise ajalugu.

Analüütika eesmärk ei ole anda täielikku pedagoogilist hinnangut õppijale, vaid pakkuda õpetajale kiiret ülevaadet sellest, millised teemad või õppijad vajavad rohkem tähelepanu. Kuna prototüübis salvestatakse iga vastus koos teema, grupi, raskusastme ja õigsusega, saab nende andmete põhjal koostada nii koondstatistikat kui ka detailsemaid vaateid.

## 5.5 Adaptiivse mootori realiseerimine

Adaptiivne mootor realiseeriti taustsüsteemi tuumikloogikas. Selle ülesanne on siduda õppija senine sooritus järgmise ülesande raskusastmega. Mootor kasutab õppija ja teema põhise edenemise kirjet, milles hoitakse praegust raskusastet, adaptiivset reitingut, katsete arvu, õigete vastuste arvu, täpsust ning järjestikuste õigete ja valede vastuste arvu.

Ülesannete moodustamine toimub teemaspetsiifiliste generaatorite abil. Prototüübis on

eraldi loogika trigonomeetria ning kombinatoorika ja tõenäosuse ülesannete jaoks. Iga generaator moodustab vastava raskusastmega ülesandekandidaatide kogumi, mille hulgast valitakse õppijale järgmine ülesanne. Ülesande kirjeldus sisaldab küsimust, alatüüpi, oodatud vastusevormingut ja õiget vastust. See võimaldab taustsüsteemil hiljem õppija vastust automaatselt kontrollida.

Järgmise ülesande valimisel arvestatakse õppija praegust taset. Kui õppijal puudub varasem edenemise kirje, alustatakse lihtsaimast tasemest. Kui varasem kirje on olemas, kasutatakse seal salvestatud raskusastet. Süsteem arvestab ka hiljuti kuvatud küsimusi, et vähendada sama küsimuse kordumist lühikese aja jooksul.

Vastuste kontrollimine realiseeriti oodatud vastusevormingu põhjal. Süsteem toetab mitut tüüpi vastuseid, sealhulgas täisarve, murde, kraadi- ja radiaanivormingus nurki, nurkade loendeid, tekstivalikuid, märgiga seotud vastuseid ning tekstina esitatud avaldise, sealhulgas ruutuurt sisaldavaid vastuseid. Vastuse kontrollimisel normaliseeritakse sisendit, et vähendada vormistuslike erinevuste mõju. Näiteks saab võrrelda arvulisi vastuseid ja murde ning trigonomeetria ülesannete puhul arvestada kraadi- või radiaanivorminguga.

Pärast iga vastust uuendatakse õppija edenemist. Kui vastus on õige, suureneb õigete vastuste arv ja järjestikuste õigete vastuste loendur. Vale vastuse korral suureneb järjestikuste valede vastuste loendur. Lisaks arvutatakse uuesti täpsus ja uuendatakse Elo-laadne adaptiivne reiting. Seejärel määratakse, kas õppija peaks jääma samale tasemele või liikuma lihtsamale või keerulisemale tasemele.

Raskusastme muutmise reeglid kasutavad korraka mitut näitajat. Näiteks kõrgemale tasemele liikumine eeldab piisavat katsete arvu, head täpsust, järjestikuseid õigeid vastuseid ja piisavat adaptiivset reitingut. Madalamale tasemele liikumine võib toimuda siis, kui õppija teeb mitu viga järjest või kui täpsus ja reiting langevad alla määratud piiri. Selline kombinatsioon muudab adaptiivse käitumise stabiilsemaks kui ainult ühe näitaja kasutamine.

## **5.6 Rakenduse käivitamine ja konteineriseerimine**

Rakenduse käivitamiseks kasutati Dockerit ja Docker Compose'i. Konteineriseerimine võimaldab käivitada andmebaasi, taustsüsteemi ja kasutajaliidese ühtse konfiguratsiooniga. See vähendab käsitsi seadistamise vajadust ning aitab tagada, et arenduskeskkond oleks eri

masinates võimalikult sarnane.

Arenduskeskkonna konfiguratsioonis käivitatakse PostgreSQL andmebaas, FastAPI taustsüsteem ja Vite arendusserveriga Reacti kasutajaliides. Taustsüsteem rakendab käivitamisel andmebaasimigratsioonid ning seejärel käivitab API serveri. Kasutajaliidese konteiner paigaldab vajalikud sõltuvused ja käivitab arendusserveri, mille kaudu saab rakendust brauseris kasutada.

Tootmislaadsema käivitamise jaoks on eraldi Docker Compose konfiguratsioon, kus kasutajaliides ehitatakse staatilisteks failideks ja serveeritakse veebiserveri kaudu. Taustsüsteem suhtleb PostgreSQL andmebaasiga sisemise teenusenime kaudu ning vajalikud saladused, näiteks andmebaasi parool, rakenduse salajane võti ja õpetaja registreerimiskood, antakse keskkonnamuutujate kaudu.

Konteineriseeritud ülesehitus toetab prototüübi testimist ja edasiarendamist. Kuna süsteemi põhiosad on teenustena eraldatud, saab vajadusel muuta või täiendada ühte osa ilma kogu rakendust ümber kujundamata. Näiteks saab laiendada taustsüsteemi ülesannete moodustamise loogikat, täiendada kasutajaliidest uute vaadetega või muuta andmebaasi skeemi migratsioonide abil.

Kokkuvõttes realiseeriti prototüüp kihilise veebirakendusena, kus kasutajaliides, taustsüsteem ja andmebaas täidavad eraldatud ülesandeid. Selline teostus võimaldas siduda õppija ülesande lahendamise voo, õpetaja haldusfunktsiooni ja adaptiivse raskusastme muutmise üheks töötavaks tervikuks.

## 6 Testimine ja tulemuste analüüs

### 6.1 Testimise eesmärk ja metoodika

Testimise eesmärk oli hinnata, kas loodud prototüüp vastab töö eesmärgile ja peatükis 3 määratletud nõuetele. Selleks kontrolliti ühelt poolt süsteemi tehnilist toimimist ning teiselt poolt seda, kuidas sihtrühma esindajad prototüüpi kasutavad ja hindavad. Tehnilise testimise käigus kontrolliti süsteemi põhifunktsionaalsust, API päringuid ja adaptiivse mehhanismi käitumist. Kasutatavuse hindamisel keskenduti õppija ja õpetaja vaadete arusaadavusele, ülesannete lahendamise protsessile, raskusastme kohandamise tajutavusele ning tulemuste ülevaate kasulikkusele.

Kasutatavuse hindamine viidi läbi Kohtla-Järve Gümnaasiumis kolmes 11. klassi matemaatikatunnis. Tunnid viisid läbi töö autorid ning iga tunni juures osales ka vastava klassi matemaatikaõpetaja. Testimise käigus kasutasid õppijad prototüüpi matemaatika harjutusülesannete lahendamiseks. Õpetajad tutvusid õpetaja vaatega, sealhulgas gruppide, õpeteemade ja õppijate tulemuste ülevaatega.

Testimiseks valiti 11. klassid, sest nende matemaatikakursus sobis prototüübis käsitletud teemadega kõige paremini. 10. klassi õppijad ei olnud testimise ajaks veel kõiki vajalikke teemasid läbinud, samal ajal kui 12. klassi õppijad keskendusid lõpueksamiteks valmistumisele. Seetõttu võimaldas 11. klasside kaasamine hinnata prototüüpi õppijatega, kellel oli ülesannete lahendamiseks vajalik eelteadmine olemas, ilma et testimine oleks oluliselt seganud lõpueksamiteks valmistumist.

Pärast prototüübi kasutamist täitsid õppijad ja õpetajad Google Forms küsimustiku. Õppijate küsimustik keskendus platvormi arusaadavusele, ülesannete lahendamise protsessile, ülesannete tekstidele, raskusastme sobivusele ja platvormi kasutamise valmisolekule. Õpetajate küsimustik keskendus õpetaja vaate loogilisusele, gruppide ja teemade haldamisele, õppijate tulemuste jälgimisele ning adaptiivse lähenemise sobivusele gümnaasiumiastme matemaatika harjutamiseks. Küsimustikule vastas 42 õppijat ja 3 õpetajat. Õppijate vas-

tuseid analüüsiti arvuliselt ja avatud vastuste põhjal, õpetajate vastuseid käsitleti väikese osalejate arvu tõttu eelkõige eksperthinnangutena.

## **6.2 Funktsionaalne testimine**

Funktsionaalse testimise eesmärk oli kontrollida, kas prototüübi põhivood töötavad terviklikult õppija ja õpetaja rollis. Testimisel lähtuti peatükis 3 kirjeldatud funktsionaalsetest nõuetest ning peatükis 4 kavandatud kasutusvoogudest.

Õppija rollis kontrolliti, kas kasutaja saab süsteemi sisse logida, liituda grupiga kutsekoodi abil, näha grupis aktiivseid teemasid, lahendada ülesandeid ning vaadata oma edenemist. Õpetaja rollis kontrolliti, kas õpetaja saab luua ja hallata gruppe, kasutada kutsekoodi, aktiveerida õppeemasid ning vaadata õppijate tulemuste ülevaadet.

Testimise põhjal töötasid prototüübi põhifunktsioonid kavandatud viisil. Õppija nägi ainult nende gruppide ja teemadega seotud sisu, millele tal oli ligipääs. Õpetaja sai hallata enda loodud gruppe ning vaadata nendega seotud õppijate tulemusi. Ülesande lahendamise voos kontrollis süsteem vastust automaatselt, salvestas katse ja uuendas õppija teemakohast edenemist.

Funktsionaalne testimine oli oluline eeldus kasutatavuse hindamisele. Kui põhivood ei töötaks usaldusväärselt, ei oleks õppijate ja õpetajate tagasiside põhjal võimalik hinnata adaptiivse õppeplatvormi kasutatavust ega praktilist sobivust.

## **6.3 API testimine**

API testimine keskendus sellele, kas taustsüsteem rakendab kasutajaliidese päringute töötlemisel vajalikke kontrollimehhanisme. Kuna prototüübis on õppija ja õpetaja õigused erinevad, oli oluline kontrollida, et rollipõhine ligipääs ei sõltuks ainult kasutajaliidese, vaid oleks jõustatud ka serveri poolel.

Autentimise puhul kontrolliti registreerimist, sisselogimist, väljalogimist ning tokenipõhist ligipääsu kaitstud päringutele. Gruppide ja teemadega seotud päringute puhul kontrolliti, et õpetaja saaks muuta ainult enda gruppe ning õppija saaks ligi ainult nendele teemadele, mis olid tema grupis aktiivsed. Ülesannete puhul kontrolliti, et õppija ei saaks küsida ülesannet

mitteaktiivse teema kohta ega esitada vastust ülesandele, mis ei kuulu talle.

API testimine täiendas funktsionaalset testimist. Funktsionaalne testimine näitas, et kasutaja põhivood töötavad kasutajaliideses, API testimine aga kontrollis, et nende voogude taga olevad serveripoolsed reeglid on korrektselt rakendatud. See on oluline, sest õppijate tulemused ja grupiandmed peavad olema kaitstud ka juhul, kui päringuid tehakse otse API vastu.

## 6.4 Adaptiivse algoritmi testimine

Adaptiivse algoritmi testimine oli töö seisukohalt üks oluliseim osa, sest prototüübi keskne eesmärk oli kohandada ülesannete raskusastet õppija seniste tulemuste põhjal. Testimisel kontrolliti, kas süsteem seob õppija vastused tema teemakohase edenemisega ning kasutab seda infot järgmise ülesande raskusastme määramisel.

Prototüübis hoitakse õppija edenemist iga teema kohta eraldi. See tähendab, et sama õppija võib olla ühes teemas madalamal ja teises teemas kõrgemal tasemel. Selline lahendus on oluline matemaatika õppimisel, sest teadmised ei arene kõigis teemades ühtlaselt. Testimisel kontrolliti, et uue teema puhul alustab õppija lihtsamast tasemest ning pärast vastuste esitamist uuendatakse tema teemakohast statistikat ja adaptiivset reitingut.

Tabel 6. Adaptiivse mehhanismi testimisel kontrollitud olukorrad.

Kontrollitud olukord	Oodatud tulemus
Õppijal puudub valitud teemas varasem edenemine	Süsteem alustab lihtsaimast raskusastmest
Õppija vastab järjest õigesti	Katsete arv, õigete vastuste arv, täpsus, seeria ja adaptiivne reiting uuenevad
Õppija teeb mitu viga järjest	Süsteem arvestab valevastuste seeriat ja võib suunata õppija lihtsamale tasemele
Õppija tulemused paranevad piisava arvu katsete järel	Süsteem võib liikuda kõrgemale raskusastmele
Õppija lahendab sama teemat korduvalt	Järgmise ülesande valikul kasutatakse õppija salvestatud teemakohast taset
Õppija lahendab eri teemasid	Edenemine säilib teemade kaupa eraldi

Testimise põhjal toimus adaptiivne mehhanism prototüübi eesmärgile vastavalt. Süsteem ei valinud ülesande raskusastet juhuslikult ega kasutajaliidesest saadud väärtuse põhjal, vaid lähtus õppija salvestatud edenemisest. Õige või vale vastus mõjutas õppija koondstatistikat ja Elo-laadset adaptiivset reitingut ning süsteem kasutas neid näitajaid raskusastme uuendamisel.

Oluline on rõhutada, et prototüübis kasutatav algoritm on teadlikult lihtsustatud. Selle eesmärk ei ole luua täielikku teadmiste mudelit iga matemaatilise alamteema kohta, vaid näidata, kuidas õppija soorituse põhjal saab ülesannete raskusastet praktiliselt kohandada. Testimine kinnitas, et selline mehhanism töötab prototüübi ulatuses ning on kasutajatele tajutav: õppijate tagasisides märgiti mitmel korral, et ülesanded muutusid järk-järgult raskemaks või et tasemed aitasid harjutamist suunata.

Lisaks raskusastme muutmisele kontrolliti ka ülesannete kordumise vähendamist. See ei ole adaptiivse algoritmi põhieesmärk, kuid mõjutab otseselt õppija kogemust. Kui süsteem pakub lühikese aja jooksul liiga sarnaseid ülesandeid, võib õppijale jääda mulje, et platvorm ei kohandu piisavalt. Seetõttu arvestatakse ülesande valikul hiljuti kuvatud küsimusi ja võimaluse korral välditakse sama alatüübi kordumist.

## **6.5 Kasutatavuse hindamine**

Õppijate tagasiside põhjal hinnati, kas prototüüp on sihtrühmale arusaadav ja kas see sobib matemaatika harjutusülesannete lahendamiseks. Tabelis 7 on esitatud 42 õppija vastuste koondtulemused.

Tabel 7. Õppijate hinnangud prototüübile.

Väide	Keskmine	Hinnang 4 või 5
Platvormi kasutamine oli arusaadav	4,74	97,6%
Vajalikud tegevused olid kergesti leitavad	4,88	97,6%
Gruppide ja teemade vaated olid arusaadavad	4,83	97,6%
Ülesannete lahendamise protsess oli selge	4,40	85,7%
Kujundus toetas õppimist ega seganud	4,74	97,6%
Ülesannete tekstid olid arusaadavad	4,14	78,6%
Ülesannete raskusaste tundus sobiv	4,43	90,5%
Soov kasutada platvormi edaspidi matemaatika harjutamiseks	4,12	81,0%
Platvorm sobiks tavapärase matemaatikaõppe toetamiseks	4,45	88,1%

Tabelis esitatud tulemused näitavad, et õppijad hindasid prototüübi üldist kasutatavust positiivselt. Kõige kõrgemalt hinnati vajalike tegevuste leidmist, gruppide ja teemade vaadete arusaadavust ning platvormi üldist kasutamist. See viitab sellele, et kasutajaliidese põhistruktuur toetas õppija tegevust ega tekitanud olulist lisakoormust.

Madalamad hinnangud olid seotud ülesannete tekstide ja lahendamise protsessiga. See tulemus on töö eesmärgi seisukohalt oluline, sest adaptiivse platvormi kvaliteet ei sõltu ainult tehnilisest raskusastme muutmisest. Kui ülesande sõnastus või vastuse oodatud vorming ei ole õppijale piisavalt selge, võib süsteem hinnata õppija vastust valesti või tekitada õppijas segadust. Avatud vastustes mainiti mitmel korral vajadust selgema sõnastuse, paindlikuma vastusevormingu ja põhjalikuma tagasiside järele.

Eraldi hinnati, kas õppijate arvates muutus ülesannete raskusaste loogiliselt. Vastuste jaotus on esitatud tabelis 8.

Tabel 8. Õppijate hinnang raskusastme muutumise loogilisusele.

Vastus	Vastuste arv
Jah	12
Pigem jah	26
Pigem ei	2
Ei oska öelda	2
Ei	0

Tabel 8 põhineb samal õppijate küsimustikul ning andmed on arvatatud 42 vastuse põhjal. Kokku 38 õppijat vastas, et raskusastme muutumine oli kas pigem loogiline või loogiline. See toetab järeldust, et adaptiivne raskusastme muutmine oli prototüübi kasutamisel õppijatele märgatav ja üldiselt mõistetav.

Õpetajate tagasiside oli samuti positiivne. Kõik kolm õpetajat hindasid õpetaja vaate arusaadavust, gruppide haldamist, kutsekoodi kasutamist, teemade aktiveerimist ja tulemuste ülevaadet hindegas 5. Samuti peeti õppija täpsuse, katsete arvu ja taseme kuvamist piisavaks esmaseks ülevaateks õppija edenemisest. Kuna õpetajate vastuseid oli kolm, ei ole nende põhjal võimalik teha laiapõhjalisi statistilisi üldistusi, kuid vastused annavad väärtusliku esmase hinnangu prototüübi sobivusele õpetaja töö toetamisel.

Tabel 9. Õpetajate hinnangute koondtulemused.

Väide	Keskmine
Õpetaja vaade oli arusaadav ja loogilise ülesehitusega	5,00
Gruppide loomine ja haldamine oli arusaadav	5,00
Kutsekoodi kasutamine õppijate gruppi lisamiseks oli sobiv lahendus	5,00
Õppeteemade aktiveerimine ja deaktiveerimine oli arusaadav	5,00
Õppijate tulemuste ülevaade oli kasulik	5,00
Õppija täpsuse, katsete arvu ja taseme kuvamine andis esmase ülevaate	5,00
Raskusastme kohandamine õppija tulemuste põhjal on kasulik	5,00
Prototüübi adaptiivne lähenemine sobib gümnaasiumiastme matemaatika harjutamiseks	4,67
Automaatne vastuste kontrollimine vähendaks käsitsi kontrollimise koormust	4,67
Platvorm võiks täiendada tavapärast matemaatikaõpet	4,67

Õpetajate avatud vastustes toodi kõige olulisemate edasiarendustena välja uute teemade ja ülesannete lisamine, tulemuste eksportimine, detailsem õppija analüütika, gruppide võrdlus ning õpetaja võimalus ülesandeid muuta. Lisaks mainiti vajadust jaotada suuremaid teemasid alateemadeks ja näha ülesannete lahendamiseks kulunud aega. Need ettepanekud on otseselt seotud õpetaja tööga: õpetaja ei vaja ainult üksiku vastuse õigsust, vaid ka infot selle kohta, millistes alamteemades või ülesandetiüpides õppijatel raskused tekivad.

## 6.6 Tulemuste analüüs

Testimise tulemused näitasid, et prototüüp vastab töö põhieesmärgile. Süsteem võimaldab õppijal lahendada matemaatika harjutusülesandeid, kontrollib vastuseid automaatselt, uuendab õppija teemakohast edenemist ning kohandab järgmiste ülesannete raskusastet. Õpetaja saab hallata gruppe ja teemasid ning jälgida õppijate tulemusi.

Kõige tugevamalt kinnitas testimine kasutajaliidese ja põhivoogude toimimist. Õppijate hinnangud platvormi arusaadavusele, navigeerimisele ja kujundusele olid kõrged. See tähendab, et prototüübi kasutamine ei olnud õppijate jaoks peamine takistus ning nad said

keskenduda ülesannete lahendamisele.

Töö keskse teema ehk adaptiivsuse seisukohalt oli oluline, et enamik õppijaid hindas raskusastme muutumist loogiliseks või pigem loogiliseks. See ei tõesta veel adaptiivse õppe mõju õpitulemustele, kuid näitab, et loodud mehhanism täitis prototüübi tasemel oma põhifunktsiooni: kõik õppijad ei saanud ühesugust fikseeritud ülesannete jada, vaid süsteem kasutas nende seniseid tulemusi järgmise raskusastme määramiseks.

Testimine tõi samal ajal välja ka nõrgad kohad. Kõige olulisemad probleemid olid seotud ülesannete sõnastuse, vastusevormingute ja tagasiside sisukusega. Need tähelepanekud on adaptiivse õppeplatvormi puhul olulised, sest adaptiivne algoritm saab teha õigeid otsuseid ainult siis, kui õppija vastus on korrektselt tõlgendatav. Kui süsteem ei aktsepteeri matemaatiliselt samaväärset vastust või kui ülesande tekst on ebaselge, võib probleem peituda mitte õppija teadmistes, vaid süsteemi sisendis või valideerimises.

Pärast kasutatavuse testimist tehti prototüübis väiksemaid parandusi. Laiendati aktsepteeritavaid vastusevorminguid, täpsustati osa kombinatoorika ülesannete sõnastust ning vähendati sarnaste ülesandetüüpide järjestikust kordumist. Suuremad edasiarendused, näiteks lahenduskäigu kuvamine, vihjete süsteem, õpetaja võimalus ülesandeid muuta ja detailsem alamteemade analüütika, jäid edasise arenduse osaks.

## **6.7 Lahenduse piirangud**

Prototüübi tulemusi tuleb tõlgendada selle mahu ja testimiskonteksti piires. Süsteem toetab piiratud hulka matemaatikateemasid ega kata kogu gümnaasiumi matemaatika õppekava. Seetõttu ei saa tulemuste põhjal väita, et lahendus sobib muutmata kujul kõigi matemaatikateemade õpetamiseks.

Teine piirang puudutab testimise kestust. Prototüüpi katsetati kolme matemaatikatunni jooksul, mis sobib esmase kasutatavuse ja töökindluse hindamiseks, kuid ei võimalda mõõta pikaajalist mõju õppija teadmiste jaoks. Selleks on vaja pikemat kasutusperioodi, kontrollitud õppeülesandeid ja õppijate tulemuste võrdlust enne ja pärast platvormi kasutamist.

Kolmas piirang on seotud adaptiivse mudeli lihtsustatusega. Prototüübis kasutatakse Elo-laadset reitingut ja reegleid raskusastme muutmiseks, kuid süsteem ei modelleeri veel

õppija teadmisi detailsete alamoskuste kaupa. Selline lahendus sobib prototüübi eesmärgiga, kuid edasises arenduses on vaja täpsemat teemade ja alamteemade põhise õppija mudelit.

Neljandaks vajab edasiarendamist õppijale antav tagasiside. Praegune prototüüp annab teada, kas vastus oli õige või vale, kuid ei paku veel piisavalt põhjalikku selgitust selle kohta, miks vastus oli vale või kuidas ülesannet lahendada. Kasutatavuse hindamise põhjal on see üks olulisemaid suundi, mis suurendaks platvormi õppeväärtust.

Kokkuvõttes näitas testimine, et loodud prototüüp täidab töö eesmärgi: süsteem realiseerib adaptiivse matemaatika harjutamise põhivoo ning toetab nii õppija kui ka õpetaja rolli. Testimine kinnitas kasutajaliidese ja põhifunktsioonide toimimist, kuid tõi välja ka selged edasiarenduse vajadused, eelkõige ülesannete mitmekesisuse ja sõnastuse, vastuste paindlikuma kontrollimise, põhjalikuma tagasiside ning detailsema õpetaja analüütika osas.

## 7 Edasiarenduse võimalused

Prototüübi arendamise ja testimise käigus selgus mitu suunda, mille kaudu saaks loodud lahendust edasi arendada. Kuna käesoleva töö eesmärk oli luua toimiv adaptiivse matemaatika õppeplatvormi prototüüp, keskendus lahendus piiratud hulga teemadele ja põhifunktsionaalsusele. Edasiarenduse eesmärk oleks muuta süsteem sisuliselt mitmekesisemaks, õppijale toetavamaks ja õpetajale analüütiliselt kasulikumaks.

Üks olulisemaid edasiarenduse suundi on õppeteemade ja ülesandetüüpide laiendamine. Praeguses prototüübis on toetatud piiratud arv matemaatikateemasid, mis võimaldas adaptiivse mehhanismi toimimist testida, kuid ei kata kogu gümnaasiumiastme matemaatika vajadusi. Süsteemi praktilise kasutatavuse suurendamiseks tuleks lisada uusi teemasid ning iga teema sees rohkem ülesandetüpe ja variatsioone. See vähendaks ülesannete kordumise tunnet ja võimaldaks platvormi kasutada pikema aja jooksul.

Teine oluline suund on õppijale antava tagasiside täiendamine. Praegune prototüüp annab õppijale teada, kas vastus oli õige või vale, ning kuvab vajadusel õige vastuse. Kasutatavuse hindamisel tõid õppijad aga välja, et vale vastuse korral oleks kasulik näha selgitust, vihjet või lahenduskäiku. Edasiarenduses võiks süsteem pakkuda lühikesi teemakohaseid vihjeid, näidata tüüpilisi lahendusvõtteid või selgitada, millise vea õppija võis teha. Selline tagasiside suurendaks platvormi õppeväärtust, sest õppija ei saaks ainult tulemust, vaid ka tuge vea mõistmiseks.

Kolmanda suunana võiks täiendada vastuste kontrollimist. Matemaatikaülesannetes võib sama vastust esitada mitmel kujul, näiteks murruna, kümnendmurruna, sümboolse avaldisena või trigonomeetrilise tähistusena. Kuigi prototüübis on juba arvestatud mitme vastusevorminguga, vajab süsteem edasises arenduses veel paindlikumat matemaatiliste avaldiste tõlgendamist. See aitaks vähendada olukordi, kus sisuliselt õige vastus loetakse valeks ainult vormistusliku erinevuse tõttu.

Õpetaja vaate edasiarendamisel oleks oluline lisada detailsem analüütika. Praegune lahendus

kuvab õppija täpsust, katsete arvu, taset ja teemakohast edenemist, kuid õpetaja jaoks oleks kasulik näha ka alamteemade või ülesandetüüpide kaupa tekkivaid raskusi. Näiteks võiks süsteem eristada, kas õppijal on probleem kombinatoorika valemite rakendamise, tõenäosuse arvutamise või trigonomeetriliste väärtuste teisendamisega. Selline analüütika aitaks õpetajal paremini otsustada, milliseid teemasid klassis korrata ja millised õppijad vajavad täiendavat tuge.

Lisaks võiks õpetaja vaatesse lisada tulemuste eksportimise võimaluse. See võimaldaks õpetajal salvestada grupi või õppija tulemused eraldi failina ning kasutada neid hilisemas analüüsis või õppetöö planeerimisel. Samuti võiks õpetajal olla võimalus ülesandeid muuta, lisada või vähemalt määrata, millised ülesandetüübid on konkreetses grupis aktiivsed. See annaks õpetajale suurema kontrolli selle üle, kuidas platvorm toetab konkreetse klassi õppeprotsessi.

Adaptiivse mehhanismi edasiarendamisel võiks liikuda täpsema õppija mudeli suunas. Praeguses prototüübis hoitakse õppija edenemist teemade kaupa ning raskusastet muudetakse Elo-laadse reitingu ja lihtsate reeglite alusel. Edasises arenduses võiks süsteem jälgida õppija oskusi ka alamteemade või konkreetsete pädevuste kaupa. See võimaldaks valida ülesandeid täpsemalt mitte ainult üldise raskusastme, vaid ka õppija konkreetsete nõrkuste põhjal.

Ühe võimaliku edasiarendusena võib kaaluda ka tehisintellektil põhinevat ülesannete loomist. Selline lahendus võiks aidata suurendada ülesannete hulka ja varieerida ülesannete sõnastust. Samas ei tohiks tehisintellektil põhinev ülesannete loomine toimuda ilma kontrollita. Matemaatikaülesannete puhul on oluline, et ülesande tingimus oleks korrektne, vastus üheselt kontrollitav ja raskusaste sobiv. Seetõttu peaks tehisintellekti kasutamisel säilima valideerimismehhanism, mis kontrollib ülesande matemaatilist korrektsust ja vastuse kontrollitavust.

Viimaks tuleks prototüüpi testida pikema aja jooksul reaalses õppetöös. Käesolevas töös tehtud kasutatavuse hindamine andis esmase tagasiside platvormi arusaadavuse ja sobivuse kohta, kuid ei mõõtnud pikaajalist mõju õppijate tulemustele. Edasises uuringus võiks platvormi kasutada mitme nädala või perioodi jooksul ning võrrelda õppijate edenemist enne ja pärast platvormi kasutamist. See võimaldaks hinnata mitte ainult süsteemi kasutatavust,

vaid ka selle võimalikku mõju matemaatika õppimisele.

Kokkuvõttes on loodud prototüüp sobiv alus edasiseks arenduseks. Kõige olulisemad edasiarenduse suunad on ülesannete hulga ja teemade laiendamine, õppijale antava tagasiside parandamine, vastuste paindlikum kontrollimine, õpetaja analüütika täiendamine ning adaptiivse mehhanismi täpsem sidumine õppija konkreetsete oskustega.

## 8 Kokkuvõte

Käesoleva bakalaureusetöö raames täideti edukalt püstitatud eesmärgid, mille tulemusena projekteeriti ja realiseeriti funktsionaalne adaptiivse matemaatika õppeplatvormi prototüüp. Loodud lahendus pakub õppijatele personaliseeritud harjutuskeskkonda, mis kontrollib vastuseid automaatselt ja kohandab ülesannete raskusastet reaajas vastavalt kasutaja senisele sooritusele. Lisaks integreeriti süsteemi terviklik õpetajamoodul, mis võimaldab tõhusalt hallata õppegrupe, aktiveerida vajalikke õppeteemasid ning jälgida reaajas õppijate individuaalset progressi ja tulemusi.

Töö käigus anti ülevaade e-õppe platvormidest, personaliseeritud õppest, adaptiivsetest õppesüsteemidest ja adaptiivses õppimises kasutatavatest algoritmilistest lähenemistest. Prototüübi jaoks valiti lihtne ja läbipaistev Elo-laadne lähenemine, sest see sobib piiratud mahuga prototüübi jaoks ning võimaldab õppija taset uuendada iga vastuse järel. Samuti määratleti süsteemi sihtrühm, kasutajarollid, funktsionaalsed ja mittefunktsionaalsed nõuded ning põhjendati valitud tehnoloogiaid.

Praktilise töö tulemusena kavandati ja realiseeriti kolmekihiline veebirakendus, mille taustsüsteem on loodud FastAPI raamistikuga, kasutajaliides Reacti ja TypeScriptiga ning andmekiht PostgreSQL-i abil. Süsteem toetab õppija ja õpetaja rolle, grupipõhist teemade haldamist, ülesannete moodustamist teemaspetsiifiliste generaatorite abil, automaatselt vastuste kontrollimist, õppija teemakohase edenemise salvestamist ja õpetajale tulemuste ülevaate kuvamist.

Prototüüpi testiti tehniliselt ning katsetati kolmes 11. klassi matemaatikatunnis. Kasutatavuse hindamisele vastas 42 õppijat ja 3 õpetajat. Tulemused näitasid, et õppijad hindasid platvormi üldist arusaadavust, navigeerimist ja kujundust positiivselt. Enamik õppijaid leidis, et ülesannete raskusaste muutus loogiliselt või pigem loogiliselt. Õpetajate tagasiside kinnitas, et grupihaldus, teemade aktiveerimine ja õppijate tulemuste ülevaade on praktiliselt vajalikud funktsioonid.

Testimine tõi samal ajal välja ka loodud lahenduse piirangud. Edasiarendust vajavad eelkõige ülesannete hulga ja mitmekesisuse suurendamine, ülesannete sõnastuse täpsustamine, vastusevormingute paindlikum kontrollimine, õppijale põhjalikuma tagasiside andmine ning õpetaja analüütikavaate täiendamine. Kokkuvõttes täitis töö seatud eesmärgi: valmis toimiv adaptiivse matemaatika õppeplatvormi esmane lahendus, mis demonstreerib õppija soorituse põhjal raskusastme kohandamist ning loob aluse süsteemi edasiseks arendamiseks.

## Kasutatud kirjandus

- [1] European Commission. *Digital Education Action Plan: policy background*. URL: <https://education.ec.europa.eu/focus-topics/digital-education/actions/plan> (vaadatud 06.05.2026).
- [2] OECD. *OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem*. Paris: OECD Publishing, 2023. DOI: 10.1787/c74f03de-en.
- [3] Benő Csapó ja Gyöngyvér Molnár. „Online Diagnostic Assessment in Support of Personalized Teaching and Learning: The eDia System“. *Frontiers in Psychology* 10 (2019), lk. 1522. DOI: 10.3389/fpsyg.2019.01522.
- [4] Sharon Klinkenberg, Marthe Straatemeier ja Han L. J. van der Maas. „Computer Adaptive Practice of Maths Ability Using a New Item Response Model for on the Fly Ability and Difficulty Estimation“. *Computers & Education* 57.2 (2011), lk. 1813–1824. DOI: 10.1016/j.compedu.2011.02.003.
- [5] UNESCO. *Global Education Monitoring Report 2023: Technology in Education: A Tool on Whose Terms?* Paris: UNESCO, 2023. URL: <https://www.unesco.org/gem-report/en/publication/technology> (vaadatud 06.05.2026).
- [6] Kelly Wauters, Piet Desmet ja Wim Van den Noortgate. „Adaptive Item-Based Learning Environments Based on the Item Response Theory: Possibilities and Challenges“. *Journal of Computer Assisted Learning* 26.6 (2010), lk. 549–562. DOI: 10.1111/j.1365-2729.2010.00368.x.
- [7] Radek Pelánek. „Bayesian Knowledge Tracing, Logistic Models, and Beyond: An Overview of Learner Modeling Techniques“. *User Modeling and User-Adapted Interaction* 27.3–5 (2017), lk. 313–350. DOI: 10.1007/s11257-017-9193-2.
- [8] Radek Pelánek. „Applications of the Elo Rating System in Adaptive Educational Systems“. *Computers & Education* 98 (2016), lk. 169–179. DOI: 10.1016/j.compedu.2016.03.017.
- [9] Moodle. *Open Source Online Learning*. URL: <https://moodle.com/about/open-source/> (vaadatud 06.05.2026).
- [10] Khan Academy. *How Do I Use MAP Recommended Practice on Khan Academy?* URL: <https://support.khanacademy.org/hc/en-us/articles/236355907-How-do-I-use-MAP-Recommended-Practice-Mappers-on-Khan-Academy> (vaadatud 06.05.2026).
- [11] IXL. *Personalized Learning Made Simple*. URL: <https://www.ixl.com/membership/teachers/how-it-works> (vaadatud 06.05.2026).

- [12] Matific. *Matific for Teachers*. URL: <https://www.matific.com/us/en-us/home/teachers/> (vaadatud 06.05.2026).
- [13] FastAPI. *FastAPI Documentation*. URL: <https://fastapi.tiangolo.com/> (vaadatud 07.05.2026).
- [14] Meta Open Source. *React Documentation*. URL: <https://react.dev/> (vaadatud 07.05.2026).
- [15] Microsoft. *TypeScript Documentation*. URL: <https://www.typescriptlang.org/docs/> (vaadatud 07.05.2026).
- [16] PostgreSQL Global Development Group. *JSON Types*. URL: <https://www.postgresql.org/docs/current/datatype-json.html> (vaadatud 07.05.2026).
- [17] Docker. *Docker Compose Documentation*. URL: <https://docs.docker.com/compose/> (vaadatud 07.05.2026).

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Meie, Dmitri Kurtukov ja Ruslan Melnikov

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Adaptiivse õppeplatvormi prototüübi arendus”, mille juhendaja on Siim Rebane
  - 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. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

31.05.2026

---

<sup>1</sup>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 oma ühise 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 – Õppija kasutatavuse küsimustik

### Adaptiivse matemaatika õppeplatvormi prototüübi kasutatavuse hindamine

Küsimustiku eesmärk on hinnata adaptiivse matemaatika õppeplatvormi prototüübi kasutatavust ja arusaadavust. Vastuseid kasutatakse bakalaureusetöö raames prototüübi analüüsimiseks ja edasiarenduse võimaluste hindamiseks. Küsimustik on anonüümne. Vastamine võtab umbes 5–7 minutit.

### Kasutatavus ja arusaadavus

Järgnevad väited hinnatakse skaalal 1–5, kus 1 = ei nõustu üldse, 5 = nõustun täielikult.

1. Platvormi kasutamine oli minu jaoks arusaadav.
2. Leidsin vajalikud tegevused platvormil kergesti üles.
3. Gruppidega ja teemadega seotud vaated olid arusaadavad.
4. Ülesannete lahendamise protsess oli selge ja arusaadav.
5. Platvormi kujundus toetas ülesannete lahendamist ega seganud õppimist.

### Ülesanded ja tagasiside

Väited skaalal 1–5, küsimus 8 on valikvastustega.

6. Ülesannete tekstid olid arusaadavad.
7. Ülesannete raskusaste tundus minu tasemele sobiv.
8. Kas ülesannete raskusaste muutus sinu arvates loogiliselt? (*Jah / Pigem jah / Pigem ei / Ei / Ei oska öelda*)

### Õppimise toetamine

Väited skaalal 1–5.

9. Sooviksin sellist platvormi kasutada ka edaspidi matemaatika harjutamiseks.
10. Platvorm sobiks minu arvates tavapärase matemaatikaõppe toetamiseks.

### **Avatud küsimused**

11. Mis platvormi kasutamisel meeldis kõige rohkem?
12. Mis tekitas platvormi kasutamisel segadust või raskusi?
13. Mida võiks platvormis parandada või juurde lisada?

## **Lisa 3 – Õpetaja kasutatavuse küsimustik**

### **Adaptiivse matemaatika õppeplatvormi prototüübi hindamine õpetaja vaates**

Küsimustiku eesmärk on hinnata adaptiivse matemaatika õppeplatvormi prototüüpi õpetaja vaates. Vastuseid kasutatakse bakalaureusetöö raames prototüübi kasutatavuse, funktsionaalsuse ja edasiarenduse võimaluste analüüsimiseks. Küsimustik on anonüümne. Vastamine võtab umbes 5–7 minutit.

### **Kasutatavus õpetaja vaates**

Järgnevad väited hinnatakse skaalal 1–5, kus 1 = ei nõustu üldse, 5 = nõustun täielikult.

1. Õpetaja vaade oli arusaadav ja loogilise ülesehitusega.
2. Gruppide loomine ja haldamine oli arusaadav.
3. Kutsekoodi kasutamine õppijate gruppi lisamiseks oli sobiv lahendus.
4. Õppeteemade aktiveerimine ja deaktiveerimine oli arusaadav.
5. Platvormi kasutajaliides sobib õpetaja igapäevase töö toetamiseks.

### **Analüütika ja õppijate jälgimine**

Väited skaalal 1–5.

6. Õppijate tulemuste ülevaade oli kasulik.
7. Teemade kaupa tulemuste kuvamine aitab õppijate arengut paremini jälgida.
8. Õppija täpsuse, katsete arvu ja taseme kuvamine annab piisava esmase ülevaate õppija edenemisest.
9. Platvorm aitaks märgata õppijaid, kes vajavad lisatuge.
10. Platvorm aitaks märgata teemasid, mis tekitavad grupile rohkem raskusi.

### **Adaptiivsus ja pedagoogiline väärtus**

Väited skaalal 1–5.

11. Ülesannete raskusastme kohandamine õppija tulemuste põhjal on matemaatika harjutamisel kasulik.
12. Prototüübi adaptiivne lähenemine sobib gümnaasiumiastme matemaatika harjutamiseks.
13. Automaatne vastuste kontrollimine vähendaks õpetaja käsitsi kontrollimise koormust.
14. Selline platvorm võiks täiendada tavapärast matemaatikaõpet.

### **Piirangud ja edasiarendus**

Küsimus 15 on mitme valikuga, küsimus 16 on valikvastustega.

15. Millised oleksid olulisimad edasiarendused? (*Mitme vastuse valik*)
  - Uute ülesannete või teemade lisamine
  - Tehisintellektil põhinev ülesannete loomine
  - Detailsem õppija analüütika
  - Klasside/gruppide võrdlus
  - Õpetaja võimalus ülesandeid muuta
  - Õppijale põhjalikum tagasiside
  - Tulemuste eksportimine
16. Kas praegune prototüüp oleks teie hinnangul sobiv katsetamiseks reaalses õppetöös?  
(*Jah / Pigem jah / Pigem ei / Ei / Ei oska öelda*)

### **Avatud küsimused**

17. Millised funktsioonid oleksid õpetaja vaates kõige olulisemad edasiarendused?
18. Mis oli prototüübi õpetaja vaates kõige kasulik osa?
19. Mis jäi õpetaja vaates ebaselgeks või ebamugavaks?
20. Milliseid andmeid või analüütikat sooviksite õpetajana veel näha?
21. Milliseid muudatusi soovitaksite enne platvormi kasutamist päris õppetöös?