

**TALLINNA TEHNIKAÜLIKOOL**

INSENERITEADUSKOND

Virumaa kolledž

**Eriarsti vastuvõtuaja broneerimise süsteem**

RAM0541 Veebiprogrammeerimine

Üliõpilane: Mark Britševski, Ksenia Adel

Üliõpilaskood: ..., 231789EDTR

Õppejõud: Natalja Ivleva

**Sisukord**

[1 Veebirakenduse arenduse plaan 3](#_Toc196529192)

[1.1 Projekti kirjeldus 3](#_Toc196529193)

[1.2 Eesmärk 3](#_Toc196529194)

[1.3 Arhitektuur ja tehnoloogiad 3](#_Toc196529195)

[1.4 Funktsionaalsed nõuded 3](#_Toc196529196)

[1.5 Andmebaasi skeem (Sequelize - PostgreSQL) 4](#_Toc196529197)

[2 II etapp 5](#_Toc196529198)

[2.1 API Dokumentatsioon 5](#_Toc196529199)

# Veebirakenduse arenduse plaan

## Projekti kirjeldus

Eriarsti vastuvõtuaja broneerimise süsteem on veebirakendus, mis võimaldab patsientidel kiiresti ja mugavalt broneerida arstivastuvõtte. Arstid saavad hallata oma ajakava ja teenuseid, samal ajal kui administraatorid saavad hallata kasutajaid, ajakavasid ja teenuste kataloogi. Süsteem tagab turvalise autentimise ja hõlbustab broneeringute haldamist.

## Eesmärk

Projekti eesmärk on luua kasutajasõbralik, tõhus ja turvaline infosüsteem, mis võimaldab:

* Patsientidel mugavalt leida ja broneerida arstiaegu.
* Arstidel hallata oma ajakava ja pakutavaid teenuseid.
* Administraatoritel hallata kasutajaid, teenuseid ja broneeringuid.
* Minimeerida telefonibroneeringute hulka ja parandada teenuse kättesaadavust.

## Arhitektuur ja tehnoloogiad

**Backend**

FrameWork: Node.js + Express

Database ORM: Sequelize (PostgreSQL)

Autentimine: JWT + bcrypt

API: REST (Express)

**Frontend**

FrameWork: React (Vite)

Olekuhaldus: Redux, Context API

API: Fetch / Axios

## **Funktsionaalsed nõuded**

Autentimine ja autoriseerimine

* Kasutajate registreerimine, sisselogimine ja väljalogimine
* Rollipõhine autoriseerimine (administraator, arst, patsient)
* JWT-põhine autentimine ja sessioonihaldus

Andmebaasi haldus (ORM - Sequelize)

* Kasutajaandmete salvestamine (administraatorid, arstid, patsiendid)
* Teenuste kataloogi ja arstide ajakavade haldamine
* Broneeringute loomine, muutmine ja kustutamine
* Statistika genereerimine ja haldamine

**API CRUD-toimingute jaoks**

Administraator:

1. Lisab ja haldab arste
2. Haldab teenuste kataloogi
3. Genereerib statistikat

Arst:

1. Lisab ja haldab oma ajakava
2. Näeb ja broneeringuid
3. Lisab oma profiilile pakutavaid teenuseid

Patsient:

1. Otsib arsti ja teenust
2. Broneerib ja tühistab aegu
3. Maksab broneeringu eest
4. Vaatab oma broneeringud

**Reaktiivne UI ja kasutajakogemus**

* Dünaamilised komponendid (React + Vite)
* Oleku haldus (Redux, Context API)
* Kasutajaliides (CSS + Tailwind)

**Vigade käsitlemine ja vormide valideerimine**

* Kasutajasõbralikud veateated (frontend + backend)
* Vormide valideerimine (React Hook Form, Yup)
* Serveripoolsed veakoodid ja logimine

## Andmebaasi skeem (Sequelize - PostgreSQL)

Tabelid ja seosed

admin(adminid, firstname, lastname, email, phone, userid)

doctor(doctorid, firstname, lastname, specialty, email, phone, adminid, userid)

patient(patientid, firstname, lastname, personalcode, email, phone, address, userid)

servicecatalog(servicecatalogid, name, description, duration, price, adminid)

doctorservice(doctorserviceid, doctorid, servicecatalogid)

schedule(scheduleid, date, starttime, endtime, doctorid)

booking(bookingid, patientid, servicecatalogid, scheduleid, ispaid)

statistic(statisticid, reporttype, generateddate, adminid, reportcontent)

user(userid, email, password, role)

# ERD

# II etapp

## API Dokumentatsioon

**ADMIN:**

Login - POST /api/auth/login

Find/add/update/delete **admin**:

GET /api/admins

POST /api/admins

PUT /api/admins/:adminid

DELETE /api/admins/:adminid

Find/add/update/delete **doctor**:

GET /api/doctors

POST /api/doctors

PUT /api/doctors/:doctorid

DELETE /api/doctors/:doctorid

Find/update/delete **patients**:

GET api/patients

PUT api/patients/:patientid

DELETE api/patients/:patientid

Find/add/update/delete **services**:

GET /api/services

POST /api/services

PUT /api/services/:id

DELETE /api/services/:id

Generate/delete **statistic**:

GET /api/admins/stats

DELETE api/admins/stats/:statisticid

**PATIENT:**

Register/ login - POST /api/auth/register и POST /api/auth/login

Login - POST /api/auth/login

Find doctors - GET /api/doctors

Find doctor services – GET /api/doctorservices/:doctorid

Find doctor schedule – GET /api/schedules/:doctorid

Make booking - POST /api/bookings

Look at own bookings – GET /api/bookings

Delete booking – DELETE /api/bookings/:id

Pay booking – PUT api/bookings/:bookingid/pay

**DOCTOR:**

Create/change/delete schedule

GET /api/schedules

POST /api/schedules

DELETE /api/schedules/:scheduleid

Find/add/remove service

GET /api/doctorservices

POST /api/doctorservices

DELETE /api/doctorservices/:doctorserviceid

Find own bookings

GET /api/doctors/bookings

Lisame juba mõned andmed andmebaasi.

admin1@example.com admin1

doctor1@example.com doctor1

patient1@example.com patient1

patient2@example.com patient2

SET search\_path TO onlinedoc;

-- USERS

INSERT INTO "user" (userid, email, password, role) VALUES

(1, 'admin1@example.com', '$2b$10$0CrGHX.n.xT0DuvsNCck9ulzH6Cpt3FL0MYgX3TuZkRRWQTr.6lL6', 'admin'), -- admin123

(2, 'doctor1@example.com', '$2b$10$q8UftNi7aL/y.Yj2DFcFMu3RQlySMuULaeKnEpSnQo/RTxdTfRDRG', 'doctor'), -- doctor123

(3, 'patient1@example.com','$2b$10$rq/LYVxGgniEK/hqG6RihOcUlbW6VL1XDwhElZOkS23TQfONnmjZ2', 'patient'), -- patient123

(4, 'patient2@example.com','$2b$10$K5vHX7XrLPXYA9sn4O7zqOVuT/G0pbdbFX07NABNLv0A0nCSDBR3u', 'patient'); -- patient456

-- ADMIN

INSERT INTO admin (adminid, firstname, lastname, email, phone, userid) VALUES

(1, 'Anna', 'Adminova', 'admin1@example.com', '+37251234567', 1);

-- DOCTOR

INSERT INTO doctor (doctorid, firstname, lastname, specialty, email, phone, adminid, userid) VALUES

(1, 'James', 'Smith', 'Cardiologist', 'doctor1@example.com', '+37251211111', 1, 2);

-- PATIENTS

INSERT INTO patient (patientid, firstname, lastname, personalcode, email, phone, address, userid) VALUES

(1, 'Maria', 'Petrov', '49002012345', 'patient1@example.com', '+37253331111', 'Tallinn, Estonia', 3),

(2, 'Ivan', 'Ivanov', '50001019999', 'patient2@example.com', '+37256665555', 'Tartu, Estonia', 4);

-- SERVICE CATALOG

INSERT INTO servicecatalog (servicecatalogid, name, description, duration, price, adminid) VALUES

(1, 'Cardiologist visit', 'Heart & blood pressure check', 30, 60, 1),

(2, 'Blood test', 'Basic blood test', 15, 25, 1);

-- DOCTOR SERVICES

INSERT INTO doctorservice (doctorserviceid, doctorid, servicecatalogid) VALUES

(1, 1, 1),

(2, 1, 2);

-- SCHEDULE

INSERT INTO schedule (scheduleid, date, starttime, endtime, doctorid) VALUES

(1, '2025-05-02', '10:00', '10:30', 1),

(2, '2025-05-02', '11:00', '11:15', 1),

(3, '2025-05-03', '09:00', '09:30', 1);

-- BOOKINGS

INSERT INTO booking (bookingid, patientid, servicecatalogid, scheduleid, ispaid) VALUES

(1, 1, 1, 1, true),

(2, 2, 2, 2, false);

**ADMIN LOGIN**

POST /api/auth/login

{

"email": "admin1@example.com",

"password": "admin123"

}

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjEsInJvbGUiOiJhZG1pbiIsImlhdCI6MTc0NTU5NzE0MSwiZXhwIjoxNzQ1NjAwNzQxfQ.dlaCbhPX2waJd7GvFr7qrISnc8\_zrWX5uhP\_8fAPAUY",

"user": {

"userid": 1,

"email": "admin1@example.com",

"role": "admin"

}

}

**ADD/UPDATE/DELETE ADMIN**

SELECT \* FROM onlinedoc.admin

GET /api/admins

POST /api/admins

{

"firstName": "Admin2",

"lastName": "Admin",

"email": "admin2@example.com",

"password": "admin2",

"role": "admin"

}

PUT /api/admins/:id

{

"firstName": "Admin4",

"lastName": "Admin",

"email": "admin2@example.com",

"password": "admin2",

"role": "admin"

}

DELETE /api/admins/:id

**ADD/UPDATE/DELETE DOCTOR**

SELECT \* FROM onlinedoc.doctor

GET /api/doctors

POST /api/doctors

{

"firstname": "Anna",

"lastname": "Mets",

"email": "annamets@example.com",

"phone": "+37251234567",

"specialty": "Hambaarst"

}

PUT /api/doctors/:id

DELETE /api/doctors/:id

DOCTOR LOGIN

{

"email": "mark1@example.com",

"password": "uUcCNYtgFn"

}

**REGISTER/LOGIN PATIENT**

POST /api/auth/register

{

"firstname": "Anna",

"lastname": "Petrova",

"personalcode": "49412310011",

"email": "anna@example.com",

"phone": "+37251234567",

"address": "Tallinn, Estonia",

"password": "anna1"

}

POST /api/auth/login

{

"email": "anna@example.com",

"password": "anna1"

}

**SERVICECATALOG**

GET /api/services

POST /api/services

{

"name": "Hambaravi",

"description": "Täielik hambaarsti konsultatsioon ja ravi",

"duration": 45,

"price": 75

}

PUT api/services/1

{

"name": "Hambaravi",

"description": "Täielik hambaarsti konsultatsioon ja ravi",

"duration": 55,

"price": 70

}

DELETE api/services/1

**DOCTORSERVICE**

GET /api/doctorservices

POST /api/doctorservices

{

"servicecatalogid": 1 - servicecatalogid

}

DELETE /api/doctorservices/:doctorservice**id**

**SCHEDULE**

GET /api/schedule (iga arst näeb ainult oma ajakava)

POST /api/schedule

{

"date": "2025-04-27",

"starttime": "10:00",

"endtime": "11:00"

}

DELETE /api/schedule/:scheduleid

**BOOKING**

POST /api/booking

{

"servicecatalogid": 2,

"scheduleid": 2

}

DELETE /api/booking/:booking**id**