

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Lisette Sirel 185739IABB

# **Veebilehe arendamine ettevõttele Bachata Studio Tallinn**

Bakalaureusetöö

Juhendajad: Gunnar Piho  
PhD  
Radoslav Glinsky

Tallinn 2020

## **Autorideklaratsioon**

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

Autor: Lisette Sirel

18.05.2021

## Annotatsioon

Antud lõputöö eesmärgiks on valmis saada veebileht ettevõttele Bachata Studio Tallinn, mis tegeleb ladina-ameerika tantsustiilide õpetamisega. Arendatud veebileht vähendab töötajate koormust ning teeb klientide elu mugavamaks. Antud projekt võimaldab edasi anda suurel hulgal informatsiooni tantsustuudio kohta, maksete tegemist ning veebilehe haldamist.

Veebileht kuvab informatsiooni stuudio, treenerite, kursuste, tantsupakettide, ürituste, korduma kippuvate küsimuste, tunniplaani ja kodukorra kohta. Kliendil on võimalus valida endale sobiv treeningpakett ning selle eest maksta. Tantsustuudio mäenedžerile valmis eraldi liides, kust on võimalik hallata kogu veebilehte - lisada, muuta ja kustutada informatsiooni.

Veebilehe *front-endi* ja *Admin* projekti arendamisel kasutati Visual Studio Code'i koos ReactJS raamistikuga ning *back-endi* jaoks Visual Studio ASP.NET Core raamistikku. Andmebaasiks valiti Microsoft Azure SQL andmebaas.

Projekt avalikustati Microsoft Azure Portali kaudu ning veebileht on kättesaadav kõikidele klientidele. Tulevikus võib tekkida vajadus veebilehe täiendamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 52 leheküljel, 5 peatükki, 41 joonist.

## **Abstract**

### **Website development for Bachata Studio Tallinn**

The aim of this thesis is to develop a website for Bachata Studio Tallinn, which teaches Latin American dance styles. This website reduces the workload of employees and makes customers' lives more comfortable. This project enables to give information about the dance studio, make payments and manage the website.

The website displays information about studio, instructors, courses, dance packages, events, frequently asked questions, timetable and privacy policy. The client has the opportunity to choose a training package that suits them and pay for it. A separate interface has been developed for the manager of the dance studio, from where it is possible to manage the entire website - add, change and delete information.

The website was developed using Visual Studio Code with ReactJS framework for the *front-end* and *Admin* project development. Visual Studio ASP.NET Core framework was used for *back-end* development. The Microsoft Azure SQL database was chosen as the database.

The project was published through the Microsoft Azure Portal and the website is available to all customers. There may be a need to update the website in the future.

The thesis is in Estonian and contains 52 pages of text, 5 chapters, 41 figures.

## Lühendite ja mõistete sõnastik

BST	<i>Bachata Studio Tallinn</i> , tantsustuudio
JavaScript	objektorienteeritud programmeerimiskeel veebilehede skriptimiseks
HTML	<i>Hypertext Markup Language</i> , märgistuskeel veebilehe sisu struktureerimiseks
CSS	<i>Cascading Style Sheets</i> , stiilireeglite keel HTML kujundamiseks
ReactJS	Javascripti raamistik kasutajaliideste loomiseks
API	<i>Application Programming Interface</i> , tarkvara vahendaja - võimaldab kahel rakendusel omavahel suhelda
C#	<i>C Sharp</i> , objektorienteeritud programmeerimiskeel turvaliste ja kindlate rakenduste arendamiseks .NET raamistikus
Azure Data Studio	platvormidevaheline andmebaasitööriist
<i>commit</i>	versiooni kontrolli süsteemi operatsioon, mis saadab viimased koodimuudatused lähtekoodi repositooriumisse
<i>milestone</i>	tööjaotuse etapp, mis teeb tööprotsessi jälgimise lihtsamaks
ASP.NET Core	raamistik veebirakenduste arendamiseks .NET platvormil
Azure Static Web App	teenus, mis ehitab ja juurutab veebirakendused automaatselt koodihoidlast Azure'i
Azure App Service	teenus veebirakenduste ja API-de avalikustamiseks internetti
Azure SQL Database	platvorm, mis haldab enamikku andmebaasihalduse funktsioone
Azure Portal	veebipõhine portaal, kus on võimalik hallata ja jälgida veebirakendusi
.NET	<i>raamistik</i> , Microsofti tarkvaraplatvorm ja integreeritud arenduskeskkond
HTTP	<i>Hypertext Transfer Protocol</i> , teabe edastamiseks veebis
CRUD	<i>create, read, update and delete</i> , programmeerimisel kasutatav tarkvara arhitektuurne stiil
Admin	kasutajaliides veebilehel oleva informatsiooni haldamiseks
REST	<i>Representational State Transfer</i> , veebiteenuste arhitektuuristiil

## Sisukord

1 Sissejuhatus .....	10
2 Metoodika .....	12
2.1 Projekti detailne kirjeldus .....	12
2.2 Tööriistade kirjeldus .....	13
2.3 Tööprotsessi kirjeldus .....	14
3 Töö tulemused .....	16
3.1 Arendusprojekti nõuded .....	16
3.2 Admin projekt.....	16
3.2.1 Sisselogimine Admin lehele .....	16
3.2.2 Admin liidese avaleht .....	17
3.2.3 „Customers“ vaade .....	18
3.2.4 Toiminguid võimaldavad vaated .....	18
3.3 Veebileht.....	23
3.3.1 Avaleht.....	23
3.3.2 Tantsustuudio kirjelduse vaated .....	24
3.3.3 Kursuste vaade.....	26
3.3.4 Treeningpakettide valik .....	27
3.3.5 Maksete tegemine .....	27
3.3.6 Tunniplaan .....	29
3.3.7 Ürituste vaade .....	30
3.3.8 Kontaktvorm.....	31
3.4 Arhitektuur.....	31
3.4.1 Arendusprojekt .....	31
3.4.2 Andmebaas .....	32
3.4.3 Back-end projekti arhitektuur .....	34
3.4.4 Front-endi ja Admin liidese struktuur .....	35
3.5 Disain.....	36
3.6 Kood .....	36
3.6.1 Back-end projekt.....	36

3.6.2 Front-end projekt .....	39
3.6.3 Admin liides .....	41
3.7 Testimine .....	43
3.8 Arendusprojekti juurutamine .....	44
4 Veebilehe analüüsimine ja järeldused .....	45
4.1 Tehnilise teostuse põhjendus .....	45
4.1.1 Nõuded.....	45
4.1.2 Arhitektuur.....	45
4.1.3 Disain.....	46
4.1.4 Kood .....	47
4.1.5 Testid .....	48
4.2 Tööriistade valiku põhjendus.....	48
4.3 Teostatud tööde logimine .....	50
4.4 Hinnang projekti teostusele .....	50
5 Kokkuvõte .....	52
Kasutatud kirjandus .....	53
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	54
Lisa 2 – Nädal 1 (01.03.2021-07.03.2021).....	55
Lisa 3 – Nädal 2 (08.03.2021-14.03.2021).....	56
Lisa 4 – Nädal 3 (15.03.2021-21.03.2021).....	57
Lisa 5 – Nädal 4 (22.03.2021-28.03.2021).....	58
Lisa 6 – Nädal 5 (29.03.2021-04.04.2021).....	59
Lisa 7 – Nädal 6 (05.04.2021-11.04.2021).....	60
Lisa 8 – Nädal 7 (12.04.2021-18.04.2021).....	61
Lisa 9 – Nädal 8 (19.04.2021-25.04.2021).....	62
Lisa 10 – Nädal 9 (26.04.2021-02.05.2021).....	63
Lisa 11 – Nädal 10 (03.05.2021-09.05.2021).....	64
Lisa 12 – Nädal 11 (10.05.2021-16.05.2021).....	65
Lisa 13 – Nädal 12 (17.05.2021-23.05.2021).....	66

## Jooniste loetelu

Joonis 1 - Sisselogimine Admin liidesesse.....	17
Joonis 2 - Admin liidese avaleht.....	17
Joonis 3 - Klientide vaade tulekul .....	18
Joonis 4 - Kirjelduse vaade.....	19
Joonis 5 - Treenerite vaade .....	19
Joonis 6 - Kursuste vaade .....	20
Joonis 7 - Treeningpakettide vaade .....	20
Joonis 8 - Ürituste vaade .....	21
Joonis 9 - Korduma kippuvate küsimuste vaade .....	21
Joonis 10 - Kustutamise teavitus .....	22
Joonis 11 - Eduka toimingute teavitus.....	22
Joonis 12 - Eduka kustutamise teavitus .....	23
Joonis 13 - Veebilehe avalehe vaade .....	24
Joonis 14 - Veebilehe avalehe vaade .....	24
Joonis 15 - Veebilehe kirjelduse vaade .....	25
Joonis 16 - Veebilehe treenerite vaade .....	25
Joonis 17 - Veebilehe korduma kippuvate küsimuste vaade .....	26
Joonis 18 - Veebilehe kursuste vaade.....	26
Joonis 19 - Veebilehe treeningpakettide vaade .....	27
Joonis 20 - Teenuse kirjeldus .....	28
Joonis 21 - Andmete sisestamise vorm.....	28
Joonis 22 - Panga valikud maksete tegemiseks .....	29
Joonis 23 - Veebilehe tunniplaani vaade .....	30
Joonis 24 - Veebilehe ürituste vaade .....	30
Joonis 25 - Veebilehe kontaktvormi vaade.....	31
Joonis 26 - Arendusprojekti arhitektuuri diagramm.....	32
Joonis 27 - Andmebaasi struktuur .....	33
Joonis 28 - Tabeli loomine kasutades SQL programmeerimiskeelt .....	34
Joonis 29 - REST arhitektuuri diagramm (Moody, 2019).....	35



Joonis 30 - Andmebaasi ühendus <i>appsettings.Development.json</i> failis .....	36
Joonis 31 - <i>Models</i> kaustas määratletud <i>Packages</i> klass.....	37
Joonis 32 - Lisamise ja kustutamise meetodite näited <i>InstructorsController.cs</i> failis...	38
Joonis 33 - <i>Packages</i> aadressi sisestamisel saadud tulemus.....	38
Joonis 34 - Näide <i>contact.js</i> HTML elementidest .....	39
Joonis 35 - Näide <i>contact.css</i> failist .....	40
Joonis 36 - Andmebaasi ühendus <i>fetch.js</i> failis .....	40
Joonis 37 - Andmete kätte saamine ja kuvamine HTML elementides .....	41
Joonis 38 - Funktsioonid uue informatsiooni ja väljade tühjendamise haldamiseks.....	42
Joonis 39 - <i>Update</i> funktsioon <i>actions.js</i> failis.....	42
Joonis 40 - Treenerite klassi testide näited .....	43
Joonis 41 - Juurutamise protsess .....	44

# 1 Sissejuhatus

Bachata Studio Tallinn (edaspidi BST) on tantsustuudio, mis keskendub ladina-ameerika tantsude õpetamisele. Lisaks sellele pakub stuudio show grupi treeninguid ja kehaliikumise tunde. Tulevikus on ettevõttel plaan laiendada ja tuua sisse ka teisi tantsustiile. BST tegeleb kliendi soovi korral ka erinevate ürituste ja festivalide korraldamisega. (Sirel, Kirs, Sein, & Andresen, 2020)

BST peamiseks probleemiks oli veebilehe puudumine, kus oleks võimalik klientidele edastada informatsiooni, võimaldada tundidesse registreerimist ning maksete tegemist. Samuti oli vajadus liidesele, kust on võimalik kogu veebilehte hallata. Need puudused tõid kaasa suurel hulgal tööd, mida oleks kindlasti võimalik automatiseerida ning lihtsamaks muuta. Ettevõtte juht otsustas koostada projekti ning luua tantsustuudiale veebileht ja eraldi kasutajaliides, kus eelnevalt loetud toiminguid oleks võimalik teha.

Esimeseks eesmärgiks oli valmis saada veebileht, kus on teave stuudio kohta ning registreerimise ja maksete tegemise võimalus. Esimeseks sammuks oli staatilise veebilehe loomine informatsiooni kuvamiseks. See pidi vastama ka mitte funktsionaalsetele nõuetele, milleks olid lihtsus, arusaadavus ja kasutajasõbralikkus. Teiseks valmis dünaamiline veebileht, kus kogu informatsioon tuli andmebaasist. Edaspidine samm oli erinevate funktsionaalsuste lisamine vastavalt nõuetele - kontaktvorm kirjade saatmiseks, registreerimine tundidesse, maksete tegemine läbi pangalinkide. Teine eesmärk oli veebilehe kõrvale luua *Admin* liides, kus on võimalik lisada, muuta ja kustutada veebilehel olevat informatsiooni.

Antud töö annab ülevaate loodud projektist. Töö koosneb viiest olulisest peatükist: meetodika, töö tulemused, analüüs ja järeldused, kokkuvõte ning lisad. (Sirel, Kirs, Sein, & Andresen, 2020)

Metoodika osas on kirjeldatud kasutatud tehnoloogiaid, meetodeid ning tööprotsessi. Töö tulemuste all on detailne arendusprojekti dokumentatsioon. Analüüsi ja järelduste

peatükk toob välja põhjendatud töö tulemused ning hinnang projekti protsessi kohta.  
(Sirel, Kirs, Sein, & Andresen, 2020)

## 2 Metoodika

### 2.1 Projekti detailne kirjeldus

Projekti eesmärgiks oli arendada BST tantsustuudiale kasutajasõbralik veebileht, mille kaudu edastada klientidele informatsiooni studios toimuva kohta ning võimaldada neile tundidesse registreerimist ning maksete tegemist. Nende funktsionaalsustega oli võimalik teha klientide elu lihtsamaks ning vältida suurel hulgal manuaalset tööd ettevõttes. Teiseks suureks sammuks oli studio mäenedžerile kasutajaliidese loomine, kust oleks võimalik hallata kogu veebilehte.

Projekt valmis täiesti algusest autori käe läbi. Esimeses etapis loodi staatiline veebileht (*front-end*), mis kuvas informatsiooni tantsustuudio kohta. Teises etapis valmis API (*Application Programming Interface*), teise sõnaga *back-end*, kus hoiti andmetega seotud loogikat. See ühendati Azure SQL andmebaasiga, kuhu olid sisse kirjutatud kõik vajalikud andmed, mis veebilehel oleks vaja kuvada. Kolmandas etapis oli eesmärgiks omavahel ühendada *front-end* ja *back-end*. Valmis sai dünaamiline veebileht, kus *front-end* päris andmeid *back-end*'i kaudu, mis oli ühenduses andmebaasiga. Neljandaks etapiks oli *Admin* liidese loomine, mille jaoks tehti eraldi projekt. Algselt valmis samuti staatiline liides, kus ei olnud võimalik mingeid toiminguid teha. Hiljem ühendati see *back-end*'ga, et võimaldada *Admin* liideses lisada, muuta ja kustutada veebilehel olevat informatsiooni. See tähendas seda, et *Admin* liidese kaudu sai lisada, muuta ja kustutada andmebaasis olevat informatsiooni. Pärast tehtud toiminguid muutus informatsioon ka veebilehel olemasoleva andmebaasi ühenduse tõttu. Viimases etapis keskenduti veebilehe funktsionaalsuste lisamisele. Peamisteks olid tundidesse registreerimine ning maksete tegemine. Lisaks sai tehtud kontaktvorm, mille kaudu on klientidel võimalik saata kirju otse ettevõtte meilile. (Sirel, Kirs, Sein, & Andresen, 2020)

Lõpuks sai valmis veebileht, mis on avalikult klientidele kättesaadav ning kus neil on võimalik registreerida ja sooritada makseid endale sobiva pangalingi kaudu. Samuti on neile nähtav kogu informatsioon studios toimuva kohta. Nähtavalt vähenes ka töötajate töömaht ning enamus toimingutest on automatiseeritud.

## 2.2 Tööriistade kirjeldus

BST veebilehe arendamiseks kasutati mitmeid tarkvaraarenduse platvorme ja erinevaid programmeerimiskeeli. Tööriistade valik oli juba alguses selge ning kasutusele võeti need, millega oli kogemus eelmistest aastatest olemas. (Sirel, Kirs, Sein, & Andresen, 2020)

*Front-end* ja *Admin* projekti arendamiseks kasutati Microsoft Visual Studio Code'i ning ReactJS raamistikku. Programmeerimiskeeled olid JavaScript, HTML ja CSS. Kõik komponendid ja vajalikud navigeerimised veebilehel ja *Admin* projektis kirjutati JavaScripti ja HTMLi kasutades. Veebilehe ja *Admin* projekti disain loodi CSS failides. Vajalikud koodiread andmebaasiga ühenduse saamiseks kirjutati JavaScript failides, et andmebaasist andmeid kätte saada ning veebilehel kuvada. (Sirel, Kirs, Sein, & Andresen, 2020)

*Back-end* projekti arendamiseks kasutati Microsoft Visual Studiot ning raamistikuks valiti ASP.NET Core. Programmeerimiskeeleks oli C# (C-Sharp). See projekt sisaldas kogu andmetega seotud loogikat ning andmebaasi ühendust. Andmebaasi valikul otsustati Microsoft Azure SQL andmebaasi kasuks, millele on ligipääs Azure Portalis. Samuti võeti andmebaasi haldamiseks kasutusele Azure Data Studio, kus oli võimalik näha andmebaasis olevaid tabeleid ning lisada, muuta ja kustutada andmeid. (Sirel, Kirs, Sein, & Andresen, 2020)

Klientidele maksete võimaldamiseks integreeriti projekt Mairu teenuste arveldamise platvormiga. Kontole loodi vajalikud teenused treeningpakettide jaoks, kus tuli määrata nimi, hind ja kirjeldus. Igale teenusele genereeriti oma kood, mille tuli lisada Mairu veebiaadressi lõppu (mairu.ee/xxxxx). Aadresside jaoks tehti andmebaasi vajalik tabeli väli ning lisati need sinna.

Meilide saatmiseks loodud kontakvorm integreeriti EmailJS teenusega JavaScript failis. Vajalik oli luua EmailJS konto ning ühendada see tantsustudio meiliaadressiga. Seal sai vajaliku teenuse ID, mida oli vaja koodis kasutada, et tekitada ühendus. Lisaks sellele oli võimalik disainida mall, kuidas saadetud kiri välja näeb.

Azure DevOps keskkonda kasutati koodi hoidmiseks, kuid see laeti hiljem üles GitHubi, et oleks kergem koos Azure Portaliga juurutamist läbi viia. Seal sai vaadata kõiki

*commit*'e ja *milestone*'e. Iga *commit*'le prooviti panna võimalikult täpne nimi, et oleks arusaadav, mis antud hetkel koodis tehti. *Milestone*'id aitasid paika panna töomahu mingiks kindlaks perioodiks. Selle projekti jooksul neid nii palju ei kasutatud, vaid loodi eraldi dokument Google Docs'i, kuhu pandi kirja iganädalaselt valmis saanud ja pooleli olevad tööülesanded.

Töötunnid logiti TogglTracki. See on hea platvorm, kus näha töömahtu ning lisada kirjeldused, mida täpselt tehti. Nii saab tervel tööprotsessil silma peal hoida ning vaadata kui palju aega mingi kindla ülesande peale kulub. (Sirel, Kirs, Sein, & Andresen, 2020)

Esmakordselt prooviti kogu projekt teha kasutajatele avalikuks. Selleks kasutati Azure Portalit ning GitHubi. Azure Portalis loodi *front-end*, *back-end* ja *admin* projektide jaoks veel kolm eraldi projekti. *Front-end*'i ja *Admin*'i jaoks Azure Static Web App ja *back-end*'i jaoks Azure App Service. Kõik need olid omavahel ühenduses, samuti GitHubiga, mis tähendas seda, et kood oli Azure jaoks kättesaadav. GitHubi kaudu on seda väga lihtne teha, sest see teeb automaatsed töövood pärast igat uut *commit*'i. Eduka katse puhul oli võimalus veebilehte näha Azure Portalis, kus kuvati juurutatud veebilehe aadress. Nüüdseks on veebileht avalik ning klientidel on võimalik seda kasutada.

## 2.3 Tööprotsessi kirjeldus

BST veebilehe arendamise idee sündis väga lihtsalt ja kiiresti. Varasemalt ei olnud tantsustuudiol korralikku veebilehte olnud, kuid vajadus oli suur. Esimene veebilehe loomise katse toimus eelmise semestri „Meeskonnaprojekt“ aine raames. Arendamisel löid kaasa neli tudengit. Valmis sai tagasihoidliku disainiga dünaamiline veebileht, kus klientidel oli võimalik sisse logida, tutvuda treeningpakettide ja tunniplaaniga ning küsimuste korral võtta stuudioga ühendust läbi kontaktvormi. Lisaks sellele valmis väga algeline *Admin* projekt, mille kaudu oli võimalik veebilehel olevat infot muuta ja kustutada ning vajadusel uut lisada.

Meeskonnaprojekti raames ei saanud veebileht täielikult valmis. See vajab paari uut funktsionaalsust, milleks oli registreerimine tantsutundidesse ning nende eest maksmine. Antud töö autor osales samuti meeskonnaprojektis ning otsustas veebilehte lõputöökss edasi arendada. Võeti uuesti ühendust ettevõtte juhiga ning lepiti kokku kohtumine, kus projektist pikemalt rääkida. Muudatusi tehti ka nõuetes - millist informatsiooni veebilehel

kuvada ja millist mitte, registreerimise ja maksmise funktsionaalsus, veebilehe disain. Kui kõik vajalik informatsioon käes, sai veebilehe arendamisega alustada.

Töö käik oli algusest peale selge. Esmalt valmis eelmisest veebilehest erineva disainiga staatiline veebileht. Teisena loodi ühendus andmebaasiga ning siis juba lisati vajalikud funktsionaalsused. Kolmandaks valmis *Admin* projekt veebilehe haldamiseks. Kõige algsem veebileht tehti ka kohe GitHubi ja Azure Portali abil avalikuks. Nii sai järkjärgult väiksemaid *commit*'e juurtutatud ja ei tekkinud probleeme.

Pidev tööprotsessi jälgimine on oluline ning selleks loodi dokument, kuhu kanti nädalate kaupa tehtud tööülesanded. Kui midagi jäi tegemata või oli poolik, siis märgiti need teiste värvidega, et hiljem nende juurde tagasi tulla. Takistuste ja probleemide korral võttis töö autor rohkem aega, et probleemi lähemalt uurida, ise lahenduseni jõuda ning selle käigus ka õppida ja areneda. Aeg-ajalt sai räägitud ka juhendajatega, kes suunasid ja vajaduse korral aitasid.

Projektiga tegeleti peaaegu igapäevaselt. Tuli ette periood, kus töö tegemine seiskus ning projekti arendamine toimus aeglasemalt, kui planeeritud.

## **3 Töö tulemused**

### **3.1 Arendusprojekti nõuded**

BST veebilehe arendamine toimus etteantud nõuete järgi. Esimeseks suurimaks nõudeks oli dünaamilise veebilehe loomine, et edastada klientidele võimalikult palju informatsiooni veebilehe kaudu. Teiseks funktsionaalseks nõudeks oli maksete tegemise võimaldamine erinevate pangalinkide kaudu läbi veebilehe. Samuti oli vajalik *Admin* liides stuudio mäenedžerile, et hallata kogu veebilehel olevat infot. Mittefunktsionaalseteks nõueteks olid kasutajasõbralikkus, disaini lihtsus ja kerge navigeerimine veebilehel.

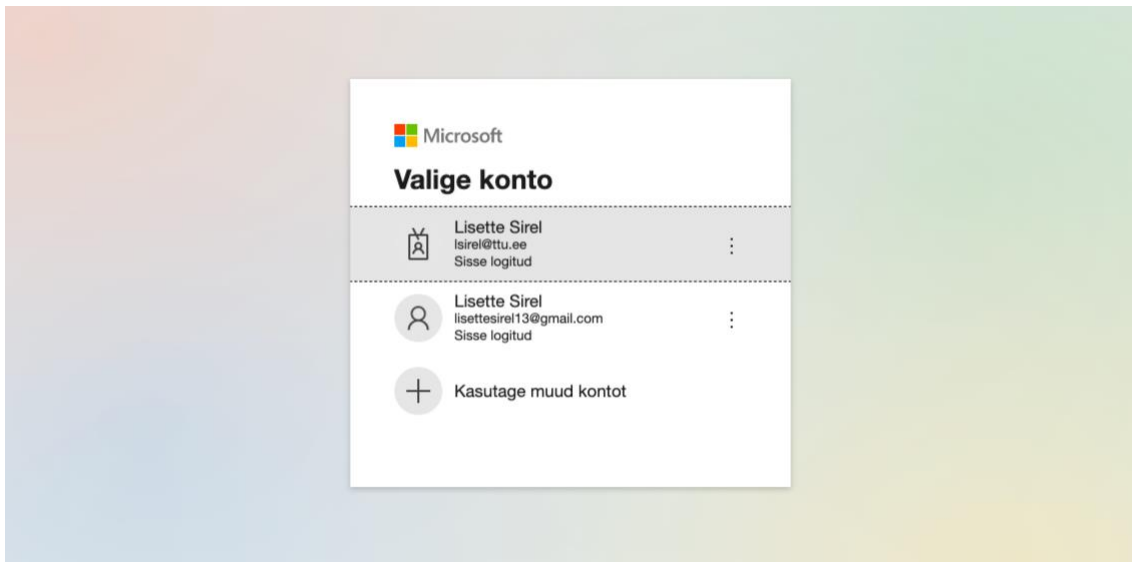
### **3.2 Admin projekt**

BST veebilehe haldamiseks loodi *Admin* projekt, kus tantsustuudio mäenedžeril on võimalik informatsiooni muuta, kustutada ja vajadusel lisada. Toimingute tegemiseks on vajalik sisselogimine. Pärast sisselogimist kuvatakse kasutajale avaleht, kus on võimalik menüüst valida toimingute jaoks sobiv kategooria. (Sirel, Kirs, Sein, & Andresen, 2020)

#### **3.2.1 Sisselogimine Admin lehele**

*Admin* lehele minnes avaneb vaade sisselogimise vormile (Joonis 1), kuhu tuleb sisestada kasutajanimi ja parool, et saada ligipääsu kogu projekti sisule. Hetkel saab sisse logida Microsoft kontoga, kuid seda on võimalik muuta. Ligipääs on ainult teatud kasutajatel, kellel on õigus toimingute tegemiseks, mis määrati Azure Portalis. (Sirel, Kirs, Sein, & Andresen, 2020)





Joonis 1 - Sisselogimine Admin liidesesse

### 3.2.2 Admin liidese avaleht

Pärast edukat sisselogimist ilmub ekraanile avalehe vaade (Joonis 2). Vasakul ääres asub menüü, millel on erinevad kategooriad („Customers“, „About“, „Team“, „Courses“, „Packages“, „Events“ ja „FAQ“).



Joonis 2 - Admin liidese avaleht

### 3.2.3 „Customers“ vaade

Valides menüüst „Customers“ kuvatakse teade, et leht on peagi tulemas (Joonis 3). Seda lehte ei saanud veel lõpuni teha, sest hetkel Mairu makseteenusel puudub võimalus klientide listi tagastada. Neid on võimalik näha ainult Mairu konto alt.



Joonis 3 - Klientide vaade tulekul

### 3.2.4 Toiminguid võimaldavad vaated

Valides menüüst „About“ (Joonis 4), „Team“ (Joonis 5), „Courses“ (Joonis 6), „Packages“ (Joonis 7), „Events“ (Joonis 8) või „FAQ“ (Joonis 9), kuvatakse ekraanil sarnaseid vaateid. Kõikidel on olemas sisestusväljad, tabelid andmebaasist tulevate andmetega ning „Submit“ ja „Reset“ nupud. Informatsiooni lisamiseks on vajalik sisestusväljad täita ning vajutada nuppu „Submit“, mis salvestab uue informatsiooni ühenduses oleva andmebaasi vastavasse tabelisse. „Reset“ nupule vajutades tühjendatakse kõik sisestusväljad.

**BST**

Customers

About

Team

Courses

Packages

Events

FAQ

## About

SUBMIT
RESET

Description	Image
<p>Bachata Studio Tallinn was established in 2020 and obtained a great reputation from the very beginning. We are a dance school based in Estonia focusing on Latin American dance styles. We host regular weekly classes in Tallinn and Haapsalu. Bachata Studio Tallinn offers classes for beginners, improvers, intermediate and advanced students. All of our dance courses are run by highly trained and friendly instructors. In addition to regular dance courses, we organise various special events and social dance nights. We offer private classes which are specifically tailored for every individual. Bachata Studio Tallinn strives to create a sociable atmosphere by celebrating great music, dance and by getting to know each other. Whether people would like to just try something new or wish to carve out a dance career - Bachata Studio Tallinn welcomes everyone!</p>	<div style="text-align: right; margin-bottom: 5px;"> <span style="color: #0070c0; font-size: 1.2em;">✎</span> <span style="color: red; font-size: 1.2em;">🗑</span> </div> <p style="font-size: 0.9em;">/images/homepic.jpg</p>

Joonis 4 - Kirjelduse vaade

**BST**

Customers

About

Team

Courses

Packages

Events

FAQ

## Team

SUBMIT
RESET

First Name	Last Name	Description	Image
Anastassija	Gluhankova		<div style="text-align: right; margin-bottom: 5px;"> <span style="color: #0070c0; font-size: 1.2em;">✎</span> <span style="color: red; font-size: 1.2em;">🗑</span> </div> <p style="font-size: 0.9em;">/images/nastja.jpg</p>
Liisi	Sagor	<p>Liisi has been dancing almost the whole of her life, but Bachata is something that has stayed in her heart. She is true fan of the style - she always gives her 100% in teaching and training. On the dance floor, she has a connection with a complete stranger. Bachata is like a language on its own - no matter where you go, you will understand other dancers. This how she started travelling and going to festivals - you can have something really unique with an unknown person.</p>	<div style="text-align: right; margin-bottom: 5px;"> <span style="color: #0070c0; font-size: 1.2em;">✎</span> <span style="color: red; font-size: 1.2em;">🗑</span> </div> <p style="font-size: 0.9em;">/images/liisi.jpg</p>
		<p>Lisette started her journey quite randomly in 2018. She had a bet with her friend and you can guess three times, who lost? Lisette started teaching at the beginning of 2020 and always gives her full energy, so the classes would be memorable for all of the dancers. Bachata feels like</p>	<div style="text-align: right; margin-bottom: 5px;"> <span style="color: #0070c0; font-size: 1.2em;">✎</span> <span style="color: red; font-size: 1.2em;">🗑</span> </div>

Joonis 5 - Treenerite vaade

**BST**

Customers

About

Team

**Courses**

Packages

Events

FAQ

### Courses

SUBMIT
RESET

Name	Description	Image	
Ladies Shines	A woman shines when she is able to express herself through dance. Bachata Ladies Shines is a mixture of Latin flavour and elegance, which brings out every dancer's personality, attitude and independence. During the classes, our instructors focus on several details: head, hands, use of arms, timing, spins, hip movements. Get creative, expressive and spontaneous!	/images/bachatalady.jpeg	
Bachata	Bachata is a beautiful and sensual dance. It is a great way to develop your body movement, learn musicality and free yourself in the midst of bachata music. There are different bachata dance styles, which are taught in our classes: Dominican Bachata, Sensual Bachata, Modern/Urban Bachata. We have courses for different levels: beginners (A), improvers (B), intermediate (C) and advanced (D).	/images/nastjarado1.jpeg	

Joonis 6 - Kursuste vaade

**BST**

Customers

About

Team

Courses

**Packages**

Events

FAQ

### Packages

SUBMIT
RESET

Name	Description	Price	Payment Link	
One course (8h)	Bachata Couples, 2 hours a week	49		
Two courses (16h)	Bachata Couples, 4 hours a week	69		
Shine Ladies Open	Bachata Lady Styling, 1 hour a week	33		
Shine Ladies & one course	Lady Styling, Bachata Couples	65		

https://purple-tree-0fa89f503.azurestaticapps.net/packages

Joonis 7 - Treeningpakettide vaade

**BST**

Customers

About

Team

Courses

Packages

**Events**

FAQ

### Events

SUBMIT
RESET

Name	Description	Image	
Studio Opening Party	🔥 Coming soon 🔥	/images/party.jpg	✎ 🗑
Dance Weekend	🔥 Coming soon 🔥	/images/danceweekend.jpg	✎ 🗑

Joonis 8 - Ürituste vaade

**BST**

Customers

About

Team

Courses

Packages

Events

**FAQ**

### Faq

SUBMIT
RESET

Name	Description	
Where are you located?	In Tallinn, we hold our dance classes at Veerenni 29-1. // In Haapsalu, we organise classes at the Haapsalu Culture Centre, at Posti 3.	✎ 🗑
Is parking available?	Yes, you can park in front of the building.	✎ 🗑
What should I wear in the classes?	Wear comfortable clothes, so you could feel good and relaxed. In that way, you can focus more on dancing.	✎ 🗑
What kind of shoes should I wear?	Please bring along clean and comfortable inner shoes. However, running shoes and trainers do not really slip whilst dancing, so we would not recommend wearing these.	✎ 🗑
Do I have to come with a partner?	No need to bring a partner. Bachata is a social dance. At the beginning of the class, our instructors will find you a partner or we will be exchanging partners during the class. This way you have an opportunity to meet new people. However, if you do have a partner, feel free to bring	✎ 🗑

Joonis 9 - Korduma kippuvate küsimuste vaade

Informatsiooni muutmiseks tuleb vajutada sinisele pliiatsile. See täidab automaatselt sisestusväljad nende andmetega, mis parasjagu listist valiti. Muudatuste sisseviimiseks andmebaasi tuleb samuti vajutada „Submit“ nuppu. Informatsiooni kustutamiseks tuleb valida punane prügikasti märk, mille peale vajutades kuvatakse ekraanile teavitus ning küsitakse veelkord üle, kas kasutaja on selles toimingus kindel (Joonis 10).

**BST**

- Dashboard
- Customers
- About
- Team
- Courses
- Packages
- Events
- FAQ

### Faq

purple-tree-0fa89f603.azurestaticapps.net says  
Are you sure you want to delete this record?

Cancel OK

Question

Answer

SUBMIT RESET

Name	Description		
Where are you located?	In Tallinn, we hold our dance classes at Veerenni 29-1. // In Haapsalu, we organise classes at the Haapsalu Culture Centre, at Posti 3.		
Is parking available?	Yes, you can park in front of the building.		
What should I wear in the classes?	Wear comfortable clothes, so you could feel good and relaxed. In that way, you can focus more on dancing.		
What kind of shoes should I wear?	Please bring along clean and comfortable inner shoes. However, running shoes and trainers do not really slip whilst dancing, so we would not recommend wearing these.		
Do I have to come with a partner?	No need to bring a partner. Bachata is a social dance. At the beginning of the class, our instructors will find you a partner or we will be exchanging partners during the class. This way you have an opportunity to meet new people. However, if you do have a partner, feel free to bring		

Joonis 10 - Kustutamise teavitus

Peale igat „Submit“ nupule vajutamist kuvatakse paremal üleval nurgas teavitus, kas toiming oli edukas või mitte (Joonis 11). Teavitus kuvatakse ka kustutamise puhul (Joonis 12).

**BST**

- Dashboard
- Customers
- About
- Team
- Courses
- Packages
- Events
- FAQ

### About

Submitted successfully

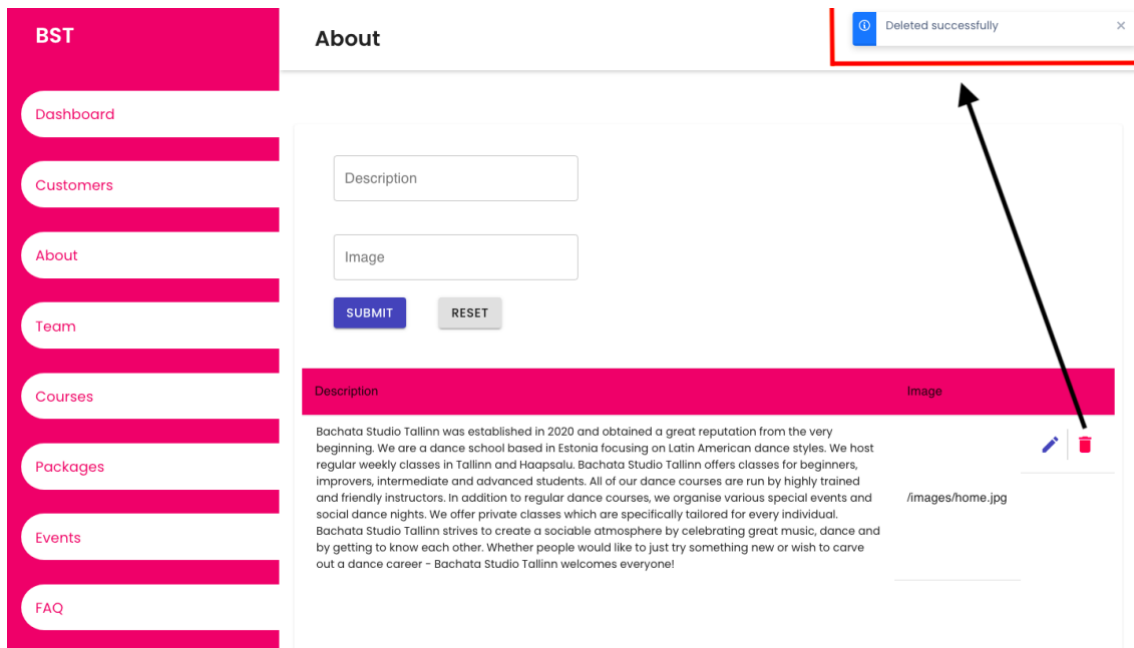
Description

Image

SUBMIT RESET

Description	Image		
Bachata Studio Tallinn was established in 2020 and obtained a great reputation from the very beginning. We are a dance school based in Estonia focusing on Latin American dance styles. We host regular weekly classes in Tallinn and Haapsalu. Bachata Studio Tallinn offers classes for beginners, improvers, intermediate and advanced students. All of our dance courses are run by highly trained and friendly instructors. In addition to regular dance courses, we organise various special events and social dance nights. We offer private classes which are specifically tailored for every individual. Bachata Studio Tallinn strives to create a sociable atmosphere by celebrating great music, dance and by getting to know each other. Whether people would like to just try something new or wish to carve out a dance career - Bachata Studio Tallinn welcomes everyone!	/images/home.jpg		
Welcome to Bachata Studio Tallinn!	/images/team.jpg		

Joonis 11 - Eduka toimingu teavitus



Joonis 12 - Eduka kustutamise teavitus

### 3.3 Veebileht

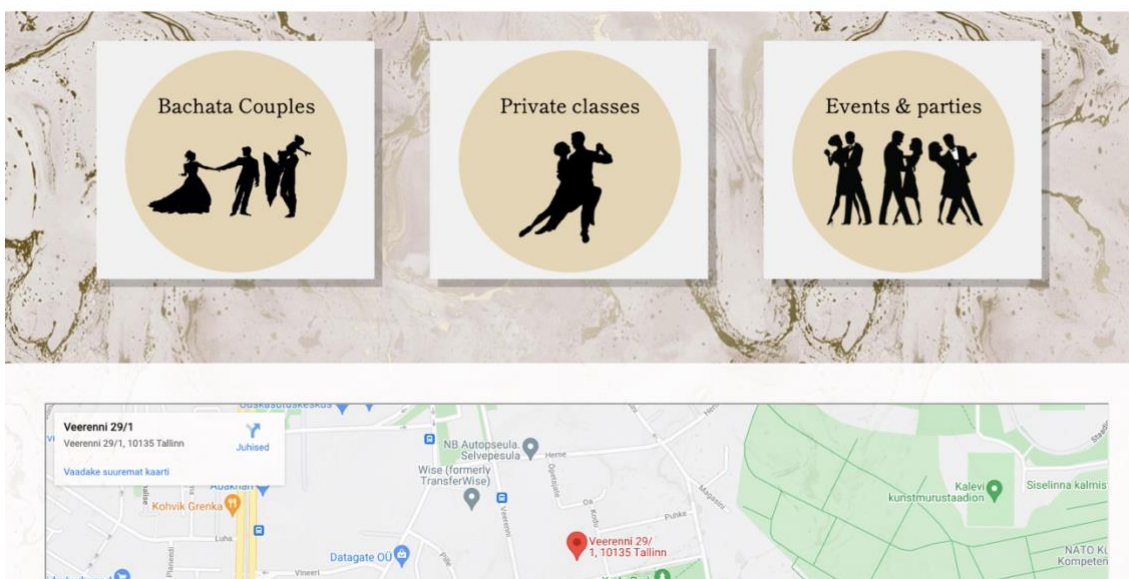
BST dünaamiline veebileht valmis vastavalt kokkulepitud nõuetele. Disaini osas järgiti kindlalt värvikombinatsiooni ning üldiselt lähtuti ka reaalsete stuudio ruumide disainist. Menüüsse tekkis juurde erinevaid valikuid, et klientidele rohkem informatsiooni edastada.

#### 3.3.1 Avaleht

Avaleht on lihtne, mis kuvab treenerite pilti, väikest mõtet tantsimise kohta, pakutavaid teenuseid ning kaarti, kus on stuudio asukoht (Joonis 13, 14).



Joonis 13 - Veebilehe avalehe vaade



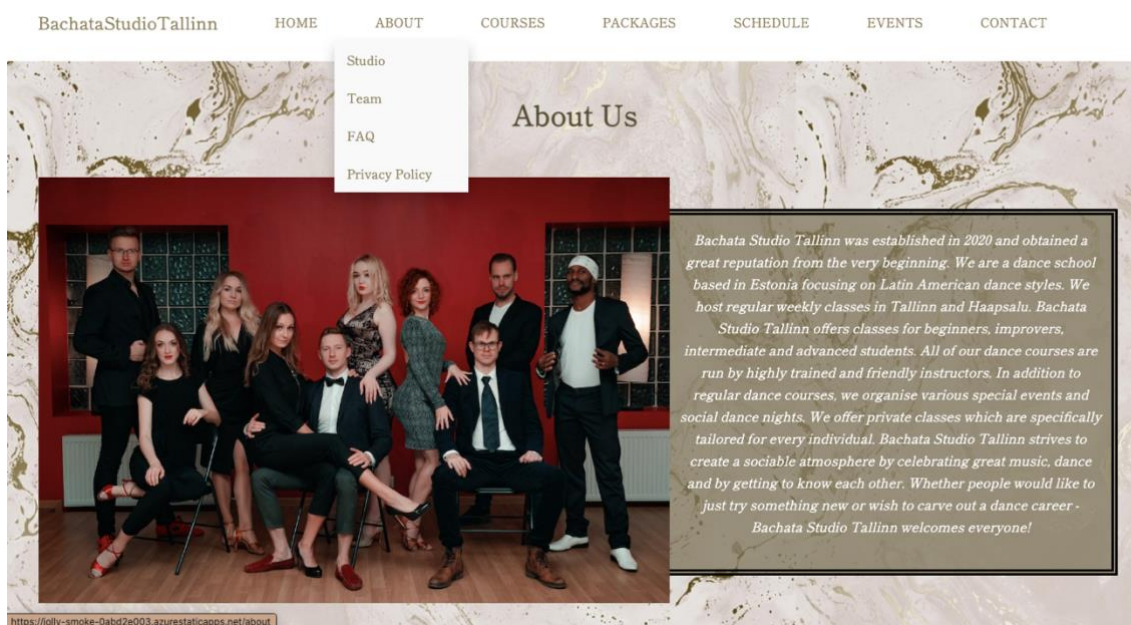
Joonis 14 - Veebilehe avalehe vaade

### 3.3.2 Tantsustuudio kirjelduse vaated

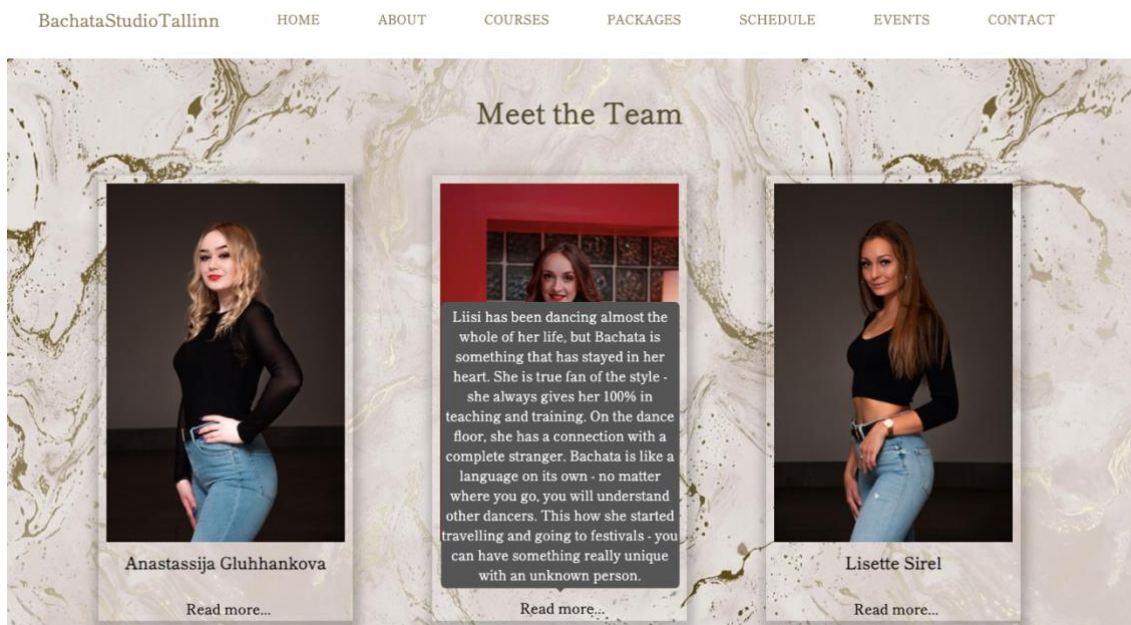
Valides ülevalt menüüst „About“, avaneb rippmenüü, kus on omakorda võimalik valida „Studio“ „Team“, „FAQ“ või „Privacy policy“. „Studio“ (Joonis 15) all on kirjeldus tantsustuudio kohta, „Team“ (Joonis 16) all on terve stuudio tiimi tutvustus, „FAQ“



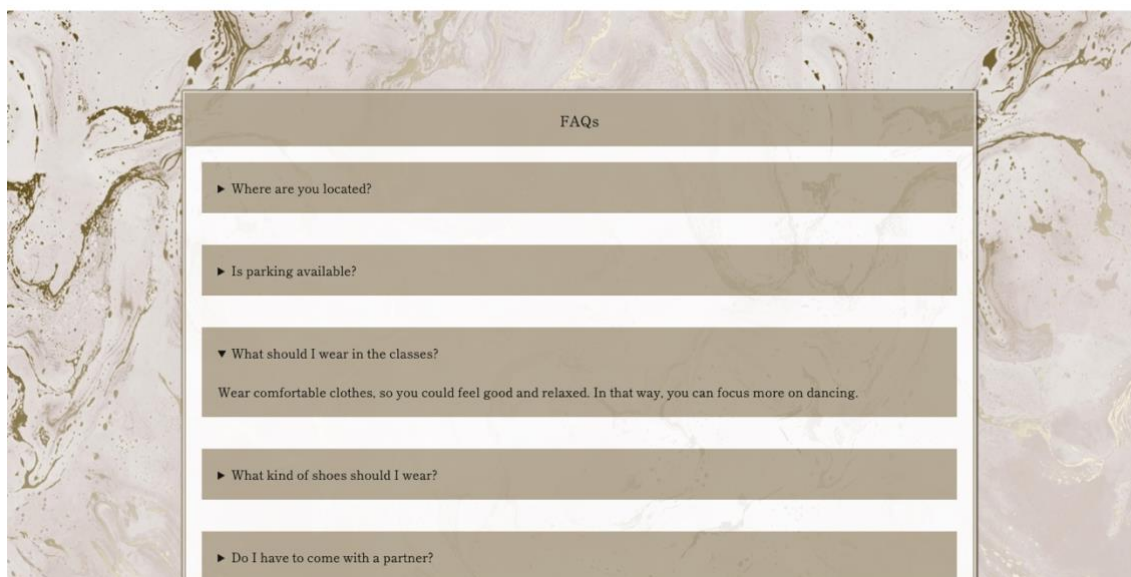
(Joonis 17) all on vastused korduma kippuvatele küsimustele ning „Privacy policy“ on alles tulemas.



Joonis 15 - Veebilehe kirjelduse vaade



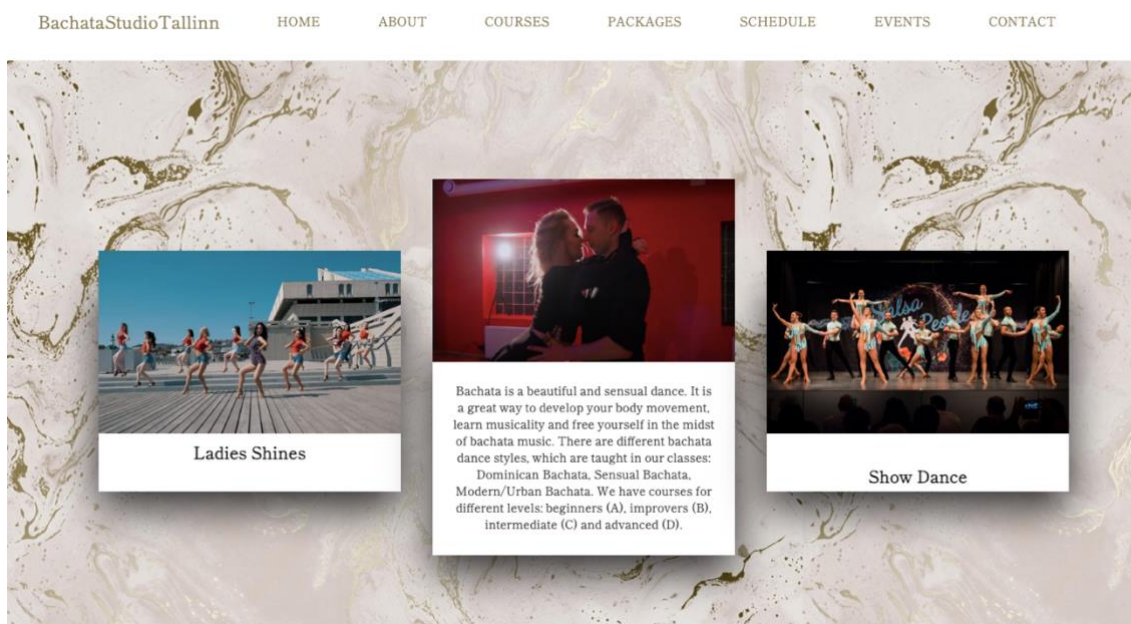
Joonis 16 - Veebilehe treenerite vaade



Joonis 17 - Veebilehe korduma kippuvate küsimuste vaade

### 3.3.3 Kursuste vaade

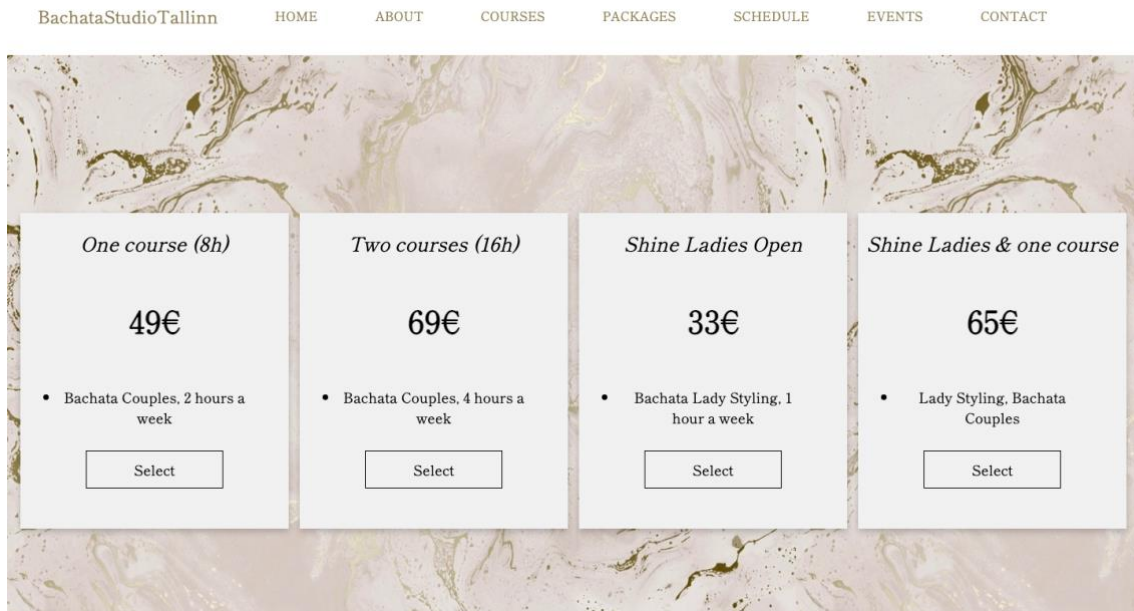
Valides ülevalt menüüst „Courses“, avaneb vaade tantsustuudio kursustele (Joonis 18). Iga kursuse kohta saab lugeda lisainfot viies kursori meelepärasele kaardile.



Joonis 18 - Veebilehe kursuste vaade

### 3.3.4 Treeningpakettide valik

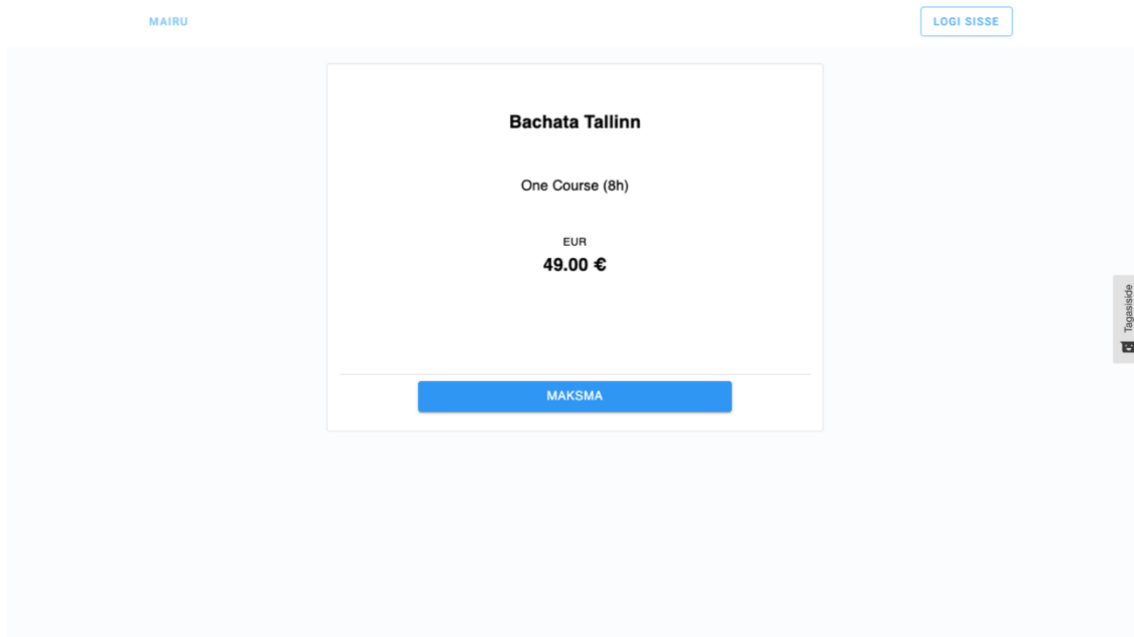
Valides ülevalt menüüst „Packages“, avaneb vaade erinevatele treeningpakettidele (Joonis 19). Vajutades „Select“ nuppu, avaneb makselink, mille kaudu on klientidel võimalus maksta valitud paketi eest.



Joonis 19 - Veebilehe treeningpakettide vaade

### 3.3.5 Maksete tegemine

Maksete tegemiseks on vajalik minna pakettide lehele ning vajutada nuppu „Select“. Uues aknas avaneb valitud teenuse kirjeldus ning maksmise nupp (Joonis 20).



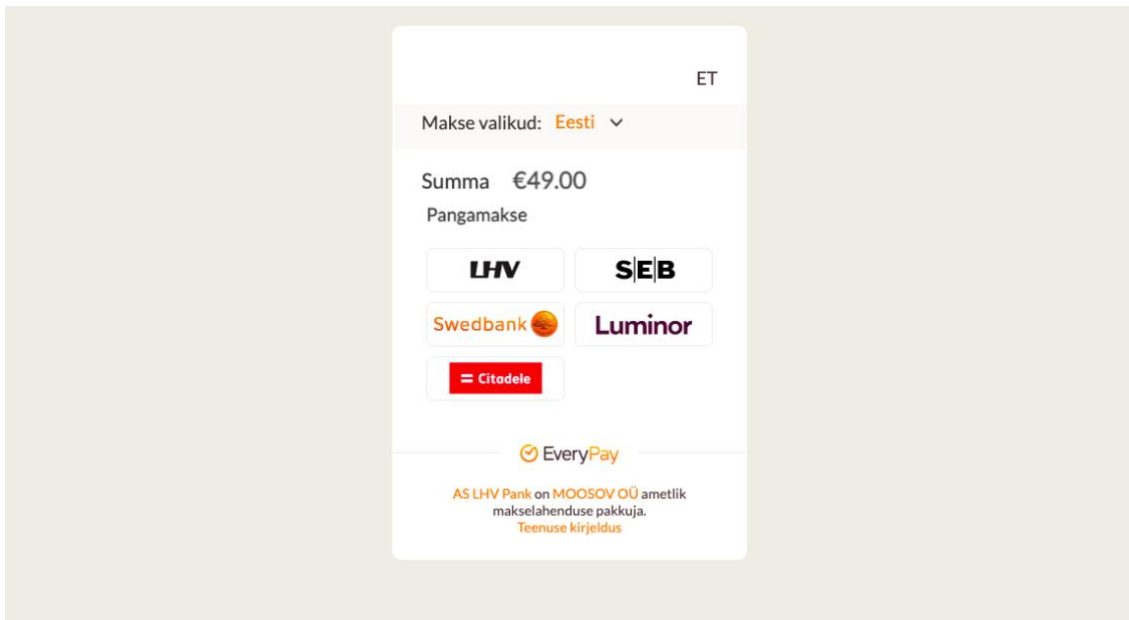
Joonis 20 - Teenuse kirjeldus

Vajutades nupule „Maksma“ avaneb vorm, kuhu klient peab kirjutama enda meiliaadressi ja nime, mis on vajalik registreerimiseks (Joonis 21).

The image shows a registration form titled 'Lisa maksja andmed'. At the top left, the word 'MAIRU' is displayed. At the top right, there is a 'LOGI SISSE' button. The form is contained within a white box with a thin border. It has three input fields: 'Email Address', 'First Name', and 'Last Name'. Below the 'Email Address' field, there is a small note: 'Siia aadressile saadame sulle arve'. At the bottom of the white box is a blue button labeled 'MAKSMA'. On the right side of the page, there is a vertical sidebar with a 'Teguside' button.

Joonis 21 - Andmete sisestamise vorm

Pärast andmete sisestamist ja nupule vajutamist avaneb uus vaade, kus on võimalik valida endale sobiv pank, mille kaudu makset teha (Joonis 22). Pärast panga valikut avaneb jälle uus vaade, kus klient saab panka sisse logida ning sooritada makse.



Joonis 22 - Panga valikud maksete tegemiseks

### 3.3.6 Tunniplaan

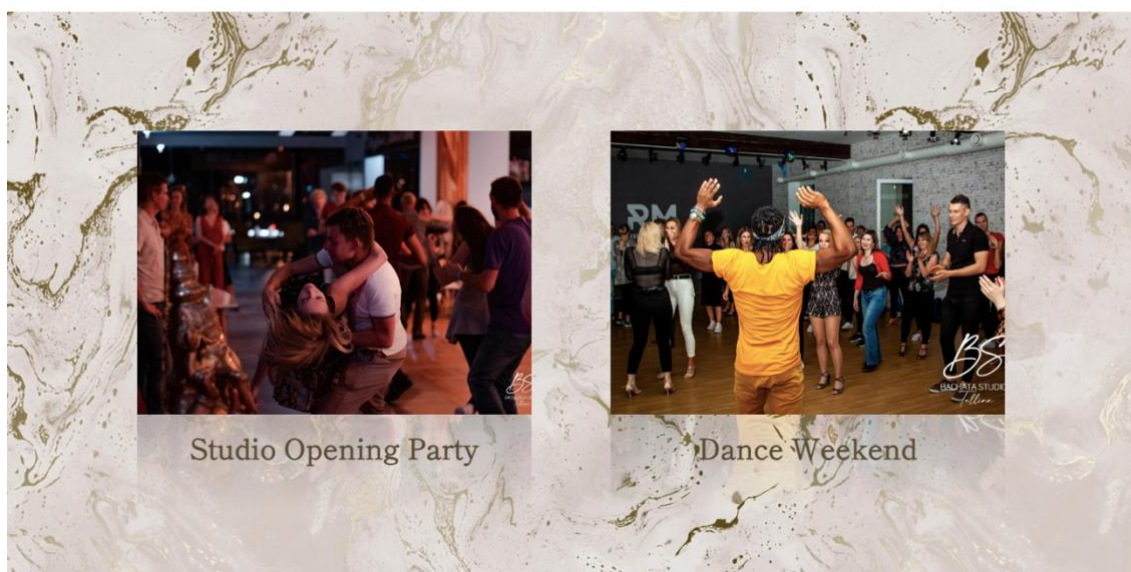
Valides ülevalt menüüst „Schedule“, avaneb vaade tunniplaanile, kuhu sisestatakse kõik toimuvad treeningud (Joonis 23).

JUNE	31 Monday	1 Tuesday	2 Wednesday	3 Thursday	4 Friday	5 Saturday	6 Sunday
18:00-19:00			Improvers B1 (2h)	Intermediate C1 (2h)			
19:00-20:00		Advanced D1 (2h)				Workshop Open (3h)	Show Dance (2h)
20:00-21:00	Beginners A1 (2h)		Ladies Shines (1h 15min)				
21:00-22:00		Instructors Training			Bachata Party		
22:00-23:00							

Joonis 23 - Veebilehe tunniplani vaade

### 3.3.7 Ürituste vaade

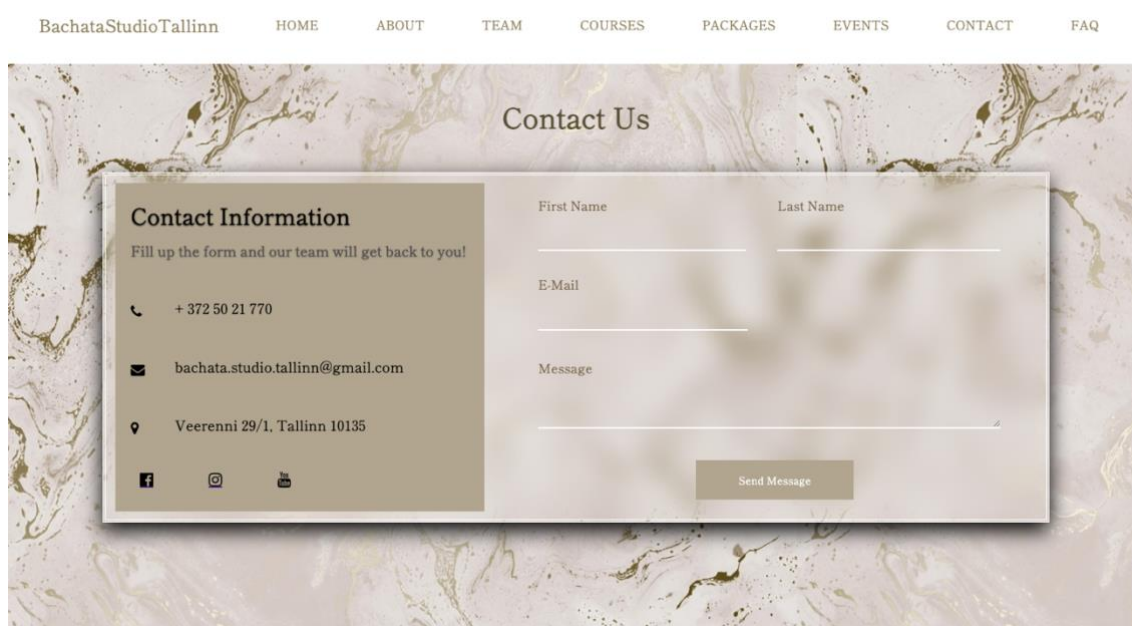
Valides ülevalt menüüst „Events“, avaneb vaade üritustele (Joonis 24). Ürituse juurde on lisatud pilt ja kirjeldus. Kirjeldust näeb kaardi tagaküljelt, milleks on vaja liikuda kursoriga ürituse kaardi peale.



Joonis 24 - Veebilehe ürituste vaade

### 3.3.8 Kontaktvorm

Valides ülevalt menüüst „Contact“, avaneb vaade kontaktvormile (Joonis 25), kus on kirjas tantsustuudio asukoht, telefoninumber ja meiliaadress. Lisaks sellele on võimalus külastada stuudio sotsiaalmeediakontosid klikkides sotsiaalmeedia ikoonide peale. Seal oleva kontaktvormi kaudu on klientidel hea võimalus tekkivate küsimuste korral stuudiole kirjutada. Pärast täidetud vormi saatmist jõuab kiri otse stuudio meilile.



Joonis 25 - Veebilehe kontaktvormi vaade

## 3.4 Arhitektuur

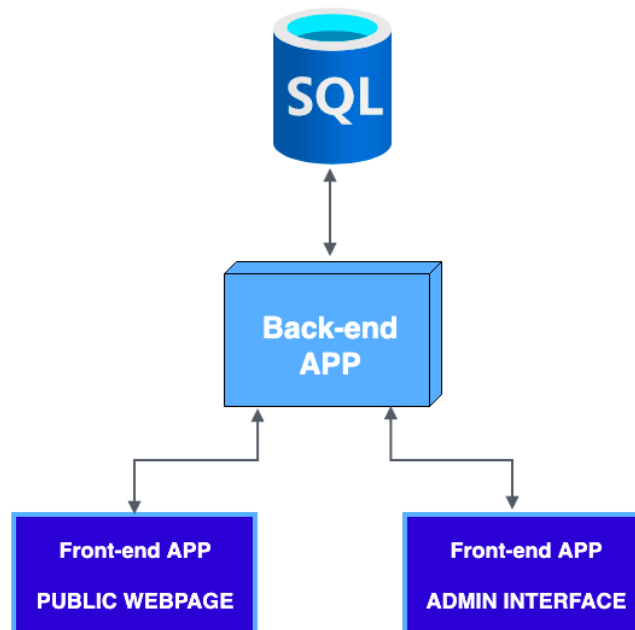
### 3.4.1 Arendusprojekt

BST arendusprojekti arhitektuur koosneb 4 suuremast komponendist - Azure SQL andmebaas, *back-end* projekt (veebi API) ja 2 *front-end* projekti, milleks on veebileht (Public Webpage) ja *Admin* liides (Admin Interface) (Joonis 26).

Arhitektuur on ülesehitatud võimalikult lihtsalt. *Back-end* on ühenduses nii SQL andmebaasiga kui ka mõlema *front-end* projektiga. *Front-end* projektide ja andmebaasi vahel otsene ühendus puudub. Arendusprojektiis kasutatakse REST arhitektuuri stiili

CRUD (*create, read, update and delete*) funktsioonidega, mis tähistab loomise, lugemise, uuendamise ja kustutamise toiminguid.

*Admin* liides teeb kõiki 4 toimingut. Andmete kuvamiseks liideses peab tegema lugemise (*read*) toimingut, et andmebaasist andmeid saada. Uute andmete lisamiseks andmebaasi *Admin* liidese kaudu tuleb teha lisamise (*create*) toiming. Andmete uuendamiseks ja kustutamiseks tehakse CRUD *update* ja *delete* toiminguid. *Front-end* projekt teeb ainult lugemise toimingut, et kuvada veebilehel andmebaasis olevat infot.



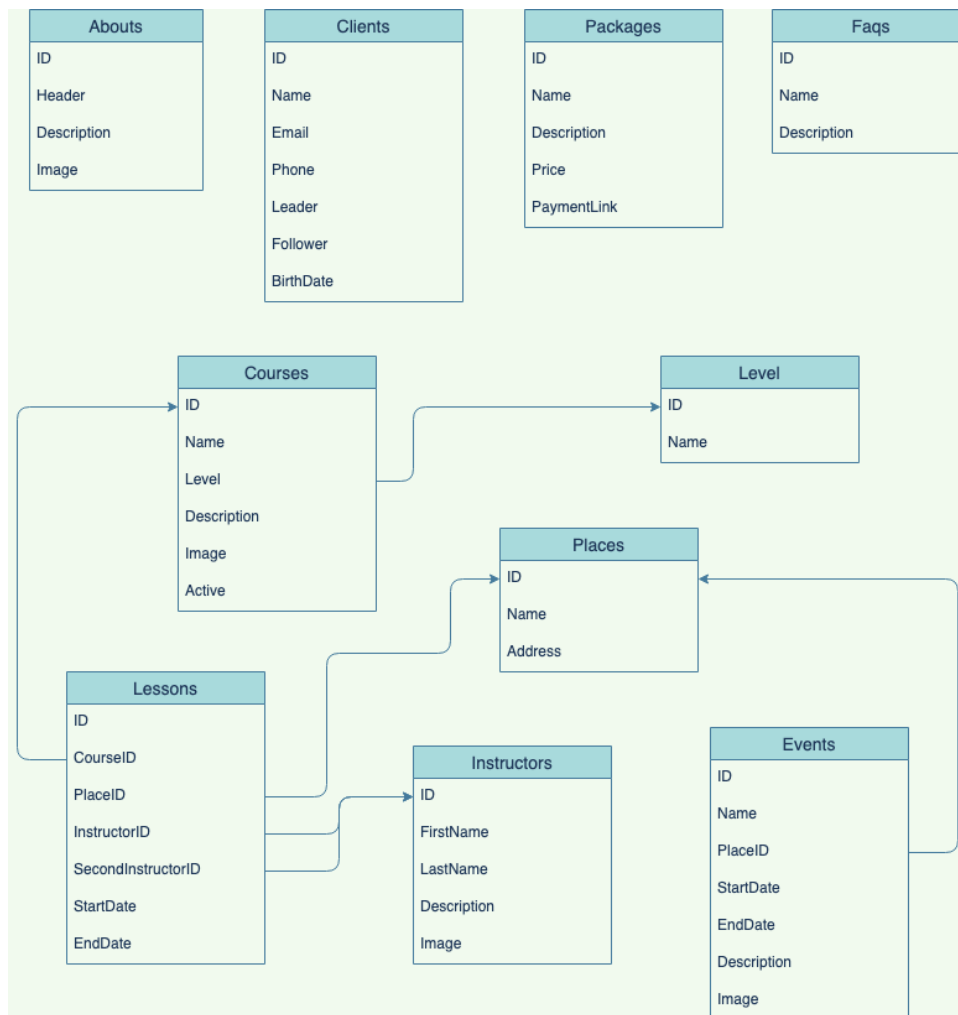
Joonis 26 - Arendusprojekti arhitektuuri diagramm

### 3.4.2 Andmebaas

Andmebaasiks valiti Microsoft Azure SQL andmebaas. Sellega on võimalik rakenduste jaoks luua kättesaadavaid ja suure jõudlusega andmesalvestuskihte. Azure SQL andmebaas võimaldab töödelda nii relatsioonilisi kui ka mitterelatsioonilisi struktuure. Antud projekti jaoks valiti relatsiooniline andmebaas. (What is Azure SQL Database?, 2020)



Andmebaasi tehti tabelid iga lehekülje jaoks. Kokku on andmebaasis 10 tabelit. Neli tabelit on iseseisvad, mis tähendab, et nendel ei ole ühendusi teiste tabelitega. Ülejäänud 6 tabelit on omavahel ühenduses (Joonis 27).



Joonis 27 - Andmebaasi struktuur

Relatsioonilise andmebaasi kasutamisel on võimalus tuvastada andmeid ja neile juurde pääseda seoses mingite muude andmetega selles samas andmebaasis. Tavaliselt on relatsiooniliste andmebaaside struktuur loodud tabelitena. Antud projekti andmebaasi tabelitele on pandud nimed lehekülgede järgi. Igasse tabelisse on loodud loogiliste nimedega veerud. Näiteks *Instructors* tabelis on treeneri ID, eesnimi, perekonnanimi, tutvustus ja pilt.

Ühenduses olevad tabelid pärivad üksteiselt vajalikke andmeid. *Lessons* tabelis on 7 veergu, milles hoitakse tantsutundidega seotud andmeid. Nendest 4 veergu pärivad

andmeid teistest andmebaasi tabelitest. *CourseID* tuleb kursuste tabelist, *PlaceID* tuleb asukoha tabelist ning *InstructorID* ja *SecondInstructorID* tulevad treenerite tabelist.

Andmebaasi tabelite loomiseks kasutati SQL (ing. k. *Structured Query Language*) programmeerimiskeelt. Erinevate SQL käskudega saab luua tabeleid, mis sisaldavad vajalikke veerge vastavate andmetüüpidega (What is Azure SQL Database?, 2020). (Joonis 28)

```
CREATE TABLE [dbo].[Events](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](max) NULL,
    [PlaceId] [int] NOT NULL,
    [StartDate] [datetime2](7) NOT NULL,
    [EndDate] [datetime2](7) NOT NULL,
    [Description] [nvarchar](max) NULL,
    [Image] [nvarchar](255) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

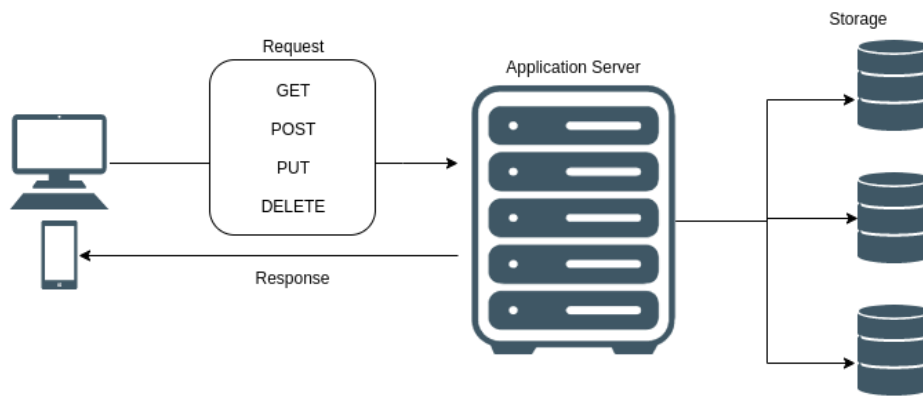
Joonis 28 - Tabeli loomine kasutades SQL programmeerimiskeelt

SQL programmeerimiskeelt kasutatakse ka tabelite ja andmetüüpide muutmiseks. Tabelites olevate andmete muutmine andmebaasis ja veebilehel toimub läbi *Admin* liidese.

Andmebaasi valimisel oli mitmeid põhjuseid. Valiti relatsiooniline andmebaas, mille puhul ei ole vaja tabelites olevaid andmeid korrata ning see säästab andmebaasi ruumi. Tabelite omavaheline sidumine nõuab vähem operatsioone, kui midagi muudetakse. Varasemalt oli töö autoril sellise andmebaasiga kogemus ning mingid teadmised olemas. Samuti oli ettevõtte juhil Azure Portal'is soetatud andmebaas, mida oli võimalik kasutada.

### 3.4.3 Back-end projekti arhitektuur

*Back-end* projekti arendamiseks valiti Visual Studio ASP.NET Core veebi API mall, mis kasutab REST arhitektuurstiili (Joonis 29). See võimaldab integreerida tarkvara süsteeme üle interneti. REST töötab koos HTTP protokolliga, mis sisaldab *get*, *post*, *put* ja *delete* meetodeid, mille kaudu on veebis võimalik klientidega suhelda. See protokoll määrab ära sõnumite edastamise ja toimingute vastamise erinevatele käskudele. REST võimaldab jaotada ressursid unikaalseteks kasutades globaalseid ID-sid, millega saab erinevaid objekte tuvastada. (Xiao-Hong, 2014)



Joonis 29 - REST arhitektuuri diagramm (Moody, 2019)

Projektis on *Models* ja *Controllers* kaustad, mis sisaldavad andmetega seotud loogikat. *Models* kaustas on klassid, mis määravad andmebaasi salvestatavate andmete omadused. *Controllers* kaust sisaldab klasse, milles on kirjutatud HTTP meetodid. Need meetodid võimaldavad suhtluse andmebaasiga ning vastavalt meetodile pärivad, muudavad, lisavad või kustutavad sealset infot. Iga meetodi juures on välja toodud aadress, mille kaudu saab näha veebis andmebaasist tulevaid andmeid.

### 3.4.4 Front-endi ja Admin liidese struktuur

*Front-endi* ja *Admin* liidese arendamiseks kasutati ReactJS raamistikku. Võrreldes teiste raamistikega ei ole ReactJS raamistikul kindlat ja nõutud arhitektuuri. React toimetab ainult vaadetega, mis jõuavad disainitud kujul kasutajateni. See võib raskendada koodi struktuuri haldamist suuremate projektide puhul. Seepärast oleks mõistlik mõelda välja mingisugune struktuur, mis teeks kogu projekti arusaadavamaks. (Bansal, 2019)

React koosneb komponentides, mis määrab ära mingisuguse projekti struktuuri. Sisukamate arendusprojektide jaoks oleks vajalik seda enda kasuks muuta, et oleks lihtne järge pidada, mis mida ja kus teeb. *Front-end* ja *Admin* projekti puhul jagati kõik veebilehel kuvatav komponentideks ning otsest arhitektuuristiili või mustrit ei kasutatud. Iga lehekülje haldamiseks tehti JavaScript fail. Nende lehekülgede jaoks loodi eraldi kaust, et kõik oleks ühes kindlas kohas ja kergesti leitavad. Andmebaasi ühendus ja loogika andmetele ligipääsu saamiseks loodi nii eraldi faili kui ka kausta. Disainifailid paigutati samuti ühte kohta. Nii oli projekti struktuur peale vaadates arusaadavam.

## 3.5 Disain

BST veebilehe disaini osas tehti võrreldes varasema versiooniga muudatusi. Väga suuri nõudmisi selle koha pealt ei olnud, kuid lepiti kokku üldine värvikombinatsioon. Samuti lähtuti päris stuudio ruumide disainist.

Igal vahelehel prooviti hoida sarnast stiili. Need koosnevad HTML programmeerimiskeele `<div>` konteineritest, kuhu sisse on lisatud veel vajalikud pildi- või tekstiväljad. Lisaks CSS failidele kasutati disainimiseks erinevaid stiiliraamistikke - Semantic-UI, Syncfusion, SweetAlert ja Material-UI. Viimast kasutati näiteks treenerite, kursuste, treeningpakettide ja ürituste lehtedel kaardi komponentide loomiseks. SweetAlert raamistikuga sai kuvada lihtsaid ja ilusaid teavitusi.

Kokkuvõttes pidi veebileht välja nägema võimalikult lihtne ja kasutajasõbralik. Välditi sisutihedat komponentide paigutamist lehtedele, mis oleks kasutaja silmale liiga kirju ja segadusse ajav.

## 3.6 Kood

Koodi kirjutamisel otsustati nende programmeerimiskeelte kasuks, mida oldi juba varem kasutatud. *Back-end* projektis kasutati C# ning *front-end* ja *Admin* projektis JavaScripti, HTMLi ja CSSi.

### 3.6.1 Back-end projekt

*Back-end* projekti ühendus SQL andmebaasiga loodi *appsettings.Development.json* failis, kuhu lisati autentimiseks vajalik koodirida (Joonis 30).

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "ConnectionStrings": {
    "DefaultConnection": "Server=tcp:bachatastudio.database.windows.net,1433;Initial Catalog=bachatastudiodb;Persist Security Info=False;"
  }
}
```

Joonis 30 - Andmebaasi ühendus *appsettings.Development.json* failis

*Models* kausta on loodud kõik vajalikud klassid, kus määrati ära andmebaasi salvestatavad omadused (Joonis 31). Kõikidele objektidele on määratud oma ID, mille

kaudu saab vastavat objekti lugeda, muuta või kustutada. Andmebaasi on samuti loodud tabelid vastavalt samade nimedega.

```
namespace api.Model
{
    public class Packages
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        public string Price { get; set; }
        public string PaymentLink { get; set; }
    }
}
```

Joonis 31 - *Models* kaustas määratletud *Packages* klass

Igale *Models* kausta klassile on loodud oma kontrolleri, mis pärinevad *ControllerBase* klassist. Need võimaldavad teha HTTP päringuid. Igale klassile on kirjutatud loogika CRUD toimingute kohta (Joonis 32). Need kontrolleriid sisaldavad lugemise, loomise, muudatuste tegemise ja kustutamise meetodeid. Joonisel 32 on näidatud kahte meetodit lisamisest ja kustutamisest *InstructorsController.cs* failis. Kustutamise meetodis kontrollitakse kõigepealt, kas sellise ID-ga objekt on üldse olemas. Kui ei ole, siis kutsutakse välja *NotFound()* tagastuse tüüp, mis kuvab veateate. Kui vastava ID-ga objekt leitakse, siis eemaldatakse see andmebaasist, salvestatakse toiming ning tagastatakse instruktorite list. Kontrolleriid on ühenduses *DataContext.cs* failiga, kus on määratud ära iga klassi jaoks *get* ja *set* meetodid. *Get* meetod tagastab väärtuseid. *Set* meetod lisab objektile uue väärtuse.

```

[HttpPost]
public async Task<ActionResult<Instructor>> PostInstructors(Instructor instructors)
{
    _context.Instructors.Add(instructors);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetInstructors", new { id = instructors.Id }, instructors);
}

// DELETE: api/Instructors/5
[HttpDelete("{id}")]
public async Task<ActionResult<Instructor>> DeleteInstructors(int id)
{
    var instructors = await _context.Instructors.FindAsync(id);
    if (instructors == null)
    {
        return NotFound();
    }

    _context.Instructors.Remove(instructors);
    await _context.SaveChangesAsync();

    return instructors;
}

```

Joonis 32 - Lisamise ja kustutamise meetodite näited *InstructorsController.cs* failis

Kontroller klassides on iga meetodi juures toodud välja aadress. Minnes veebis valitud aadressile, kuvatakse andmebaasist tulevaid andmeid (Joonis 33).

```

[
  - {
    id: 1,
    name: "One course (8h)",
    description: "Bachata Couples, 2 hours a week",
    price: "49",
    paymentLink: "https://mairu.ee/jWsYWg20"
  },
  - {
    id: 2,
    name: "Two courses (16h)",
    description: "Bachata Couples, 4 hours a week",
    price: "69",
    paymentLink: "https://mairu.ee/4FHe85Kz"
  },
  - {
    id: 3,
    name: "Shine Ladies Open",
    description: "Bachata Lady Styling, 1 hour a week",
    price: "33",
    paymentLink: "https://mairu.ee/5THC1jMx"
  },
  - {
    id: 9,
    name: "Shine Ladies & one course",
    description: "Lady Styling, Bachata Couples",
    price: "65",
    paymentLink: "https://mairu.ee/h9h9V45L"
  }
]

```

Joonis 33 - *Packages* aadressi sisestamisel saadud tulemus

### 3.6.2 Front-end projekt

*Front-end* kasutajaliidese projektis loodi iga lehekülje jaoks eraldi JavaScript fail. JavaScript failides kasutati ka HTML koodi, millega oli võimalik leheküljed seksioonideks jagada. See sisaldas `<div>` konteinereid, mille sisse sai lisada erinevaid elemente (tekstiväljasid, sisestusväljasid, pildiväljasid jms) (Joonis 34).

```
JS contact.js ×
src > components > JS contact.js > ...
33   <div>
34     <h1 className='contact'>Contact Us</h1>
35     <div id="contact-container">
36       <div className="contact-info">
37         <h4>
38           Contact Information
39         </h4>
40         <p className="alt">Fill up the form and our team will get back to you!</p>
41         <div className="icon-text">
42           <i class="fa fa-phone" aria-hidden="true"></i>
43           <span className="info">+ 372 50 21 770</span>
44         </div>
45         <div className="icon-text">
46           <i class="fa fa-envelope" aria-hidden="true"></i>
47           <span className="info">bachata.studio.tallinn@gmail.com</span>
48         </div>
49         <div className="icon-text">
50           <i class="fa fa-map-marker" aria-hidden="true"></i>
51           <span className="info">Veerenni 29/1, Tallinn 10135</span>
52         </div>
```

Joonis 34 - Näide *contact.js* HTML elementidest

Iga JavaScripti faili jaoks loodi ka CSS fail, kuhu kirjutati koodi veebilehe disainimiseks. Iga HTML elemendile on pandud nimi *className* või *ID* atribuute kasutades. Need on vajalikud, et CSS failides vastavat konteinerit või elementi disainida. Vastavale elemendile ID-ga viitamiseks kasutatakse # sümbolit ja *className* viitamiseks punkti. (Joonis 35)

```
# contact.css X
src > components > # contact.css > #contact-container
1 #contact-container{
2   display: flex;
3   flex-direction: column;
4   justify-content: center;
5   align-items: center;
6   width: 1200px;
7   overflow: hidden;
8   padding: 10px;
9   margin-left: 120px;
10  margin-top: 45px;
11  backdrop-filter: blur(15px);
12  border: 4px double white;
13  box-shadow: 0px 10px 25px 0px;
14 }
15
16 .contact-info{
17   background-color: rgb(177, 165, 143);
18   flex: 0 1 40%;
19   margin-left: -700px;
20   width: 40%;
21   padding: 20px;
22   display: flex;
23   flex-direction: column;
24 }
```

Joonis 35 - Näide *contact.css* failist

Andmebaasi ühenduse loomiseks lisati projekti *api* kaust ja sinna sisse *fetch.js* fail. Seal on ära määratud aadress juurutatud API-le ligipääsu saamiseks. Lisaks sellele on kirjutatud iga lehekülje kohta funktsioon ning määratud aadress, mille kaudu pääseb andmebaasis olevatele andmetele ligi. (Joonis 36)

```
JS fetch.js X
src > api > JS fetch.js > getCourse
1 const apiEndpoint = 'https://bstlopuprojekt.azurewebsites.net'
2
3 const apiGet = (path) => fetch(apiEndpoint + path)
4   .then(response => response.json())
5
6
7 function getInstructors () {
8   return apiGet('/api/instructors')
9 }
10
11 function getCourse () {
12   return apiGet('/api/courses')
13 }
```

Joonis 36 - Andmebaasi ühendus *fetch.js* failis



JavaScript failidesse on lisatud andmete kätte saamiseks loogika kasutades *useState* ja *useEffect*. *UseEffect* kutsus välja *fetch.js* failis oleva koodi, et saada kätte andmebaasist vajalikke andmeid ning *useState* aitab neid andmeid käsitleda. Loodud on *get* ja *set* muutujad. *Get* muutujat kasutatakse koos *map()* meetodiga HTML koodis. *Map()* meetod kutsus välja tagastusfunktsiooni iga objekti jaoks andmebaasist ning tagastab need, mis omavad väärtust. *Map()* meetodile on lisatud parameeter, mille kaudu saab andmebaasist informatsiooni kätte vastavalt tabeli väljade nimele. HTML elementidesse, kus on vaja kuvada informatsiooni, on lisatud parameetri nimi koos andmebaasi tabeli välja nimega. (Joonis 37)

```
const Events = () => {
  const [getEvents, setEvents] = useState([]);

  useEffect(() => {
    fetch.getEvents()
      .then(data => setEvents(data))
  }, [])

  return(
    <body className="events-body">
      {getEvents.map((events) => (
        <div class="scene">
          <div key={events.id} className="card-events">
            <div className="card__face card__face--front">
              <img src={events.image} alt="" />
              <h1 className="event-header">{events.name}</h1>
            </div>
            <div className="card__face card__face--back">
              <img src={white} alt="" />
              <p className="event-description">{events.description}</p>
            </div>
          </div>
        </div>
      )
    )}
  </body>
  )
}
```

Joonis 37 - Andmete kätte saamine ja kuvamine HTML elementides

### 3.6.3 Admin liides

*Admin* liidese jaoks oli vaja luua funktsioonid, et veebilehel olevat informatsiooni saaks muuta ja kustutada ning vajadusel sinna midagi uut lisada. Selle jaoks loodi igale leheküljele *components* ja *actions* kaustad. *Components* kausta tehti kolm JavaScripti faili, mille loogika kehtib kõikide lehekülgede puhul.

Näiteks kirjelduse lehekülje jaoks tehti *About.js*, *aboutForm.js* ja *useForm.js* failid. *About.js* failis loodi HTML elemendid veebilehe seksioonidesse jagamiseks (Joonis 34) ja andmebaasist andmete kätte saamise loogika (Joonis 37) sarnaselt *front-end* projektile. Vajalikud sisestus- ja tekstiväljad informatsiooni muutmiseks ja lisamiseks kirjutati *AboutForm.js* faili. *UseForm.js* failis kirjutati funktsioonid, mis haldavad uut väljadesse kirjutatud informatsiooni ning sisestusväljade tühendamist (Joonis 38).

```
const handleInputChange = e => {
  const {name, value} = e.target
  const fieldValue = {[name]: value}
  setValues({
    ...values,
    ...fieldValue
  })
  validate(fieldValue)
}

const resetForm = () => {
  setValues({
    ...initialFieldValues
  })
  setErrors({})
  setCurrentId(0)
}
```

Joonis 38 - Funktsioonid uue informatsiooni ja väljade tühendamise haldamiseks

*Actions* kaust sisaldab andmebaasi ühenduse ja CRUD funktsioonide faile. Andmebaasi ühendus loodi samamoodi nagu *front-end* projektis (Joonis 36). *Actions.js* faili kirjutati *create*, *read*, *update* ja *delete* funktsioonid, et *Admin* liideses oleks võimalus neid samu toiminguid teha (Joonis 39).

```
// UPDATE
export const update = (id, data, onSuccess) => dispatch => {
  data = formateData(data)
  api.aboutReducer().update(id, data)
  .then(res =>{
    dispatch({
      type : ACTION_TYPES.UPDATE,
      payload : {id,...data}
    })
    onSuccess()
  })
  .catch(err => console.log(err))
}
```

Joonis 39 - *Update* funktsioon *actions.js* failis

*Admin* liidese lehekülgede disainimiseks kasutati CSS programmeerimiskeelt.

### 3.7 Testimine

Arendusprojekti testiti mitmel erineval viisil. *Back-end* projekti testimiseks kasutati Postmani, mis võimaldab teha koostööd API-dega. Selle abil kontrolliti, kas vajalik SQL andmebaasi ühendus on olemas ning kas kõik brauserisse sisse kirjutatud aadressid tagastavad õiget informatsiooni.

Testide jaoks tehti ka eraldi projekt, kuhu kirjutati NUnit teste. NUnit on avatud lähtekoodiga .NET keelte testimiseks mõeldud ühikutestimise raamistik. See raamistik pakub konkreetseid mustrite ja meetodite komplekti, mida testide kirjutamisel on vaja kasutada. (Sheth, 2021) Projekti kirjutati 200 testi, mis kontrollisid rakenduse töötamist ning koodi kattuvus on 42.50%.

Iga mudeli ja kontrolleri jaoks loodi testi klass ning sinna sisse testimise meetodid. Iga klassi puhul testitakse ära, kas on võimalik saada ja määrata vastava objekti vajalikke omadusi. Näiteks *InstructorTest* klassis on testitud nimede saamine ja määramine ning treenerite kirjelduse lisamine. (Joonis 40).

```
[TestMethod]
public void InstructorSetLastNameTest()
{
    instructor.LastName = random;
    Assert.AreEqual(random, instructor.LastName);
}

[TestMethod]
public void TestInsturctorGetLastName()
{
    string name = instructor.LastName;
    Assert.AreEqual(null, name);
}

[TestMethod]
public void InstructorSetDescriptionTest()
{
    instructor.Description = random;
    Assert.AreEqual(random, instructor.Description);
}
```

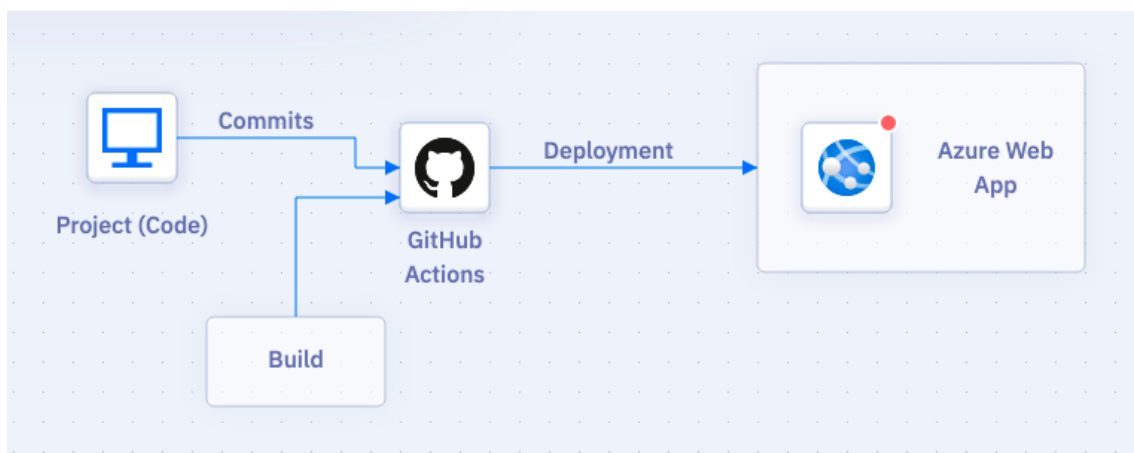
Joonis 40 - Treenerite klassi testide näited

Veebilehte ja selle funktsionaalsusi testisid ka stuudio tiimi liikmed. Tagasiside oli positiivne. Samuti testiti *Admin* liidest ning kontrolliti, kas ligipääs on ainult õigustega

inimestel. See katse oli edukas. Studio mäenedžer sai teha vajalikke toiminguid läbi *Admin* liidese.

### 3.8 Arendusprojekti juurutamine

Esmakordselt prooviti arendusprojekti juurutamist Azure Portali ja GitHubi abil (Joonis 41), et veebileht klientidele kättesaadavaks teha. Selleks laeti kood üles GitHubi, mis ühendati Azure Portaliga. Seal oli vajalik luua *front-end* ja *back-end* projekti ning *Admin* liidese jaoks eraldi projektid. *Front-end* projekti ja *Admin* liidese jaoks loodi Azure Static Web App (*eesti k.* Azure staatiline veebirakendus), mis ühendati GitHubi repositooriumiga. Pärast igit *commiti* tekitati GitHubi töövood, mis pärast edukat lõpetamist jõudsid Azure Portalisse. Iga Azure projekti jaoks genereeriti aadress, millele vajutades avanes uus vaheleht juurutatud veebilehest.



Joonis 41 - Juurutamise protsess

*Back-end* projekti jaoks loodi Azure App Service (*eesti k.* Azure rakenduste teenus), kuhu avalikustati projekt otse Visual Studiost. Samuti genereeriti selle projekti jaoks aadress. Andmebaasist andmete kuvamiseks tuli aadressile lisada vastava klassi nimi (juurutatud projekti aadress + /api + /instructors).

Eelnevalt oli probleeme tekkinud hilise juurutamisega, kus projekt sisaldas juba paljusid komponente. Seekord otsustati kõige algelisem projekt kohe juurutada. Lisati väiksemaid komponente ja funktsionaalsusi ning juurutati iga väiksema sammu tagant uuesti. Nii ei tekkinud juurutamisprotsessis probleeme ning kõik läks sujuvalt.

## 4 Veebilehe analüüsimine ja järeldused

### 4.1 Tehnilise teostuse põhjendus

#### 4.1.1 Nõuded

Veebilehe nõuded olid algusest peale arusaadavad ja põhjendatud. Dünaamiline veebileht oli vajalik nii *Admin* liideses toimingute tegemiseks kui ka klientidele informatsiooni edastamiseks. Varasemalt oli informatsiooni edastamine treenerite, asukoha, kursuste, ürituste, treeningpakettide jms kohta toimunud läbi suhtluskanalite, mis oli aeganõudev. Kõikidele klientidele oli vaja teha tantsu tasemete põhjal mitu vestlus gruppi.

*Admin* liidese põhimõte oli uue informatsiooni edasi andmine ühes kindlas kohas, mitte erinevates grupi vestlustes. Seal oli võimalik uuendada kogu veebilehte ning lisada vajalikku informatsiooni, mida klientidel oli võimalik vaadata. Lisati ka korduma kippuvate küsimuste lehekülg, kuhu kliendid saavad küsimuste korral pöörduda.

Maksete sooritamise toimus varem studios kohapeal või oli vajalik klientidele saata studio konto andmed. See nõudis ettevõtte juhil studios kohapeal käimist igapäevaselt ning kõik maksetega seotud tegevused oli liigselt keerulised.

Nõuete täitmisel sai valmis veebileht, mis vastas BST tantsustuudio juhi ootustele. See vähendas olulisel määral manuaalset tööd ja kergendas elu ka klientide jaoks. Kõik vajalik oli ühes kohas ning enam ei olnud vaja kõikide klientidega igapäevaselt eraldi suhelda.

#### 4.1.2 Arhitektuur

Arendusprojekti arhitektuuri prooviti hoida võimalikult puhtana. See võimaldab vähendada inimressursse süsteemide hooldamiseks ja loomiseks. Arhitektuuri stiile on palju ja erinevaid, kuid puhta arhitektuuri saavutamiseks jälgivad need mingeid kindlaid reegleid. Põhieesmärgiks on projekti jaotamine mitmeks kihiks. (Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, First Edition, 2017)

Antud arendusprojekti puhul jälgiti sama põhimõtet. Projekt jaotati 3 osaks, milleks olid *back-end*, *front-end* ja *Admin* liides. Kõik olid eraldi seisvad projektid ning nende vahelise suhtluse loomiseks tuli need ühendada. *Back-end* projektiga olid ühenduses nii *front-end* kui ka *Admin* liides. Kuid viimase kahe ühenduse ei olnud vajalik ega oluline,

sest põhieesmärgiks oli andmebaasiga suhtlemine ning *back-end* projekti loogika abil erinevate toimingute tegemine.

Puhta arhitektuuri kasutamisel peaks olema võimalik testida eraldi nii ärilist poolt kui ka andmebaasi ja kasutajaliidest. Antud projekti puhul testiti eraldi *back-end* projekti kasutades NUnit teste. Samuti on välja toodud Robert C. Martin raamatus, et kasutajaliides peaks olema sõltumatu. Puhta arhitektuuri järgi peaks olema võimalik seda muuta nii, et ülejäänud süsteem ei muutu. BST veebilehe kasutajaliidesed on eraldi seisvad ning nendes muudatuste tegemine ei kahjusta kuidagi API poolt. Puhast arhitektuuri defineerib ka andmebaasist sõltumatus. Vajadusel peaks olema võimalik andmebaasi vahetada. (Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, First Edition, 2017)

Arendusprojektis võeti kasutusele REST arhitektuurimuster. Alternatiivina oleks võinud kasutada ka GraphQLi, mis on päringukeel rakenduste loomiseks. REST ja GraphQL on vägagi sarnased mustrid. Mõlemad teevad päringuid läbi veebiaadressi ning tagastavad andmebaasis oleva info, mõlema mustri puhul on võimalik ära määrata, kas API päring on mõeldud lugemiseks või kirjutamiseks jms. (Stubailo, 2017)

Antud projekti arendamisel otsustati kasutada REST arhitektuurimustrit, sest GraphQL on väga sarnane ning antud projekti puhul ei ole suurt vajadust seda kasutada. Mustri valiku tegemisel arvestati varasema kogemusega ning erinevate artiklite põhjenduste ja arvamustega. REST on väga domineeriv arhitektuurimuster, mida kasutatakse kõikjal maailmas. Üks suurimaid plusse on see, et antud muster on loodud pikaajaliseks kasutamiseks. Kui projekt luuakse klientidele, kes seda pidevalt kasutavad, on REST üks parimaid valikuid. (Doerrfeld, 2019)

### **4.1.3 Disain**

Veebilehe disainimine oli autorile üks lemmikumaid projekti arendamise jooksul. Suuri nõudmisi ettevõtte juhi poolt ei olnud, seega oli disaini loomine antud töö autori enda otsustada.

Lisaks CSS programmeerimiskeelele kasutati erinevaid disainiraamistikke, mis pakuvad suurel hulgal võimalusi veebilehtede disainimiseks. *Front-end* projektis ja *Admin* liideses võeti kasutusele Ant Design, Font Awesome, Material-UI, Semantic UI, Syncfusion jms.

Neid kasutades on lihtne luua juba olemasoleva disainiga komponente. Disaini elementidest kasutati erinevaid kaardi komponente, tabeleid, sisestusväljasid, nuppe ja palju muid vajalikke elemente.

Disain muutub ajas kiiresti ning vajadusel värskendatakse veebilehte ka tulevikus. Praeguseks on veebileht kasutajatele avalikuks tehtud ning tagasiside on olnud positiivne. Disainiga oli rahul ka BST tantsustudio juht ning terve ettevõtte meeskond.

#### 4.1.4 Kood

Koodi kirjutamisel prooviti järgida puhta koodi põhimõtteid. Nendeks on hea koodi loetavus, arusaadavus ja lihtsus; funktsioonid ja meetodid teevad ainult ühte asja, milleks nad on mõeldud; koodis olevad klassid, meetodid, funktsioonid ja muutujad peavad olema asjakohaste nimedega ja sõltuvusi peaks olema minimaalselt. (Martin, Clean Code: A Handbook of Agile Software Craftsmanship, 2008)

Eelnevalt mainitud üks puhta koodi põhimõtteid on selle loetavus, lihtsus ja arusaadavus. Koodi on võimalik kirjutada väga keeruliselt ja ebaselgelt. Samuti teevad puhtas koodis funktsioonid ja meetodid ainult ühte asja, et oleks hea ülevaade koodi töötamisest. Antud projektis on samuti kõik vajalike toimingute jaoks loodud funktsioonid ja meetodid ühe kindla asja jaoks. Näiteks *Admin* liideses on lisamise, kustutamise ja muutmise meetodid, mis teevad ainult seda, mida nad on loodud tegema. Andmebaasist andmete kätte saamiseks loodi jällegi eraldi funktsioon, mis suhtleb ainult andmebaasiga.

Kõik klassid, funktsioonid, meetodid ja muutujad peaksid puhta koodi järgi olema selgelt ja teemakohaselt nimetatud. See tekitab arusaadava struktuuri projektis ning otsitavad objektid on kergesti leitavad. *Front-end* projektis ja *Admin* liideses määrati lehekülgedele just need nimed, mis kuvatakse ka veebilehel. Kõik klasside ja muutujate nimedeks olid nimisõnad. Andmebaasi ühenduse ja andmete kättesaamise loogika klassid ja funktsioonid nimetati vastavalt *back-end* projekti funktsioonidele. Lisamise, muutmise ja kustutamise funktsioonid nimetati tegusõnadega, et oleks aru saada, mida need täpselt teevad. Sama põhimõtet kasutati ka *back-end* projektis.

Kõik komponendid prooviti teha võimalikult iseseisvad, et ei oleks palju sõltuvusi. Minimaalsete sõltuvustega projekte on kergem hooldada. Vajalikud muudatused võivad rohkete sõltuvuste korral tekitada rakenduse töötamises probleeme.

Üldise hea väljanägemise jaoks jäeti meetodite ja funktsioonide vahele tühjad read, et kood ei oleks hunnikus koos. Kommentaarid lisati teatud funktsioonidele, et autoril oleks parem arusaadavus.

*Back-end* projektis on 23 klassi ning 1152 koodirida. *Front-end* projektis on 1891 koodirida ning 39 faili. *Admin* projektis on 79 faili ja 3031 koodirida.

#### **4.1.5 Testid**

Projekti testimiseks võeti kasutusele NUnit testimisraamistik. Selle kasuks otsustati varasema kogemuse põhjal. Teste kirjutati *back-end* projektis, kus testiti kõikide kontrollite ja mudelite klasse ning nende meetodeid, et olla kindel projekti töötamises.

React projektides prooviti samuti testimist, kuid see oli juba keerulisem. Sellega ei oldud kunagi varem kokku puutunud ning oskused puudusid. Katsetati *React Testing Library*'t, mis on mõeldud komponentide testimiseks (Borenkraout, 2020). Kirjutati mitmeid erinevaid teste, kuid need ei olnud edukad. Samuti sai aeg otsa ning testimine jäi React projektides poolikuks.

On võimalik, et vajadusel arendatakse projekti edasi ka tulevikus. Lisaks muudele muutustele projektis oleks vajalik tegeleda ka testimisega. See kindlustab projekti töötamise ning annab vajalikke oskusi ka järgmiste projektide testimiseks.

## **4.2 Tööriistade valiku põhjendus**

Projekti arendamiseks valiti ASP.NET Core raamistik, millega oli varasem kokkupuude ning mis nõuab programmeerimisoskusi ja teadmisi. Alternatiivina oleks võinud kasutada ka Wordpressi. Tööriistade valiku põhjenduseks võrreldi Wordpressi ja ASP.NET Core'i.

Wordpress on veebilehtede tegemiseks loodud tarkvara. See on veebilehe sisu haldamise süsteem, mida saavad kasutada ka inimesed, kes ei ole programmeerimisega kokku puutunud. See annab võimaluse väga lihtsalt luua veebilehti, blogisid, foorumeid jms. (What Is WordPress? Explained for Beginners, 2021) Küsimus tekib selles, kas see on õige ja mõistlik valik keerukamate veebilehtede turvaliseks ja kestvaks arendamiseks. Wordpress ei paku kõrget turvalisust. See on üks laialdasemalt kasutatavaid platvorme ning küberkurjategijatele on see põhiliseks sihtmärgiks. Samuti saavad kasutajad luua ise erinevaid pistikprogramme, mida on võimalik kasutusse võtta, kuid need ei ole alati kõige



puhtamad ja turvalisemad. Veebilehe hooldamiseks vajaminev tugipunkt on Wordpressi jaoks erinevad veebifoorumid, mis pakub kasutajatele minimaalset tuge. (Pros and Cons of WordPress – Should You Use It?, 2021) Wordpress pakub laias valikus teemasid, mis aga sisaldavad palju mõttetut koodi ning võivad veebilehe kiirust ja laadimisaega vähendada. Wordpress nõuab ka teatud uuenduste tegemist, mille tõttu võivad katki minna kirjutatud funktsioonid, mis kahjustavad veebilehte. (ArcStone, 2020)

ASP.NET Core on platvormide vaheline raamistik, mis pakub erinevaid funktsioone veebirakenduste ja veebi API-de välja töötamiseks. See töötab nii Windowsi, Linuxi kui ka Maci operatsioonisüsteemides. Raamistikul on Microsofti tugi, mis tagab rakenduste ohutuse ja mõistlikkuse. Rakenduste hooldus on lihtsam kasutades olemasolevaid vahendeid, mis toetavad koodibaasi ning aitavad varakult märgata vigu pakkudes välja lahendusi. ASP.NET Core pakub tuge veebi API-le ja teeb selle ühendamise *front-endiga* lihtsamaks. Antud arendusprojekti kasutati samuti Visual Studio veebi API malli. Lisaks eelnevalt toodule pakub ASP.NET Core tuge asünkroonsete programmeerimismustrite kasutamisel, mis parandab oluliselt rakenduste jõudlust. (Razvalinov, 2019) Üheks suurimaks teemaks rakenduste arendamisel on turvalisus. ASP.NET Core võimaldab hallata rakenduste turvalisust ja konfigureerimist. See sisaldab erinevaid funktsioone (autentimine, autoriseerimine, andmekaitse, HTTPS jõustamine, ressursside taotlemine teistelt domeenitelt jms), mis teevad rakenduse tugevamaks ja turvalisemaks. (Overview of ASP.NET Core Security , 2018) ASP.NET Core rakendusi on võimalik ühendada ka React raamistikel põhinevate rakendustega. Samuti on võimalik ASP.NET Core rakendusi integreerida Microsoft Azure'ga, kus saab arendusprojekte kasutajatele avalikuks teha.

Antud töö autori jaoks oli tööriistade valik algusest peale selge. Wordpressi kasutamine ei tulnud mõttesse. Läbitud programmeerimise õppeainetes kasutati nii ASP.NET kui ka ASP.NET Core raamistikku, mis on uuem versioon ASP.NET'st. Suur põhjus kasutatud tööriistade valikul oli programmeerimisoskuste arendamine ja teadmiste kinnistamine. Autor leidis, et äärmiselt vajalik on saada aru, kuidas mingid tarkvarad ja platvormid töötavad, kuidas kirjutada efektiivsemalt koodi ning kuidas erinevaid projekte omavahel ühendada. Need teadmised, oskused ja ka enesekindlus, mis sellega kaasa tuleb, on kindlasti suureks plussiks tulevastel töökohtadel. Varasemalt arendatud meeskonna projektis ei olnud võimalik kõikide asjadega tegeleda, sest tööülesanded olid meeskonna

vahel ära jaotatud. Lõputööga seoses oli võimalus tegeleda andmebaasiga, *back-end* ja *front-end* projektiga ning esimest korda proovida terve projekti juurutamist.

### **4.3 Teostatud tööde logimine**

Arendusprojektis toimuvatel tegevustel oli vajalik silma peal hoida, et protsessi oleks kergem jälgida. Selleks kasutati nii TogglTrack veebirakendust kui ka Google Docsi dokumenti. TogglTracki pandi kirja iga nädalaselt kõik töötunnid ning kirjeldused tehtud töödest (Lisa 2-13). Sinna logiti koosolekud juhendajatega, projekti arendamisele ja erinevatele artiklite lugemisele kulunud aeg ning dokumentatsiooni kirjutamine. Kokku tehti tööd 315 tundi.

Google Docs'i dokumenti kirjutati üles tööülesanded nädalate kaupa. Igal pühapäeval pandi paika ülesanded järgmiseks nädalaks. Kõike ei jõutud iga nädal valmis ning töö lükkus edasi järgmisesse nädalasse. Ülesannete lahendamise etapid märgiti erinevate värvidega, et oleks parem arusaada, mis on tehtud (roheline), mis tegemisel (kollane) ning mis vajab tegemist (punane).

### **4.4 Hinnang projekti teostusele**

Üldine töö sujus hästi. Antud projekti arendamiseks kulus aega 11 nädalat ja 2 päeva. Kõik ülesanded ja nõuded olid paigas ning kõik vajalik sai tehtud. Projekti arendamisega tegeles antud töö autor üksi. Juhendajad nii kooli kui ka ettevõtte poolt olid vajadusel alati abiks. Enamasti käis suhtlus läbi veebi, kuid oli olukordi, kus kohtuti ettevõtte juhiga.

Töö autoril on hea meel, et projekti otsustati arendada algusest peale. Soov oli sügavamalt õppida projektide omavahelist suhtlemist ning andmebaaside ja koodi töötamist. Eelmise semestri „Meeskonnaprojekt“ aine raames olid kõik ülesanded jaotatud ning töö autor tegeles enamasti *front-end* projekti arendamisega. Soov oli tegeleda rohkem ka andmebaasiga, *back-end* projekti arendamisega ja *Admin* liidese loomisega. Nendega seotud tööülesanded olid keerukamad. Nende tegemise käigus õppis autor palju ning tundis suuri edusamme enda arengus. Antud töö puhul kulus palju aega ka erinevate artiklite lugemisele, et teadmiste pagasit täiendada. Autor tunneb ennast kindlamalt programmeerimisega seotud teadmistes ning IT alasel töökohal. Samuti tekkis arusaam,

mis on täielik projekti arendus (*ing. full stack development*) ning selgus, mida sellest tegelikult meeldib teha. Autor leidis, et *front-end* projekti arendamine ja disainimine on need, mida võiks rohkem ka tulevikus teha.

Arendamistöös oli nii kergemaid kui ka raskemaid ülesandeid. Kiirelt ja lihtsalt läks staatilise ning hiljem dünaamilise veebilehe loomise ja disainimisega, sest varasemalt oli seda mitmeid kordi tehtud. Rohkem aega kulus *Admin* projekti loogika kirjutamiseks, et oleks võimalus läbi liides lisada, muuta ja kustutada informatsiooni nii veebilehel kui ka andmebaasis. Aja kulu oli suurem ka juurutamise protsessis ning andmebaasi struktuuri loomisel. Maksmise funktsionaalsus võttis samuti rohkem aega, sest seda polnud varem tehtud. Palju aega läks erinevate teenuspakkujate ning teenusete integreerimisvõimaluste uurimisele. Uueks ülesandeks oli ka õiguste andmine *Admin* kasutajatele.

Projekti arendamise käigus oli ka tagasilööke ja probleeme, mis tulenesid autori puudulikest teadmistest ning isiklikest elu sündmustest. Arendus protsess aeglustus vahepeal paariks nädalaks ning seetõttu ei jõutud kõikide asjadega nii ideaalselt valmis, kui oleks võinud. Probleemid, mis tulenesid autori puudulikest teadmistest, said enamasti lahendatud. Suures osas proovis autor leida ise teatud probleemi kohta informatsiooni ja lahendusi, mida katsetada. Ette tuli ka täielikke seisakuid, kus pöörduti abi saamiseks targemate ja vanemate inimeste poole, kes teemat paremini valdasid.

Antud projekti koostamisel paika pandud nõuded said täidetud ning tulemusega olid mõlemad osapooled rahul, kuid projekti arendamisele veel punkti ei pandud. Ettevõtte juhil ja tantsustuudio tiimil tekkis mitmeid ideid, kuidas veel arendusprojekti täiendada ning paremaks muuta. Autor oli ka ise koosolekutele kaasatud, kus arutati veebilehe tuleviku üle ning on valmis jätkama veebilehe arendamist.

## 5 Kokkuvõte

Antud lõputöö raames valmis arendusprojekt ettevõttele Bachata Studio Tallinn, milleks oli veebileht koos *Admin* liidesega. Veebilehe ja *Admin* liidese loomise põhjuseks oli klientidele informatsiooni edastamise, maksete tegemise ja studio mäenedžerile veebilehe haldamise võimaldamine. Meeskonna projektina valminud veebileht oli puudulik ning vajab edasiarendamist.

Eesmärgiks oli saada valmis uuendatud disainiga veebileht, mille kaudu on võimalik klientidele edastada informatsiooni studio, treenerite, tantsupakettide, kursuste, tunniplaani, ürituste jms kohta. Lisaks sellele tuli pakkuda klientidele registreerimise ja maksimise võimalust läbi veebilehe. *Admin* liidest oli vaja selleks, et lisada, muuta ja kustutada informatsiooni.

Projekt jaotati kolmeks osaks: *back-end* ehk API, *front-end* ja *Admin* liides. *Back-endi* arendus toimus Visual Studios kasutades C# programmeerimiskeelt. *Front-end* ja *Admin* liidest arendati Visual Studio Code'is kasutades JavaScripti, HTMLi ja CSSi. Tööprotsessil hoiti silma peal iganädalaste progressi raportide järgi.

Arendusprojekti lõpuks sai valmis uue disainiga dünaamiline veebileht, millel kuvatakse andmebaasist tulevat informatsiooni. Projekt ühendati makseportaaliga Mairu, mille kaudu on võimalik klientidel tasuta treeningpakettide eest. Valmis *Admin* liides kogu veebilehe haldamiseks, millele on ligipääsu ainult õigustega kasutajatel.

Projekti arendamise käigus järgiti ettevõtte poolseid nõudeid ja ideid. Arendamiseks kasutati mõistlikke ja turvalisi tööriistu mõeldes pikemale perspektiivile. Ettevõtte juht ning töö autor on tulemustega rahul ning klientide tagasiside oli positiivne. Veebilehe nõudmised võivad tulevikus muutuda ning pole välistatud, et antud töö autor jätkab BST veebilehe arendamisega.

## Kasutatud kirjandus

- Razvalinov, A. (15. Märts 2019. a.). *ASP.NET Core 8 Pros and 3 Cons*. Allikas: <https://www.ukad-group.com/blog/aspnet-core-8-pros-and-3-cons/>
- What Is WordPress? Explained for Beginners*. (27. Aprill 2021. a.). Allikas: <https://kinsta.com/knowledgebase/what-is-wordpress/>
- Overview of ASP.NET Core Security* . (24. Oktoober 2018. a.). Allikas: <https://docs.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-5.0>
- Pros and Cons of WordPress – Should You Use It?* (2. Aprill 2021. a.). Allikas: <https://cyberchimps.com/blog/pros-and-cons-of-wordpress/>
- ArcStone. (Jaanuar 2020. a.). *The Pros and Cons of Choosing WordPress for your Website CMS* . Allikas: <https://www.arcstone.com/blog/pros-and-cons-of-wordpress-cms>
- Xiao-Hong, L. (2014). Research and Development of Web of Things System Based on Rest Architecture. (lk 744-745). Hunan, China: IEEE. Allikas: <https://ieeexplore.ieee.org/document/6977704>
- Moody, M. (18. Detsember 2019. a.). *GraphQL vs. REST API Architecture* . Allikas: <https://medium.com/swlh/graphql-vs-rest-api-architecture-3b95a77512f5>
- Bansal, S. (16. August 2019. a.). *React Architecture | Best Practices*. Allikas: <https://suneetbansal.medium.com/react-architecture-best-practices-6ac9cfd715d5>
- Kopecky, C. (24. August 2020. a.). *Introducing React Design Patterns: Flux, Redux, and Context API*. Allikas: <https://www.educative.io/blog/react-design-patterns-best-practices>
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design, First Edition*. Pearson.
- Doerrfeld, B. (3. Oktoober 2019. a.). *Is REST Still A Good API Design Style to Use?* Allikas: <https://nordicapis.com/is-rest-still-a-good-api-design-style-to-use/>
- Stubailo, S. (27. Juuni 2017. a.). *GraphQL vs. REST* . Allikas: <https://www.apollographql.com/blog/graphql-vs-rest-5d425123e34b/>
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. New Jersey: Prentice Hall.
- What is Azure SQL Database?* (21. September 2020. a.). Allikas: <https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview>
- Sirel, L., Kirs, K., Sein, M., & Andresen, K. (2020). *Meeskonnaprojekti aruanne*. Tallinn.
- Sheth, H. (22. Märts 2021. a.). *NUnit vs. XUnit vs. MSTest: Comparing Unit Testing Frameworks In C#*. Allikas: <https://www.lambdatest.com/blog/nunit-vs-xunit-vs-mstest/>
- Borenkraout, M. (2. Detsember 2020. a.). *React Testing Library* . Allikas: <https://testing-library.com/docs/react-testing-library/intro/>

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

Mina, Lisette Sirel

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

18.05.2021

---

<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 loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Nädal 1 (01.03.2021-07.03.2021)

<span>&lt;</span> 01 Mar - 07 Mar <span>&gt;</span>		TOTAL HOURS		
		19:07:19		
<span>⚙</span> Bulk edit		<span>↻</span> Rounding		
TIME ENTRY	USER	DURATION	TIME	
Raamistikud ● BST website (lõputöö)	Lisettesirel...	0 09:10	3:08 PM-3:17 PM 03/01/2021	
Azure DevOps ● BST website (lõputöö)	Lisettesirel...	0 14:22	3:39 PM-3:54 PM 03/01/2021	
Creating Azure Devops Wiki page ● BST website (lõputöö)	Lisettesirel...	0 18:45	4:22 PM-4:41 PM 03/01/2021	
Azure DevOps Boards + new project ● BST website (lõputöö)	Lisettesirel...	0 40:34	4:46 PM-5:26 PM 03/01/2021	
Connect VS project to Azure DevOps + try to deploy ● BST website (lõputöö)	Lisettesirel...	1 38:37	9:26 PM-11:04 PM 03/01/2021	
Edit Azure DevOps project + documentation ● BST website (lõputöö)	Lisettesirel...	0 14:18	8:43 AM-8:57 AM 03/02/2021	
Deploy WebApp to Azure ● BST website (lõputöö)	Lisettesirel...	0 20:59	9:56 AM-10:17 AM 03/02/2021	
WebApp deploy ● BST website (lõputöö)	Lisettesirel...	0 11:17	12:07 PM-12:18 PM 03/02/2021	
Test if the website will accept the changes made in VS ● BST website (lõputöö)	Lisettesirel...	2 39:22	12:31 PM-3:10 PM 03/02/2021	
Trying to push the changes to live website (NOT WORKING) ● BST website (lõputöö)	Lisettesirel...	1 10:00	10:45 AM-11:55 AM 03/03/2021	
Disable browser cache + website changes + publish ● BST website (lõputöö)	Lisettesirel...	0 10:39	9:42 AM-9:53 AM 03/04/2021	
Azure DevOps wiki page & board + investigate different options for registration ● BST website (lõputöö)	Lisettesirel...	0 49:38	9:56 AM-10:46 AM 03/04/2021	
Create Admin project for studio manager ● BST website (lõputöö)	Lisettesirel...	0 55:26	2:11 PM-3:06 PM 03/04/2021	
Continue with Admin project ● BST website (lõputöö)	Lisettesirel...	4 00:00	3:18 PM-7:18 PM 03/04/2021	
New admin project + ClientApp to VS Code ● BST website (lõputöö)	Lisettesirel...	1 25:07	9:18 AM-10:43 AM 03/05/2021	
AGAIN! EVERYTHING! ● BST website (lõputöö)	Lisettesirel...	0 36:47	10:55 AM-11:32 AM 03/05/2021	
Publish VS Code project to Azure Portal ● BST website (lõputöö)	Lisettesirel...	1 16:58	11:30 AM-12:47 PM 03/05/2021	
Create new WebApp in Azure Portal ● BST website (lõputöö)	Lisettesirel...	0 02:20	1:15 PM-1:17 PM 03/05/2021	
New WebApp & deploy ● BST website (lõputöö)	Lisettesirel...	1 13:00	1:32 PM-2:45 PM 03/05/2021	
Azure DevOps + Vs Code ● BST website (lõputöö)	Lisettesirel...	1 00:00	2:50 PM-3:50 PM 03/06/2021	

## Lisa 3 – Nädal 2 (08.03.2021-14.03.2021)

<span>&lt;</span> 08 Mar - 14 Mar <span>&gt;</span>		TOTAL HOURS		
		31:23:23		
<span>≡</span> Bulk edit		<span>↻</span> Rounding		
TIME ENTRY	USER	DURATION	TIME	
Deploy to Azure Portal <span>●</span> BST website (lõputöö)	Lisettesirel...	0:49:28	9:56 AM-10:45 AM 03/08/2021	
Deploy to Azure Portal + static website + new project + pipelines and so on <span>●</span> BST website (lõputöö)	Lisettesirel...	3:26:51	11:50 AM-3:17 PM 03/08/2021	
new project + deploy <span>●</span> BST website (lõputöö)	Lisettesirel...	1:29:00	4:01 PM-5:30 PM 03/08/2021	
frontend navbar <span>●</span> BST website (lõputöö)	Lisettesirel...	3:15:00	12:15 PM-3:30 PM 03/09/2021	
Deploy to Azure <span>●</span> BST website (lõputöö)	Lisettesirel...	3:14:59	10:14 PM-1:29 AM 03/09/2021	
static webpage - navbar <span>●</span> BST website (lõputöö)	Lisettesirel...	0:32:57	12:13 PM-12:46 PM 03/10/2021	
static webpage - navbar + link pages to navbar <span>●</span> BST website (lõputöö)	Lisettesirel...	3:32:29	1:29 PM-5:01 PM 03/10/2021	
static webpage - navbar + link pages to navbar <span>●</span> BST website (lõputöö)	Lisettesirel...	3:35:48	9:02 AM-12:38 PM 03/11/2021	
search home page ideas <span>●</span> BST website (lõputöö)	Lisettesirel...	0:26:26	1:22 PM-1:48 PM 03/11/2021	
team page ideas + card component <span>●</span> BST website (lõputöö)	Lisettesirel...	0:28:51	2:12 PM-2:41 PM 03/11/2021	
team page + card component <span>●</span> BST website (lõputöö)	Lisettesirel...	1:39:05	2:53 PM-4:32 PM 03/11/2021	
Deploy to Azure <span>●</span> BST website (lõputöö)	Lisettesirel...	0:11:38	4:59 PM-5:11 PM 03/11/2021	
search team page design options + team page <span>●</span> BST website (lõputöö)	Lisettesirel...	1:20:50	5:48 PM-7:09 PM 03/11/2021	
team page + popup window <span>●</span> BST website (lõputöö)	Lisettesirel...	1:57:12	10:56 AM-12:54 PM 03/13/2021	
popup window + team page design <span>●</span> BST website (lõputöö)	Lisettesirel...	1:26:49	1:40 PM-3:06 PM 03/13/2021	
team page, popup, info <span>●</span> BST website (lõputöö)	Lisettesirel...	3:56:00	3:09 PM-7:05 PM 03/13/2021	



## Lisa 4 – Nädal 3 (15.03.2021-21.03.2021)

15 Mar - 21 Mar		TOTAL HOURS	
		22:15:52	
Bulk edit		Rounding	
TIME ENTRY	USER	DURATION	TIME
schedule component ● BST website (lõputöö)	Lisettesirel...	1:49:24	11:27 AM-1:16 PM 03/15/2021
packages/pricing page & product card component ● BST website (lõputöö)	Lisettesirel...	4:44:00	1:45 PM-6:29 PM 03/15/2021
Website design ● BST website (lõputöö)	Lisettesirel...	0:47:29	4:38 PM-5:26 PM 03/16/2021
Website design ● BST website (lõputöö)	Lisettesirel...	0:26:45	5:41 PM-6:08 PM 03/16/2021
home design & sticky navbar & hidden schedule ● BST website (lõputöö)	Lisettesirel...	2:50:53	12:25 PM-3:16 PM 03/17/2021
about page info & design ● BST website (lõputöö)	Lisettesirel...	0:50:07	3:32 PM-4:22 PM 03/17/2021
website background & change picture resolution & team section design & packages section de ● BST website (lõpu	Lisettesirel...	3:24:00	5:06 PM-8:30 PM 03/17/2021
packages design & team popup design & about page + design ● BST website (lõputöö)	Lisettesirel...	1:52:41	8:49 PM-10:41 PM 03/17/2021
Website meeting ● BST website (lõputöö)	Lisettesirel...	0:55:00	12:00 PM-12:55 PM 03/18/2021
contact form & page ● BST website (lõputöö)	Lisettesirel...	0:53:02	8:29 PM-9:22 PM 03/18/2021
contact form & page ● BST website (lõputöö)	Lisettesirel...	1:09:31	9:32 PM-10:41 PM 03/18/2021
website meeting with manager ● BST website (lõputöö)	Lisettesirel...	2:33:00	3:00 PM-5:33 PM 03/19/2021

## Lisa 5 – Nädal 4 (22.03.2021-28.03.2021)

22 Mar - 28 Mar		TOTAL HOURS	
		24:46:20	
<a href="#">Bulk edit</a>		<a href="#">Rounding</a>	
TIME ENTRY	USER	DURATION	TIME
admin projekt ● BST website (lõputöö)	Lisettesirel...	3:36:00	2:15 PM-5:51 PM 03/22/2021
admin ● BST website (lõputöö)	Lisettesirel...	2:15:00	10:30 AM-12:45 PM 03/23/2021
projects connections ● BST website (lõputöö)	Lisettesirel...	3:15:00	3:15 PM-6:30 PM 03/23/2021
admin pages ● BST website (lõputöö)	Lisettesirel...	4:05:00	2:45 PM-6:50 PM 03/25/2021
deploy problem ● BST website (lõputöö)	Lisettesirel...	3:15:00	9:15 PM-12:30 AM 03/25/2021
FAQ page ● BST website (lõputöö)	Lisettesirel...	2:11:00	9:05 PM-11:16 PM 03/27/2021
Contact page ● BST website (lõputöö)	Lisettesirel...	3:25:00	12:27 PM-3:52 PM 03/28/2021
Contact page ● BST website (lõputöö)	Lisettesirel...	0:27:20	4:41 PM-5:08 PM 03/28/2021
Contact page ● BST website (lõputöö)	Lisettesirel...	2:17:00	5:26 PM-7:43 PM 03/28/2021


## Lisa 6 – Nädal 5 (29.03.2021-04.04.2021)

29 Mar - 04 Apr		TOTAL HOURS	
		21:02:51	
<a href="#">Bulk edit</a>		<a href="#">Rounding</a>	
TIME ENTRY	USER	DURATION	TIME
Contact page ● BST website (lõputöö)	Lisettesirel...	0:13:24	12:33 AM-12:46 AM 03/29/2021
Contact page ● BST website (lõputöö)	Lisettesirel...	2:31:15	2:22 PM-4:53 PM 03/29/2021
Admin project ● BST website (lõputöö)	Lisettesirel...	1:49:18	9:09 PM-10:58 PM 03/29/2021
Admin project ● BST website (lõputöö)	Lisettesirel...	1:56:59	10:17 AM-12:14 PM 03/30/2021
admin pages ● BST website (lõputöö)	Lisettesirel...	3:03:00	3:15 PM-6:18 PM 03/31/2021
database connection ● BST website (lõputöö)	Lisettesirel...	2:49:00	2:05 PM-4:54 PM 04/01/2021
errors ● BST website (lõputöö)	Lisettesirel...	2:15:00	5:30 PM-7:45 PM 04/01/2021
Admin project ● BST website (lõputöö)	Lisettesirel...	3:09:55	9:59 AM-1:09 PM 04/03/2021
admin ● BST website (lõputöö)	Lisettesirel...	3:15:00	3:15 PM-6:30 PM 04/03/2021

## Lisa 7 – Nädal 6 (05.04.2021-11.04.2021)

05 Apr - 11 Apr		TOTAL HOURS	
		15:09:22	
<a href="#">Bulk edit</a>		<a href="#">Rounding</a>	
TIME ENTRY	USER	DURATION	TIME
Deploy Admin + database Mac + backend project ● BST website (lõputöö)	Lisettesirel...	3:07:29	1:37 PM-4:44 PM 04/05/2021
f git ● BST website (lõputöö)	Lisettesirel...	1:22:34	6:31 PM-7:53 PM 04/05/2021
git problems ● BST website (lõputöö)	Lisettesirel...	1:25:00	3:30 PM-4:55 PM 04/06/2021
backend, database connections, backend models ● BST website (lõputöö)	Lisettesirel...	1:11:56	9:02 AM-10:14 AM 04/11/2021
Data from DB (about, instructors) ● BST website (lõputöö)	Lisettesirel...	4:33:15	11:04 AM-3:38 PM 04/11/2021
pics + faq DB ● BST website (lõputöö)	Lisettesirel...	1:24:29	3:46 PM-5:11 PM 04/11/2021
faq + db + fcking errorid ● BST website (lõputöö)	Lisettesirel...	2:04:39	9:03 PM-11:08 PM 04/11/2021

## Lisa 8 – Nädal 7 (12.04.2021-18.04.2021)

<span>&lt;</span> 12 Apr - 18 Apr <span>&gt;</span>		TOTAL HOURS		
		21:04:44		
<span>≡</span> Bulk edit		<span>↻</span> Rounding		
TIME ENTRY	USER	DURATION	TIME	
azure functions vs other solution ● BST website (lõputöö)	Lisettesirel...	2:35:00	6:30 PM-9:05 PM 04/12/2021	
deployment ● BST website (lõputöö)	Lisettesirel...	1:11:00	5:00 PM-6:11 PM 04/13/2021	
Azure static webapp + api ● BST website (lõputöö)	Lisettesirel...	1:54:19	3:53 PM-5:48 PM 04/14/2021	
azure web apps + api connection ● BST website (lõputöö)	Lisettesirel...	2:48:14	6:08 PM-8:56 PM 04/14/2021	
azure functions + api ● BST website (lõputöö)	Lisettesirel...	1:35:42	12:02 PM-1:37 PM 04/15/2021	
test project with azure functions ● BST website (lõputöö)	Lisettesirel...	1:14:51	3:51 PM-5:06 PM 04/15/2021	
function application vs ● BST website (lõputöö)	Lisettesirel...	0:44:04	5:38 PM-6:22 PM 04/15/2021	
azure functions -API ● BST website (lõputöö)	Lisettesirel...	1:12:31	12:01 PM-1:14 PM 04/16/2021	
azure + api deployment ● BST website (lõputöö)	Lisettesirel...	1:19:51	9:30 AM-10:49 AM 04/18/2021	
api deployment ● BST website (lõputöö)	Lisettesirel...	1:39:56	11:07 AM-12:47 PM 04/18/2021	
azure + api ● BST website (lõputöö)	Lisettesirel...	0:29:31	1:01 PM-1:31 PM 04/18/2021	
deployment error ● BST website (lõputöö)	Lisettesirel...	4:01:23	6:25 PM-10:27 PM 04/18/2021	

## Lisa 9 – Nädal 8 (19.04.2021-25.04.2021)

<span>&lt;</span> <span>📅</span> 19 Apr - 25 Apr <span>&gt;</span>		TOTAL HOURS		
		23:54:09		
<span>⚙️</span> Bulk edit		<span>🔄</span> Rounding		
TIME ENTRY	USER	DURATION	TIME	
about & team page design update + packages code update + faq page & DB ● BST website (lõputöö)	Lisettesirel...	4:55:56	4:02 PM-8:58 PM 04/19/2021	
faq page (new) ● BST website (lõputöö)	Lisettesirel...	1:01:39	10:10 PM-11:12 PM 04/19/2021	
faq page (same) + google docs file ● BST website (lõputöö)	Lisettesirel...	0:23:20	11:21 PM-11:44 PM 04/19/2021	
gunnarile meil + dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	3:52:54	10:23 AM-2:16 PM 04/20/2021	
dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	0:50:34	3:51 PM-4:41 PM 04/20/2021	
dokumentatsioon + registration system ● BST website (lõputöö)	Lisettesirel...	0:47:42	5:00 PM-5:47 PM 04/20/2021	
gunnarile küsimused + dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	1:26:22	9:13 AM-10:39 AM 04/22/2021	
registration form ● BST website (lõputöö)	Lisettesirel...	0:42:08	11:18 AM-12:01 PM 04/22/2021	
Gunnar - meeting ● BST website (lõputöö)	Lisettesirel...	0:23:18	1:00 PM-1:24 PM 04/22/2021	
registration form ● BST website (lõputöö)	Lisettesirel...	2:02:00	4:30 PM-6:32 PM 04/22/2021	
dokumentatsioon + admin login ● BST website (lõputöö)	Lisettesirel...	3:03:20	10:30 AM-1:34 PM 04/23/2021	
admin project ● BST website (lõputöö)	Lisettesirel...	0:24:16	10:10 AM-10:34 AM 04/24/2021	
admin navbar ● BST website (lõputöö)	Lisettesirel...	0:30:32	1:06 PM-1:37 PM 04/24/2021	
admin project ● BST website (lõputöö)	Lisettesirel...	0:28:18	4:25 PM-4:53 PM 04/24/2021	
admin navbar + dashboard ● BST website (lõputöö)	Lisettesirel...	1:03:14	6:50 PM-7:53 PM 04/24/2021	
admin login ● BST website (lõputöö)	Lisettesirel...	1:58:36	12:15 PM-2:13 PM 04/25/2021	

# Lisa 10 – Nädal 9 (26.04.2021-02.05.2021)

<span>&lt;</span> 26 Apr - 02 May <span>&gt;</span>		TOTAL HOURS		
		35:00:00		
Bulk edit		Rounding		
TIME ENTRY	USER	DURATION	TIME	
admin project ● BST website (lõputöö)	Lisettesirel...	0:22:53	3:08 PM-3:31 PM 04/26/2021	
admin project - fetch data from api (GET method) ● BST website (lõputöö)	Lisettesirel...	3:26:00	3:31 PM-6:57 PM 04/26/2021	
admin project ● BST website (lõputöö)	Lisettesirel...	1:19:22	9:42 PM-11:01 PM 04/26/2021	
dokumentatsioon + admin project (create button & edit button) + db connection ● BST website (lõputöö)	Lisettesirel...	5:36:22	11:30 AM-5:06 PM 04/27/2021	
admin project ● BST website (lõputöö)	Lisettesirel...	3:04:25	2:39 PM-5:43 PM 04/28/2021	
admin page ● BST website (lõputöö)	Lisettesirel...	2:02:57	5:50 PM-7:53 PM 04/28/2021	
crud operations admin project + about page design + customers page design + dashboard ● BST website (lõputöö)	Lisettesirel...	3:28:51	7:59 PM-11:28 PM 04/28/2021	
team page + DB connection (admin) ● BST website (lõputöö)	Lisettesirel...	1:32:31	11:37 AM-1:09 PM 04/29/2021	
packages + DB connection (admin project) + FAQ page ● BST website (lõputöö)	Lisettesirel...	2:57:53	1:46 PM-4:44 PM 04/29/2021	
faq page ● BST website (lõputöö)	Lisettesirel...	1:46:23	5:24 PM-7:10 PM 04/29/2021	
faq page error (not showing content) ● BST website (lõputöö)	Lisettesirel...	0:40:55	8:59 PM-9:40 PM 04/29/2021	
FAQ page error done ● BST website (lõputöö)	Lisettesirel...	0:24:41	10:19 PM-10:44 PM 04/29/2021	
admin FAQ edit page ● BST website (lõputöö)	Lisettesirel...	0:50:25	11:40 PM-12:30 AM 04/29/2021	
lõputöö ● BST website (lõputöö)	Lisettesirel...	2:29:00	9:55 AM-12:24 PM 05/01/2021	
lõputöö ● BST website (lõputöö)	Lisettesirel...	3:03:00	2:44 PM-5:47 PM 05/01/2021	
publish backend, POST request error (packages & faq admin) ● BST website (lõputöö)	Lisettesirel...	1:08:14	1:01 PM-2:09 PM 05/02/2021	
dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	0:46:08	9:57 PM-10:43 PM 05/02/2021	

# Lisa 11 – Nädal 10 (03.05.2021-09.05.2021)

03 May - 09 May		TOTAL HOURS		
		46:36:24		
Bulk edit		Rounding		
TIME ENTRY	USER	DURATION	TIME	
dokumentatsioon & maksmise funktsionaalsus & home page • BST website (lõputöö)	Lisettesirel...	6:03:54	1:13 PM-7:17 PM 05/03/2021	
courses page + DB connection, dropdown menu for studio info, dropdown "studio" link 404 error • BST website (lõputöö)	Lisettesirel...	3:30:16	10:27 PM-1:57 AM 05/03/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	4:19:54	1:30 PM-5:50 PM 05/04/2021	
dokumentatsioon - sissehjuhatuse • BST website (lõputöö)	Lisettesirel...	0:07:28	6:29 PM-6:36 PM 05/04/2021	
dokumentatsioon - viitamine & metoodika, töö tulemused • BST website (lõputöö)	Lisettesirel...	3:52:00	8:07 AM-11:59 AM 05/05/2021	
dokumentatsioon - töö tulemused • BST website (lõputöö)	Lisettesirel...	0:59:47	3:26 PM-4:25 PM 05/05/2021	
dokumentatsioon - arhitektuur (diagrammid) • BST website (lõputöö)	Lisettesirel...	2:14:39	5:24 PM-7:38 PM 05/05/2021	
dokumentatsioon - arhitektuur (diagrammid) • BST website (lõputöö)	Lisettesirel...	2:18:41	8:14 PM-10:33 PM 05/05/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	1:18:18	2:00 PM-3:18 PM 05/06/2021	
dokumentatsioon + events page • BST website (lõputöö)	Lisettesirel...	2:48:27	4:51 PM-7:39 PM 05/06/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	1:01:08	10:21 AM-11:23 AM 05/07/2021	
dokumentatsioon + events table in DB + events page • BST website (lõputöö)	Lisettesirel...	3:55:54	12:04 PM-4:00 PM 05/07/2021	
dokumentatsioon + courses page update • BST website (lõputöö)	Lisettesirel...	1:41:09	4:56 PM-6:38 PM 05/07/2021	
dokumentatsioon - kood • BST website (lõputöö)	Lisettesirel...	2:54:16	1:45 PM-4:39 PM 05/08/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	0:44:30	8:12 PM-8:57 PM 05/08/2021	
dokumentatsioon - analüüs + makse võimalused veebilehel • BST website (lõputöö)	Lisettesirel...	2:47:00	10:07 PM-12:54 AM 05/08/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	0:36:08	9:35 AM-10:12 AM 05/09/2021	
dokumentatsioon • BST website (lõputöö)	Lisettesirel...	0:34:05	12:41 PM-1:15 PM 05/09/2021	
lõputöö, avaldus, meilid • BST website (lõputöö)	Lisettesirel...	0:41:32	7:15 PM-7:57 PM 05/09/2021	
maksevõimalused + dokumentatsioon (tööriistade põhjendus: wordpress vs .NET Core) + kood • BST website (lõputöö)	Lisettesirel...	4:07:18	11:16 PM-3:23 AM 05/09/2021	



## Lisa 12 – Nädal 11 (10.05.2021-16.05.2021)

<span>&lt;</span> <span>Last week</span> <span>&gt;</span>		TOTAL HOURS		
		43:41:39		
<span>≡</span> Bulk edit		<span>↻</span> Rounding		
TIME ENTRY	USER	DURATION	TIME	
dokumentatsioon - nõuded, analüüs, backendi arhitektuur ja klassidiagramm ● BST website (lõputöö)	Lisettesirel...	3:02:37	10:13 AM-1:15 PM 05/10/2021	
maksevõimalus + dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	0:21:22	5:34 PM-5:55 PM 05/10/2021	
maksmise doc + dokumentatsioon (arhitektuur, disaini analüüs, ekraanitõmmised) ● BST website (lõputöö)	Lisettesirel...	5:12:14	9:06 PM-2:18 AM 05/10/2021	
dokumentatsioon - arhitektuur, kood, testid ● BST website (lõputöö)	Lisettesirel...	5:27:49	8:38 AM-2:06 PM 05/11/2021	
dokumentatsioon - koodi analüüs, juurutamise protsess ● BST website (lõputöö)	Lisettesirel...	3:16:00	3:24 PM-6:40 PM 05/11/2021	
admin events page ● BST website (lõputöö)	Lisettesirel...	0:33:24	12:27 PM-1:01 PM 05/12/2021	
error ● BST website (lõputöö)	Lisettesirel...	0:15:31	1:58 PM-2:13 PM 05/12/2021	
payment + 404 error ● BST website (lõputöö)	Lisettesirel...	4:05:41	9:02 PM-1:07 AM 05/12/2021	
testimine + admin events page and courses page ● BST website (lõputöö)	Lisettesirel...	1:53:12	4:20 PM-6:13 PM 05/13/2021	
DB price to navbar + dropdown navbar + schedule page + home page ● BST website (lõputöö)	Lisettesirel...	6:38:03	7:20 PM-1:58 AM 05/13/2021	
packages page update + payment links + team popup ● BST website (lõputöö)	Lisettesirel...	2:32:56	10:51 AM-1:24 PM 05/14/2021	
dokumentatsioon - makseportaal, töötlemused, remove dashboard, add screenshots ● BST website (lõputöö)	Lisettesirel...	2:03:00	2:55 PM-4:58 PM 05/14/2021	
dokumentatsioon ● BST website (lõputöö)	Lisettesirel...	1:24:40	9:20 AM-10:45 AM 05/16/2021	
dokumentatsioon & maksmine, packages page & schedule page & contact page, db diagram ● BST website (lõputöö)	Lisettesirel...	4:50:45	2:54 PM-7:44 PM 05/16/2021	
dokumentatsioon - DB & diagram, analüüs ja järeldused ● BST website (lõputöö)	Lisettesirel...	2:04:25	10:15 PM-12:20 AM 05/16/2021	

## Lisa 13 – Nädal 12 (17.05.2021-23.05.2021)

TOTAL HOURS			
11:54:15			
<a href="#">Bulk edit</a>			
<a href="#">Rounding</a>			
TIME ENTRY	USER	DURATION	TIME
testimine NUnit + dokumentatsioon • BST website (lõputöö)	Lisettesirel...	0:50:10	10:25 AM-11:15 AM 05/17/2021
API testimine + dokumentatsioon: analüüs • BST website (lõputöö)	Lisettesirel...	7:31:04	7:38 PM-3:09 AM 05/17/2021
metaandmed, lihtlitsents, dokumentatsioon - logid • BST website (lõputöö)	Lisettesirel...	2:23:20	2:30 PM-4:54 PM 05/18/2021
dokumentatsioon - läbivaatus • BST website (lõputöö)	Lisettesirel...	0:28:39	5:04 PM-5:32 PM 05/18/2021
dokumentatsioon - läbivaatus • BST website (lõputöö)	Lisettesirel...	0:41:02	9:42 PM-10:23 PM 05/18/2021