Eesrakenduste arendamine kodutöö 1 - Vestlus chatGPT-ga

Autor: Katriin Liselle Eenmaa

Tooksin kohe alguses välja, et kuna minu täielik vestlus chatGPT-ga on tohutult pikk, siis kogu vestlust ei ole mul võimalik siin välja tuua. Tõin kõik olulisemad osad meie vestlusest kuvatõmmistena siin failis välja ning täiendasin neid ka ise teksti juurde kirjutades.

Kui alguses vestlust chatGPT-ga alustasin siis muutsin mitu korda veel enda koodi soovi, mida talle esitasin, sest avastasin jooksvalt mitu puudust ning mitu kohta, mis vajasid täiendamist. Näiteks alguses genereeris chatGPT mulle koodi kasutamata eventListener’e, kuigi palusin seda teha. Siis tuletasin seda chatGPT-le meelde ja ta ka kohe parandas ennast. Samuti oli alguses elektrooniline kell vales ajatsoonis ning seetõttu pidin oma soovis välja tooma ka ajatsooni, mida kasutada. Pidin ka enda algset soovi täiendama sellega, et soovin nädalapäeva näha eesti keeles ja millises formaadis kuupäev peaks olema.

Kui olin oma enda koodi soovi ilusti läbi mõelnud ja kõik vajalikud omadused kirja saanud siis andsin chatGPT-le järgmise käsu:

**generate me a javascript code using eventListener and functions for a full screen electronical clock, that also shows todays date and weekday, using GMT +2 timezone that could be used as a screen saver. Show the weekday information in estonian language and for the date use the dd.mm.yyyy format**

ChatGPT genereeris mulle selle peale järgneva koodi:

function updateTime() {

const now = new Date();

const options = {

hour12: false,

timeZone: 'Europe/Tallinn',

weekday: 'long',

year: 'numeric',

month: '2-digit',

day: '2-digit',

};

const time = now.toLocaleTimeString('en-US', { hour12: false, timeZone: 'Europe/Tallinn' });

const date = now.toLocaleDateString('et-EE', options).replace(/\./g, '/');

document.getElementById('time').textContent = time;

document.getElementById('date').textContent = date;

}

function startClock() {

updateTime();

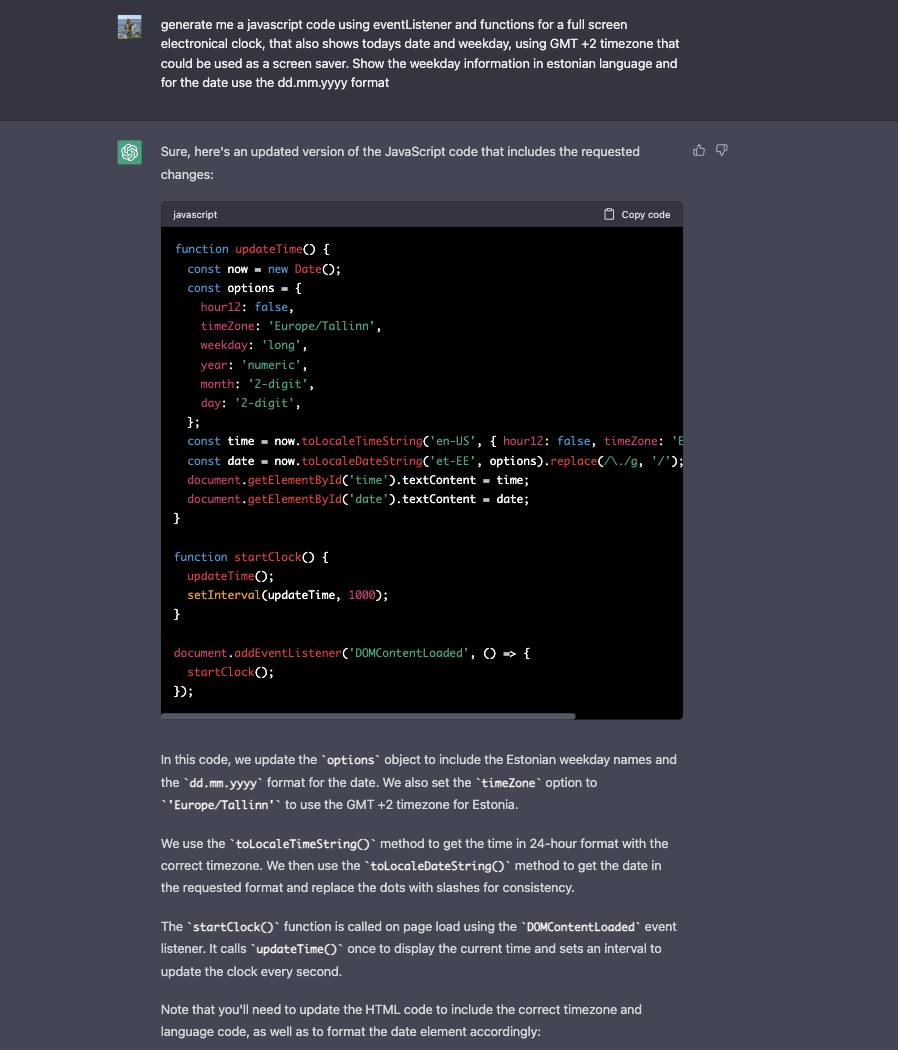
setInterval(updateTime, 1000);

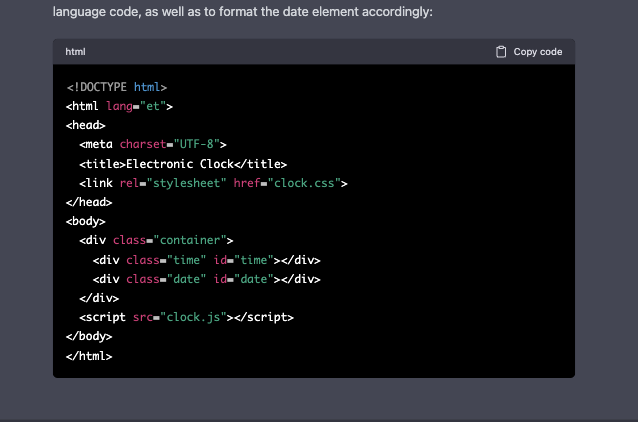
}

document.addEventListener('DOMContentLoaded', () => {

startClock();

});





Kuna olin juba enne seda genereerinud HTML koodi elektroonilise kella jaoks valmis siis tuletas chatGPT mulle ka meelde, et täiendaksin HTML koodis keelt. Alguses oli keel en ja siis panin selle asemele et.

Lasin chatGPT-l genereerida ka HTML koodi. Kuna genereerisin antud HTML koodi enne seda kui olin palunud chatGPT-l javascript koodi täiendada selliselt, et nädalapäevi kuvaks eesti keeles siis siin versioonis on keel veel en. Kuid kui palusin chatGPT-l nädalapäevi kuvada eesti keeles siis tuletas ta mulle ka meelde, et määraksin keeleks et.

Andsin chatGPT-le järgneva käsu: **also generate me html code for this clock**

ChatGPT genereeris mulle selle peale järgneva koodi:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Electronic Clock</title>

<link rel="stylesheet" href="clock.css">

</head>

<body>

<div class="container">

<div class="time" id="time"></div>

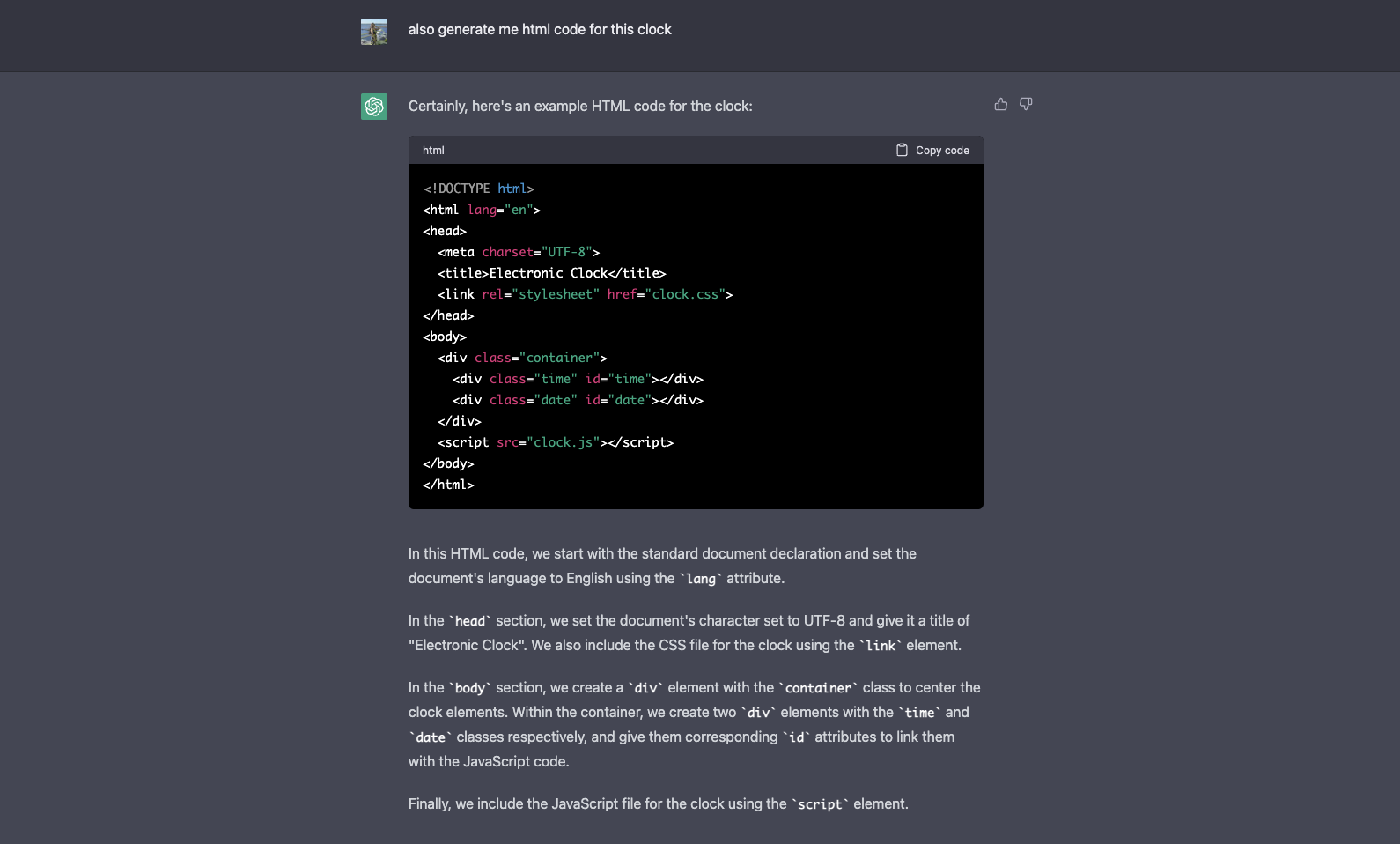
<div class="date" id="date"></div>

</div>

<script src="clock.js"></script>

</body>

</html>



Palusin chatGPT-le ka genereerida algse näidis css koodi. Andsin chatGPT-le järgneva käsu: **also generate me a css file for this clock**

ChatGPT genereeris mulle selle peale järgneva koodi:

body {

margin: 0;

padding: 0;

font-family: sans-serif;

background-color: black;

}

.container {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

height: 100vh;

}

.time {

font-size: 10em;

color: white;

margin-bottom: 0.2em;

}

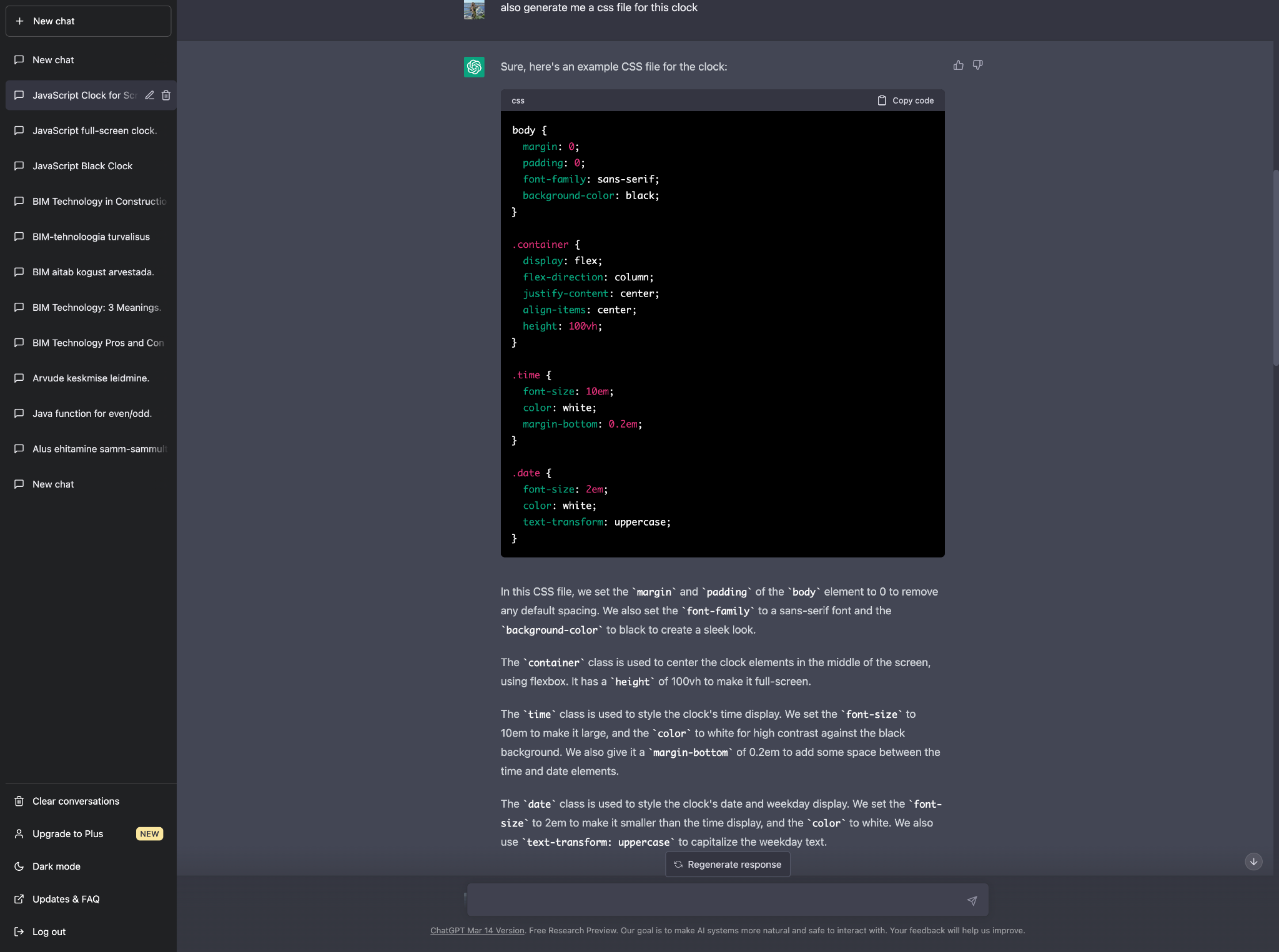
.date {

font-size: 2em;

color: white;

text-transform: uppercase;

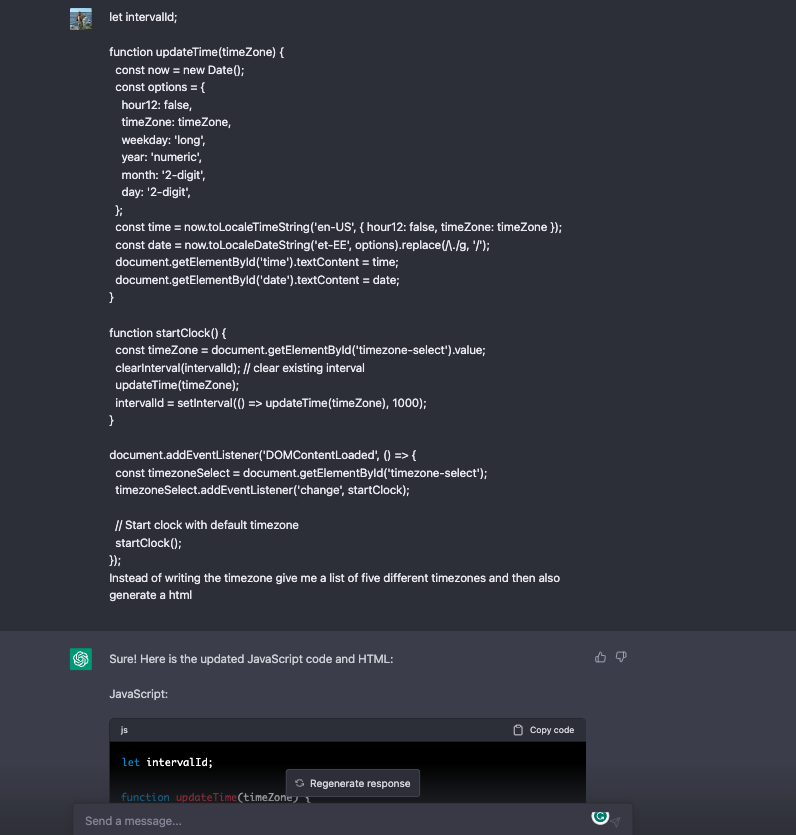
}

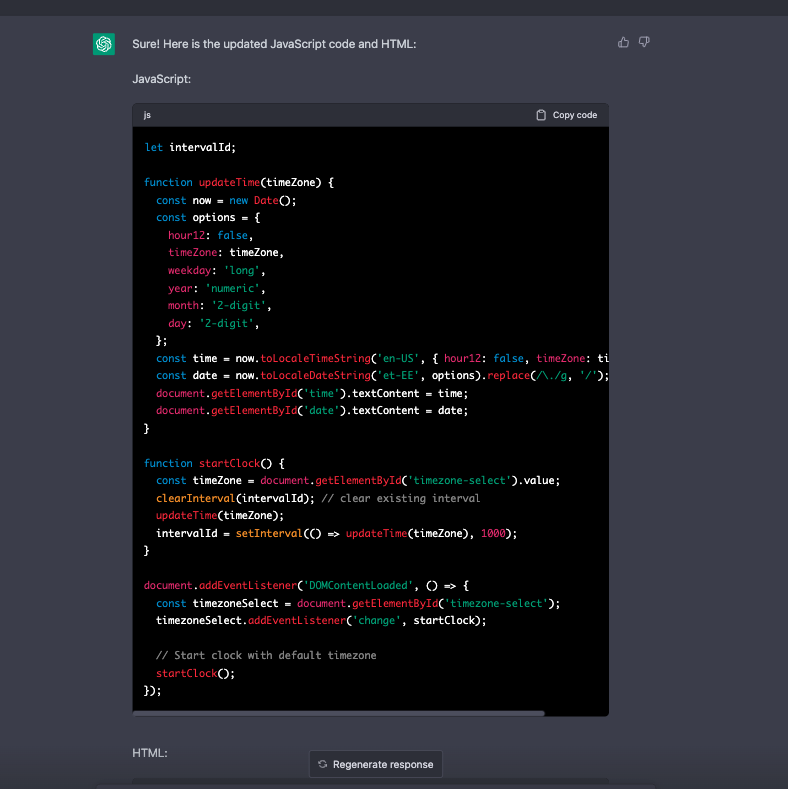


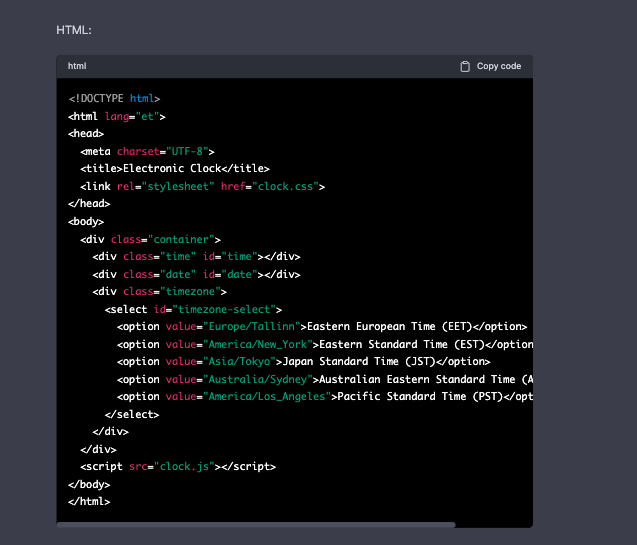
Nüüd oli mul valmis töötav elektrooniline kell, mis lisaks kellaajale näitab ka nädalapäeva ja kuupäeva koos aastaga. Samuti oli valmis algne kujundus. Nüüd oli aeg välja mõelda, milliseid funktsioone kellale juurde soovin.

Esimesena soovisin elektroonilisele kellale lisada funktsioonaalsuse, mis võimaldaks näidata erinevate ajavööndite kellaaega ja kuupäeva.

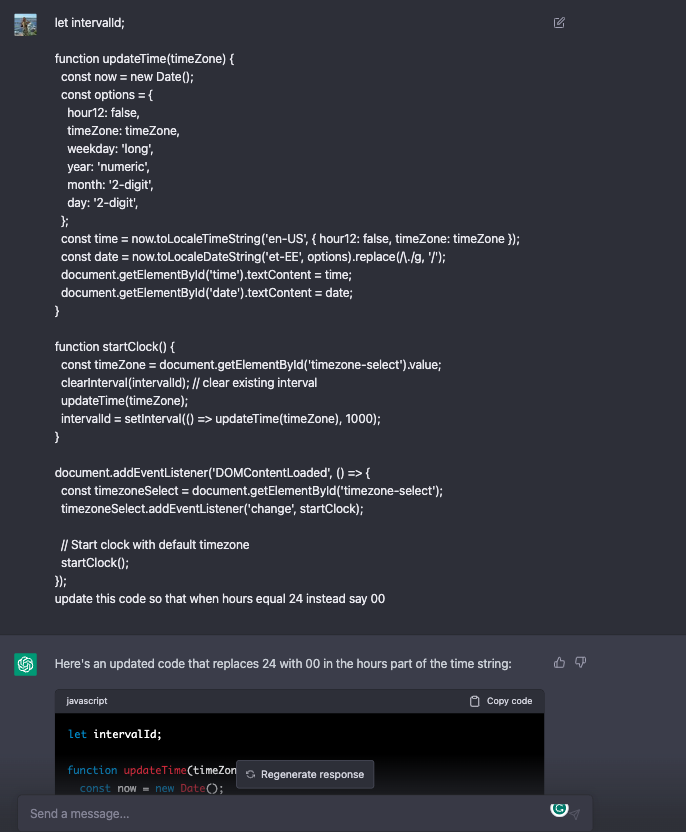
Esialgse käsu järgi genereeris chatGPT mulle sellise koodi, mis palus soovitud ajavööndi lahtrisse kirjutada, kuid palusin chatGPT-l selle asemel hoopis anda valiku ajavöönditest ja vastavalt ka HTML koodi uuendada. Selle peale genereeris chatGPT kolme valikuvõimalusega rippmenüü, kuid kuna soovisin viite ajavööndi valikut siis pidin veel oma soovi täiendama. Kuid siis enam ei klappinud HTML ja javascript, kuigi palusin ka HTML-i uuendada. Siis sain aru, et kõige parem on teha nii, et kopeerin oma praeguse koodi talle ja palun seda täiendada vastavalt ja samal ajal ka HTML-i uuendada.

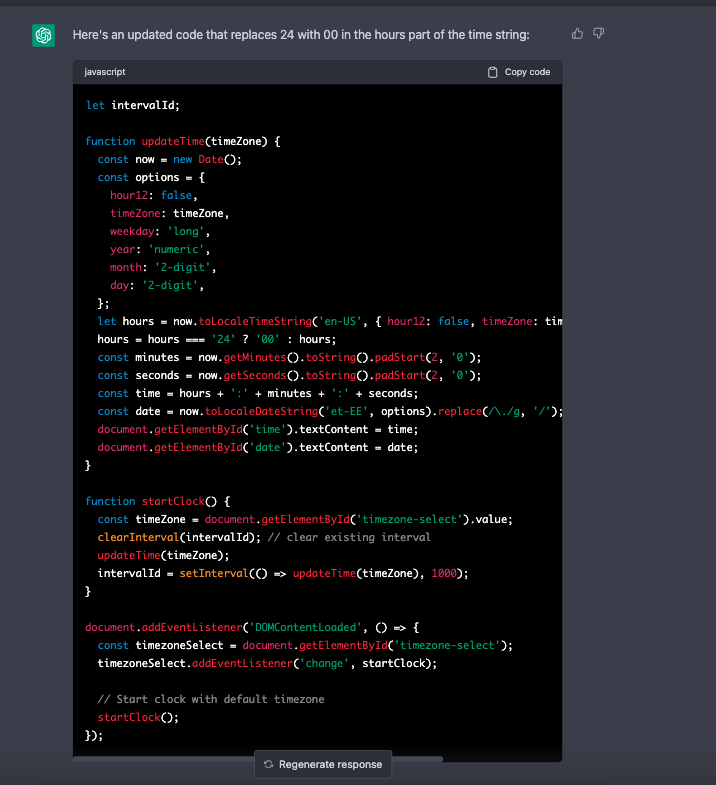




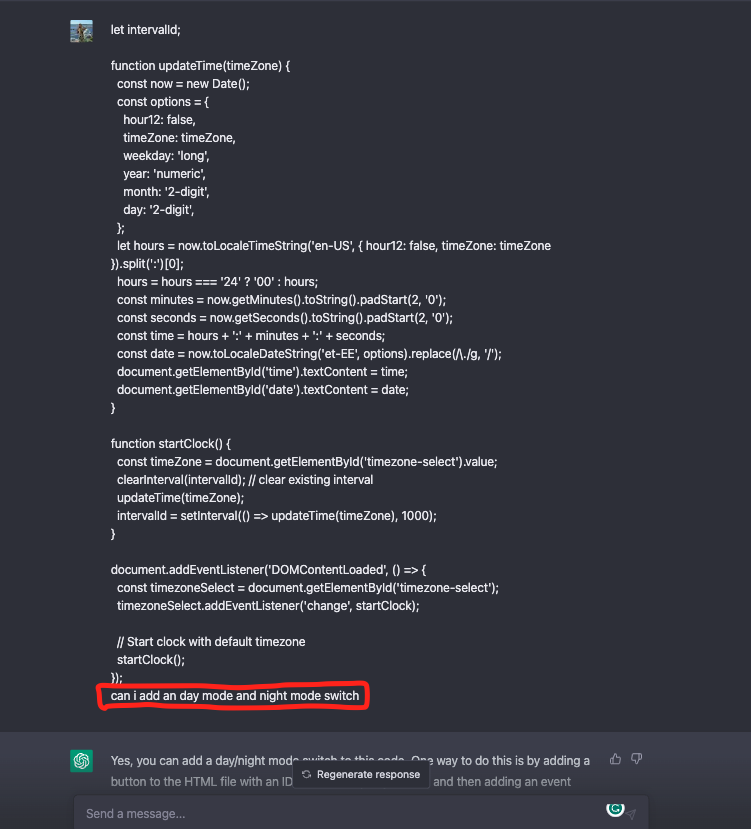


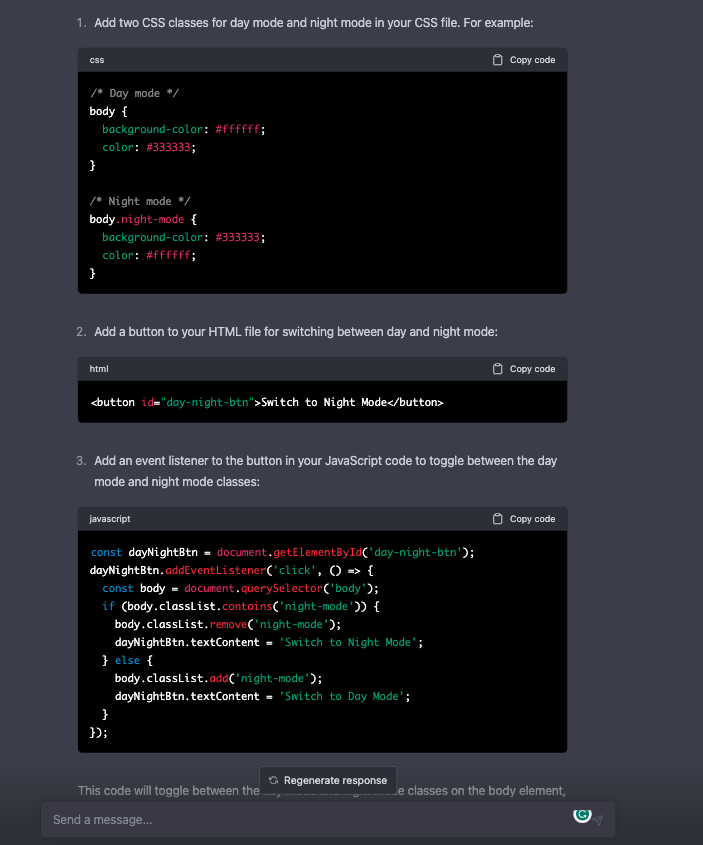
Märkasin ka, et kui kell on kesköö siis tundidena on kirjas 24, mitte 00, seega pidin paluma chatGPT-l javascripti muuta selliselt, et kui tunnid võrduvad 24-ga siis asendada see 00-ga.





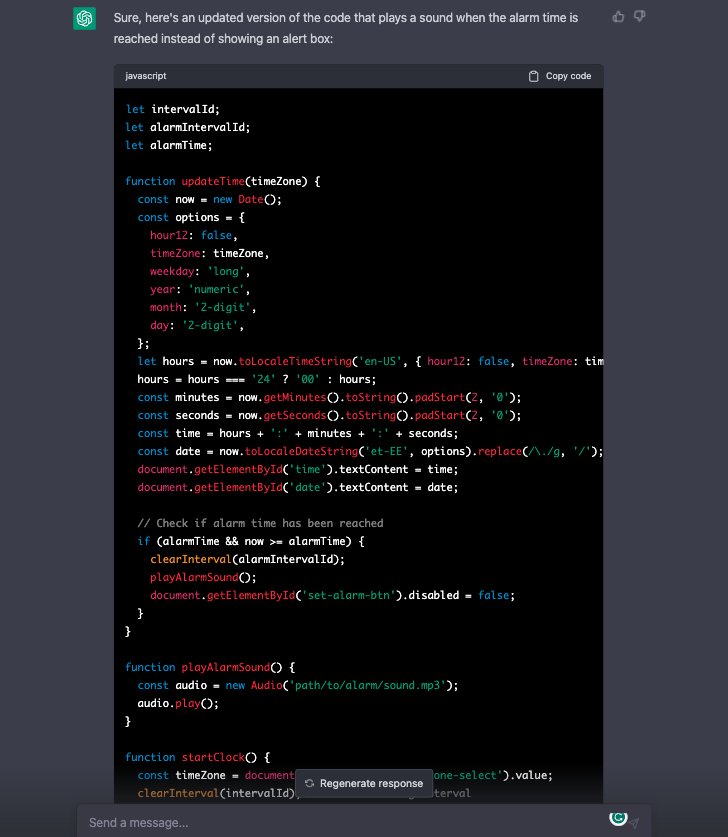
Teiseks funktsionaalsuseks soovisin kellale lisada juurde päeva- ja öörežiimi ehk nupu, mis võimaldaks panna peale dark mode (must taust ja valge tekst) või light mode (valge taust ja must tekst) ning nende vahel lülitada.

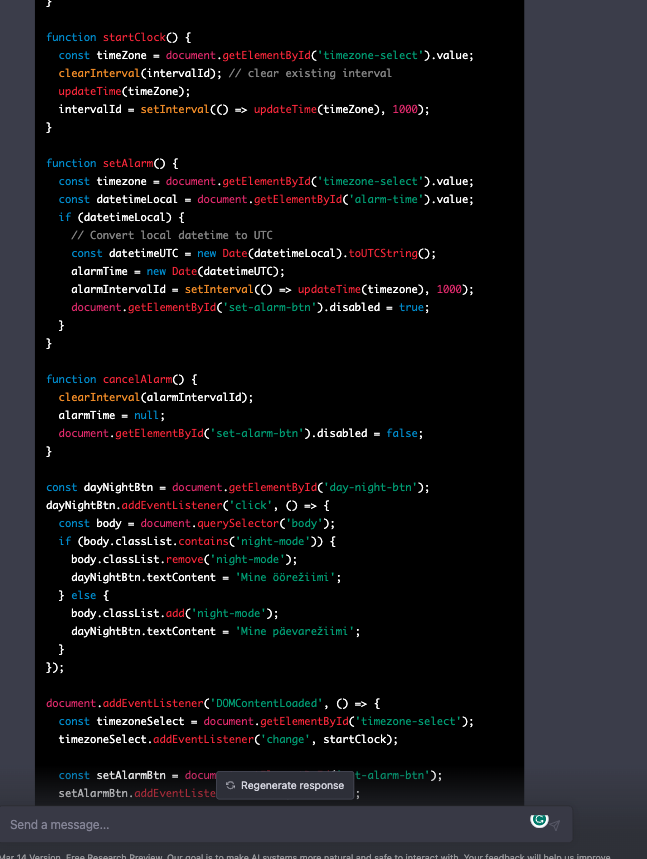


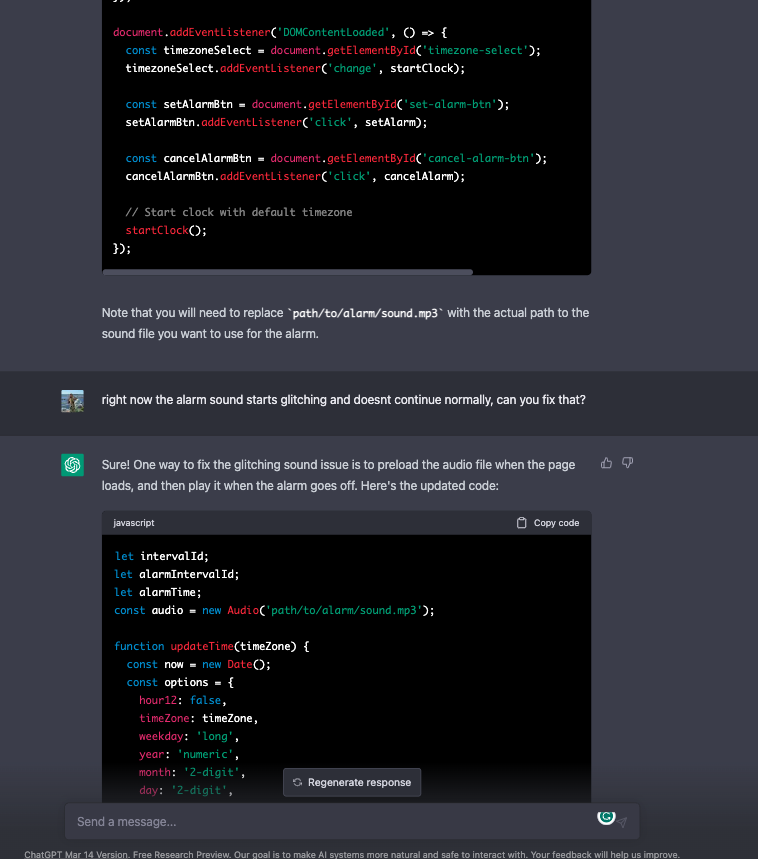


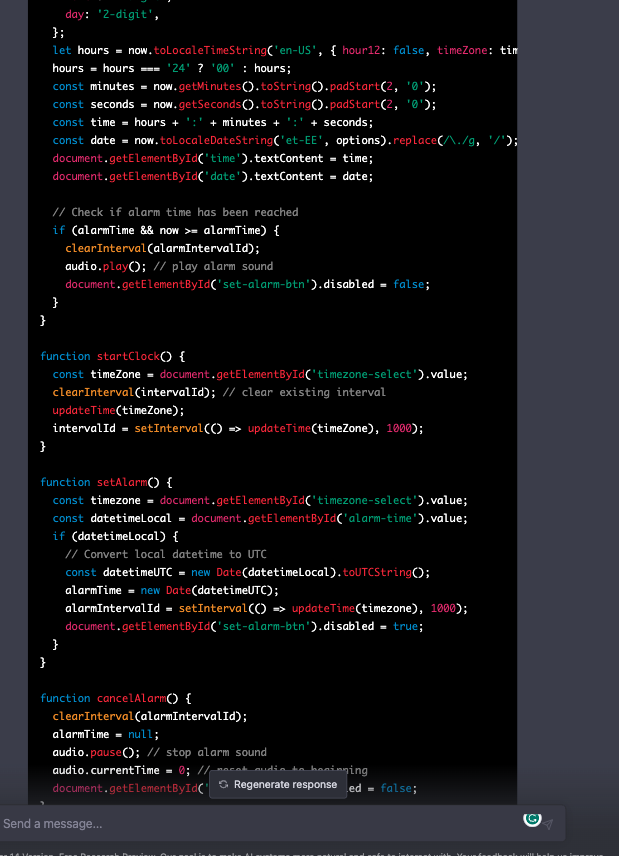
Kolmanda funktsionaalsusena soovisin lisada alarmi määramise võimaluse, mis laseks sisestada alarmi jaoks kuupäeva ja kellaaja ning, et reaalne heli ka kõlaks kui alarm helisema hakkab. Alguses sain chatGPT-lt sellise lahenduse, mis viskas ette alert box’i kui alarmile määratud aeg kätte jõudis. Siis palusin, et alert box’i asemel hakkaks hoopis heli mängima. Sisestasin chatGPT-sse oma praeguse koodi ja andsin lisaks käsu: **could you update this code so the alarm will actually also ring instead of the popup when the time comes.**

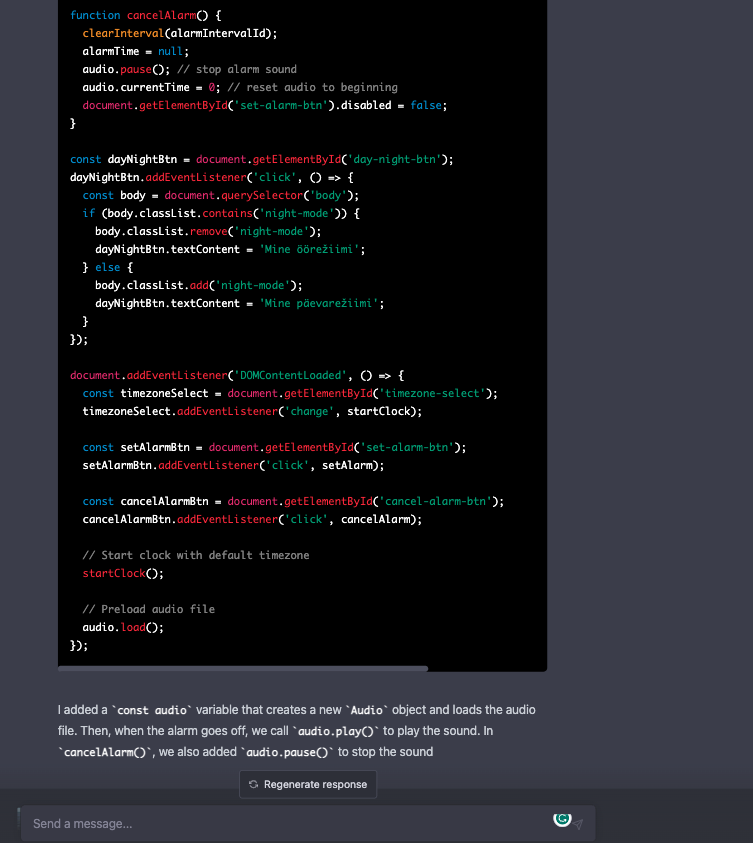
Siis omakorda tekkis selline probleem, et helis olid tõrked ja heli hakkas kogu aeg justkui otsast peale, palusin ka selle probleemi parandada.



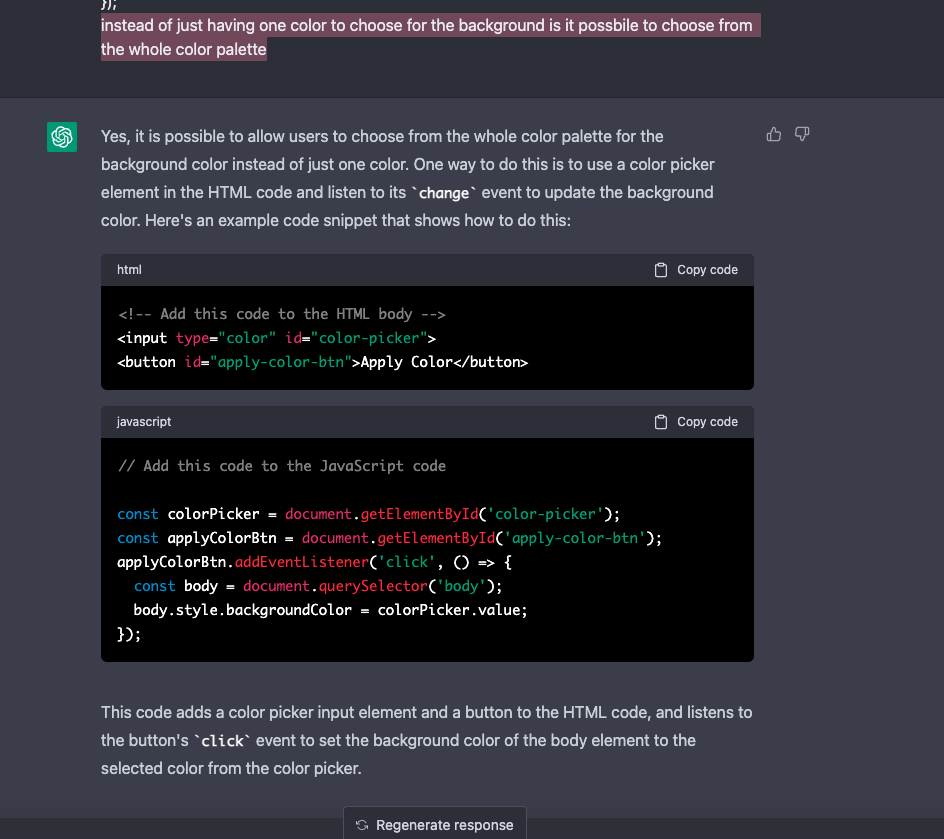








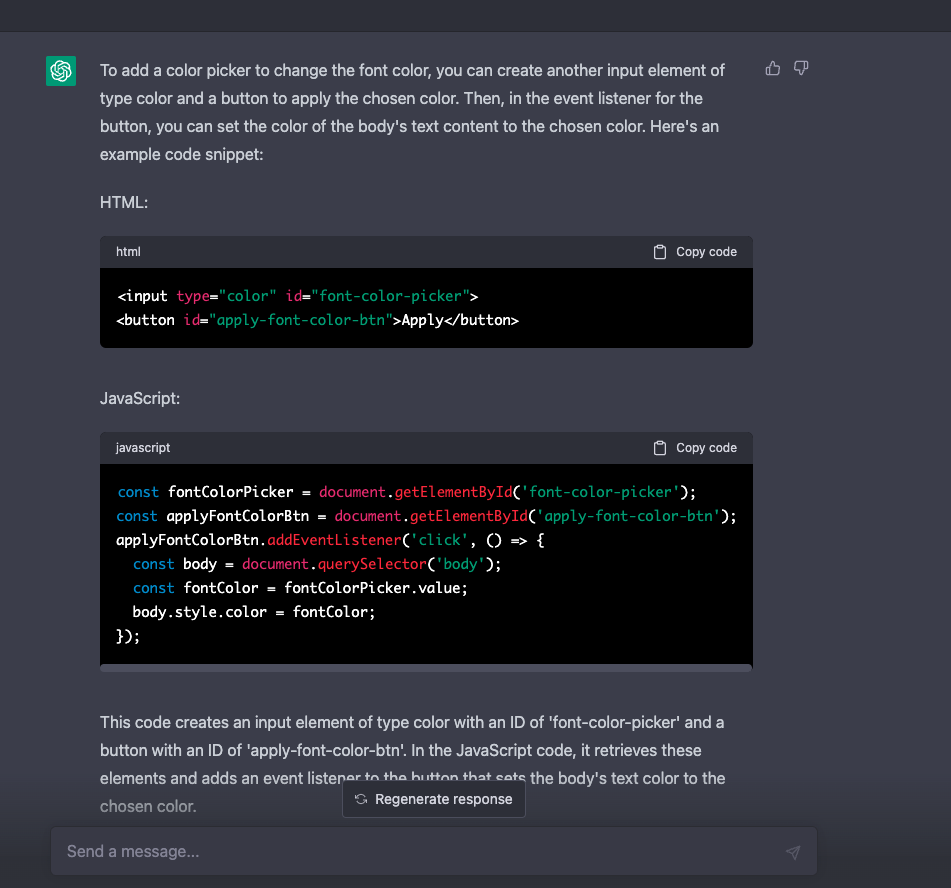
Järgmiseks soovisin lisada võimaluse taustavärvi muutmiseks. Selleks kopeerisin enda senise javascripti chatGPT-sse ja lisasin juurde käsu: **add me a button to change the background color.** Siis sain vastuseks koodi, mis võimaldas nupule klikkides muuta taustavärvi ainult siniseks, kuid kuna soovisin võimalust valida taustavärvi kogu värvipaletti hulgast siis muutsin oma käsku järgnevalt: **instead of just having one color to choose for the background is it possbile to choose from the whole color palette**

****

Siis tekkis probleem sellega, et kui vajutada “mine öörežiimi” või “mine päevarežiimi” nupule, ei toimunud need enam nii nagu ma tahaks, vaid taustavärv jäi ikkagi selleks, mis valitud oli. Mina aga soovisin, et öörežiimi puhul muutuks taust tagasi mustaks ja päevarežiimi puhul muutuks taust tagasi valgeks aga, et soovi korral saaks siis ikkagi taustavärvi muuta vastavalt soovile. Selleks andsin järgneva käsu: **could you update this code so when the dark mode button is clicked the background goes back to black and text color to white and when the day mode button is clicked the background goes back to white and text color back to black.** ChatGPT täiendas koodi järgnevalt:



Lisaks soovisin ka lisada võimaluse samamoodi teksti värvi muuta. Sisestasin chatGPT-sse oma hetke koodi ja kirjutasin lisaks **add me a color picker to change the font color.**

****

Viimase funktsionaalsusena lisasin + ja - nupud, mis võimaldavad kellaaja teksti suurust muuta. Selle võimaluse lisasin ise tunnis tehtud kella põhjal.

Viimaseks palusin chatGPT-l veel oma praegust CSS koodi täiendada selliselt, et see oleks responsive kõikidel ekraani suurustel. Kopeerisin chatGPT-sse oma hetke CSS koodi ja kirjutasin lisaks **could you update my css so that it would be responsive on all of the screen sizes** ning sain vastuseks sellise täiendatud CSS-i:

/\* General styles \*/

\*{

box-sizing: border-box;

}

body {

margin: 0;

font-family: 'Gill Sans', sans-serif;

font-size: 16px;

line-height: 1.5;

}

/\* Day mode \*/

body {

background-color: #ffffff;

color: black;

}

/\* Night mode \*/

body.night-mode {

background-color: #000000;

color: #ffffff;

}

.day-night-btn {

font-size: 1.5rem;

color: black;

margin-top: 1.5rem;

text-transform: uppercase;

border-radius: 0;

border-width: 1px;

padding: 10px;

margin-bottom: 1.5rem;

}

.alarm {

font-size: 1.2rem;

margin-top: 1.5rem;

}

.alarm-note {

font-size: 0.8rem;

margin-bottom: -2rem;

text-align: center;

}

button#set-alarm-btn,

button#cancel-alarm-btn {

font-size: 0.6rem;

color: black;

margin-top: 1.5rem;

text-transform: uppercase;

border-radius: 0;

border-width: 1px;

padding: 5px;

}

.container {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.time {

font-size: 10vw;

}

.date {

margin-top: 1.5rem;

font-size: 2rem;

text-transform: uppercase;

}

.timezone {

margin-top: 2rem;

font-size: 1.2rem;

}

.change-background-color {

margin-top: 1.5rem;

}

.change-background-color-btn,

.change-font-color-btn {

margin-top: 0.4rem;

font-size: 0.8rem;

color: black;

text-transform: uppercase;

border-radius: 0;

border-width: 1px;

padding: 5px;

}

/\* Media queries for responsive design \*/

@media screen and (min-width: 768px) {

.container {

max-width: 768px;

margin: 0 auto;

padding: 0 1.5rem;

}

.time {

font-size: 6rem;

}

.date {

margin-top: 2rem;

font-size: 2.5rem;

}

}

@media screen and (min-width: 992px) {

.container {

max-width: 992px;

}

.time {

font-size: 7rem;

}

.date {

margin-top: 2.5rem;

font-size: 3rem;

}

}

@media screen and (min-width: 1200px) {

.container {

max-width: 1200px;

}

.time {

font-size: 8rem;

}

.date {

margin-top: 3rem;

font-size: 3.5rem;

}

}

CSS-i puhul lasin chatGPT-l alguses teha algse kujunduse valmis, kuid jooksvalt täiendasin seda veel palju enda soovidele vastavalt. Ka pärast seda viimast täiendust chatGPT poolt tegin CSS-is mitmeid muudatusi ise.