Opinnäytetyö

Tieto- ja viestintätekniikka

2020

Joni Virtanen

PIKANÄPPÄINSOVELLUKSEN LUONTI WEB-TEKNOLOGIOILLA

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ AMK | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikka

2020 | 13

Joni Virtanen

PIKANÄPPÄINSOVELLUKSEN LUONTI WEB-TEKNOLOGIOILLA

Pikanäppäimet ovat näppäimiä, joihin voidaan määrätä tai ohjelmoida erinäisiä toimintoja. Eritoten näppäimistöissä on kautta vuosien ollut muokattavia pikanäppäimiä, varsinkin pelinäppäimistöissä. Pikanäppäimet näppäimistöissä ovat kuitenkin riippuvaisia kyseisestä laitteesta ja sen ohjelmistosta, mikä saattaa rajoittaa pikanäppäimien käytettävyyttä sekä toiminnallisuutta.

Työn tavoitteena oli luoda virtuaalinen pikanäppäinsovellus web-teknologioilla, jotta sen käyttö olisi mahdollista millä tahansa verkkoselaimella varustetulla laitteella.

Työ aloitettiin tutkimalla, millä web-teknologioilla sovellus olisi parhaiten toteutettavissa sekä etsimällä mahdollisia valmiita kirjastoja kyseisten web-teknologioiden käyttöön.

Toteutukseen käytettiin erinäisiä JavaScript-kirjastoja, jotka muun muassa mahdollistavat JavaScript-ohjelmoinnin selaimen ulkopuolella.

ASIASANAT:

JavaScript, Electron, Vue, pikänäppäin.

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2020 | 13

Joni Virtanen

HOTKEY APPLICATION USING WEB TECHNOLOGIES

Hotkeys are configurable keys which can be set to execute given tasks when pressed. Especially gaming keyboards have had these kind of keys for a long time. These have however been dependent on specific hardware.

The goal of this thesis was to create a standalone application for creating virtual hotkeys which can be accessed from any device with a modern web browser, for example an old mobile phone. Thus eliminating the need for purpose-built hardware.

The thesis was started by researching which web technologies would be most suitable for the application and searching for any JavaScript libraries implementing them.

The application was built using various JavaScript libraries and frameworks which were selected based on the research.

KEYWORDS:

JavaScript, Electron, Vue, hotkey

SISÄLTÖ

KUVAT

KÄYTETYT LYHENTEET TAI SANASTO

Backend	Sovelluksen taustakoodi, joka ei näy käyttäjälle
C++	Matalan tason ohjelmointikieli
Frontend	Käyttäjälle näkyvä osuus sovelluksesta
Mac OS	Applen kehittämä käyttöjärjestelmä
Material Design	Käyttöliittymän suunnittelustandardi
JavaScript	Korkean tason ohjelmointikieli
Linux	Käyttöjärjestelmäperhe, joka perustuu Linux -kerneliin
UI	User Interface eli käyttöliittymä
WebSocket	Protokolla, joka mahdollistaa reaaliaikaisen kaksisuuntaisen kommunikoinnin palvelimen ja asiakasohjelman välillä
Windows	Microsoftin kehittämä käyttöjärjestelmäperhe

1 JOHDANTO

Pikanäppäimet ovat olleet yleisiä näppäimistöissä pitkän aikaa. Kyseiset pikanäppäimet ovat kuitenkin laite- ja ohjelmistoriippuvaisia, mikä vähentää niiden käyttömahdollisuuksia. Tämän työn tarkoituksena on luoda laitteistosta riippumaton pikanäppäinsovellus, joka on mahdollisimman avoin eri käyttötarkoituksille.

Sovellus toteutetaan Windows käyttöjärjestelmälle käyttäen web-teknologioita sekä erinäisi JavaScript-kirjastoja. Tavoitteena on mahdollistaa sovelluksen komentaminen pikanäppäimillä verkkosivulta, jonka Windows-sovellus palvelee. Näin mahdollistetaan pikanäppäinten käyttö miltä tahansa verkkoselaimella varustetulta laitteelta.

Sovelluksen idea on mielenkiintoinen, sillä se mahdollistaisi pikanäppäimien käytön ilman pakollisia laitehankintoja sekä innovatiivisempien pikanäppäimien luonnin.

2 KIRJASTOT

Projekti toteutettiin JavaScript-kielellä sekä siihen pohjautuvilla koodikirjastoilla, sillä se oli ennestään tuttu muun muassa työelämän kautta. Se täytti myös kaikki sovelluksen tekniset vaatimukset, kuten reaaliaikaisuus ja samalla ohjelmointikielellä työpöytä- sekä verkkosovelluksen luonnin.

JavaScript on korkean tason ohjelmointikieli, joka on alun perin kehitetty verkkosivujen dynamisointia varten. Sen avulla on voitu tehdä verkkosivuista ”elävämpiä”, sekä mahdollistaa erinäisten toiminnallisuuksien toteutus selaimessa. [1]

Moderni JavaScript-ohjelmointi on kuitenkin viimevuosina laajentanut reviiriään kovaa vauhtia selainpohjaisen ohjelmoinnin ulkopuolelle. Sen käyttö palvelin- ja käyttöliittymäpuolella on helpottanut monien web-sovellusten elinkaaren yksinkertaistamista. [2]

2.1 Node.js

Node.js on avoimen lähdekoodin kirjasto, joka ajaa Google Chromen V8 JavaScript-moottoria selaimen ulkopuolella. Tämä mahdollistaa JavaScriptin käytön esimerkiksi palvelinpuolella, jolloin verkkosivustojen selain- ja palvelinkoodit voidaan molemmat kirjoittaa JavaScriptillä. [3]

V8 JavaScript-moottorin vuoksi Node.js on erittäin nopea ja sallii mahdollisten lisäosien kirjoittamisen C++ -ohjelmointikielellä, mitkä ovat osasyynä Node.js:n suureen suosioon. JavaScript-kielen käyttö selain- ja palvelinpuolilla nopeuttaa myös sovelluskehitystä ja vähentää tarvetta monien eri ohjelmointikielin osaajista. [4]

2.2 Electron

Electron on myös avoimen lähdekoodin kirjasto, joka yhdistää Chromium ja Node.js -teknologiat, mahdollistaen työpöytäsovellusten luonnin Mac OS, Windows ja Linux -alustoille web-teknologioilla. [5]

Myös kaikki vaikeimmat työpöytäsovelluksen luontiin liittyvät vaiheet, kuten paketointi, asennus ja päivitys, on tehty helpoksi Electron -kirjastoon rakennetuilla ominaisuuksilla

sekä lisäosilla. [6] Tämä teki Electronista täydellisen valinnan projektin työpöytäsovelluksen pohjaksi.

2.3 Socket.io

Socket.io on JavaScript-kirjasto, joka mahdollistaa reaaliaikaisen tapahtumapohjaisen kommunikoinnin laitteiden välillä, käyttäen WebSocket -teknologiaa aina kun mahdollista. [7]

Reaaliaikaisuuden ja helppokäyttöisyyden vuoksi Socket.io sopii täydellisesti projektiin, jolloin tieto pikanäppäimien painalluksista saadaan mahdollisimman nopeasti.

2.4 Vue.js

Vue.js on yksi monista JavaScript-kirjastoista, joiden tavoitteena on helpoittaa verkkosivun käyttöliittymän dynamisointia sekä parantaa reaktiivisuutta. Vue.js on kuitenkin toisia suosittuja kirjastoja, kuten Angular ja React, pienempi kooltaan sekä hieman helpommin opeteltavissa. [8]

Vue.js oli näistä syistä hyvä valinta projektin käyttöliittymien sovelluskehykseksi, jotta projekti etenisi mahdollisimman nopeasti ja vaivattomasti.

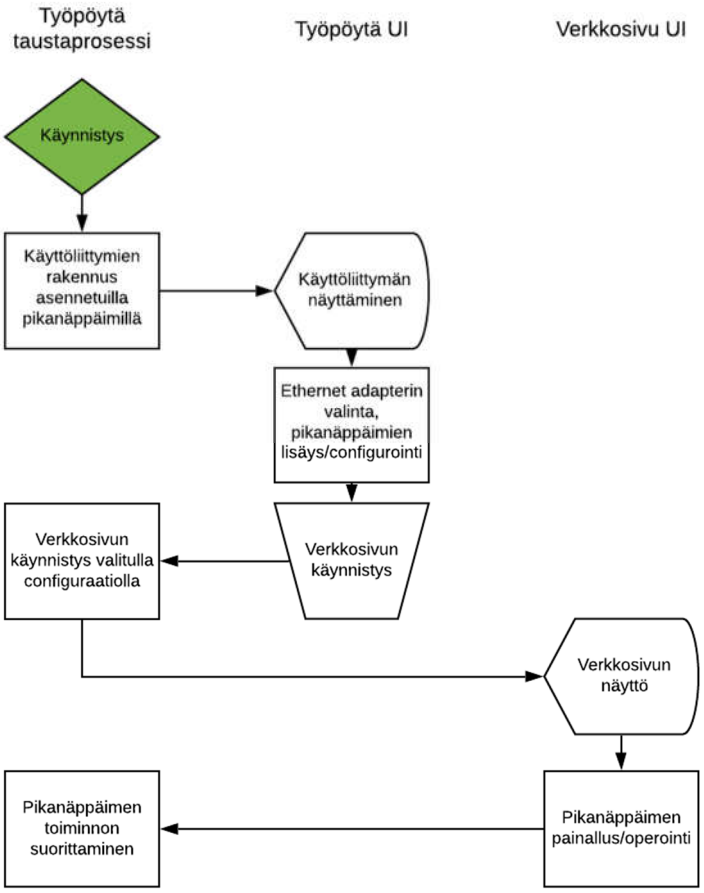
2.5 Vuetify

Vuetify on Vue.js alustalle kehitetty komponenttikirjasto, joka tarjoaa runsaasti valmiita käyttöliittymäkomponentteja, jotka noudattavat Material Design -standardeja. [9] Käyttämällä valmista komponenttikirjastoa on käyttöliittymä helpompi pitää yhtenäisenä ja säästetään huomattavasti aikaa sovelluskehityksessä.

3 TOTEUTUS

Sovelluksen perusperiaate on, että sovellukseen voidaan asentaa erinäisiä pikanäppäimiä modulaarisesti, minkä jälkeen työpöytäkäyttöliittymästä voidaan valita, mitkä niistä halutaan ottaa käyttöön. Myös mahdollisia pikanäppäinkohtaisia asetuksia voidaan määrittää. Pikanäppäimien asettamisen jälkeen voidaan esimerkiksi vanhalla puhelimella navigoida sovelluksen verkkosivulle, jonka kautta pikanäppäimiä voidaan käyttää. Pikanäppäimiä käytettäessä välittyy viesti työpöydälle ja kyseisen pikanäppäimen toiminto suoritetaan.

Ennen projektin varsinaisen koodin kirjoitusta sovelluksen toimintaa hahmoteltiin Lucidchart -verkkosivulla (kuva 1).



Kuva 1. Ylätason toimintamalli.

Toteutus aloitettiin pystyttämällä kehitysympäristö Electronille Windows 10 -alustalle käyttäen apuna Electronin omaa dokumentaatiota, jonka pohjalle luotiin sovellukselle välttämätön pohja. Koodieditorina käytettiin Visual Studio Codea.

3.1 Käyttöliittymä

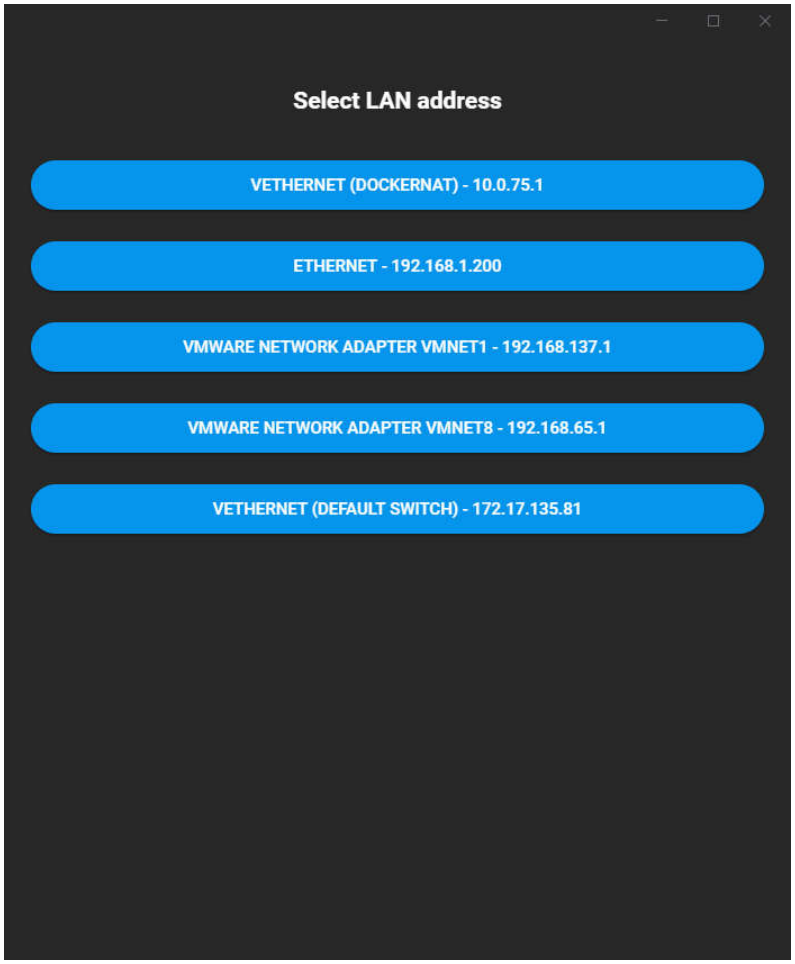
Käyttöliittymän toteuttamiseen päätettiin käyttää Vue.js kirjastoa, joka helpottaa suuresti käyttöliittymän rakennusta ja dynamisointia. Vuen kanssa käytettiin Vuetify kirjastoa käyttöliittymän rakentamiseen, joka noudattaa Material Design käytäntöjä.

Toteutuksessa pyrittiin luomaan mahdollisimman selkeä ja helppokäyttöinen ulkoasu, jonka saavuttamiseksi käytettiin apuna Vuetify -kirjastoa.

3.1.1 Työpöytäkäyttöliittymä

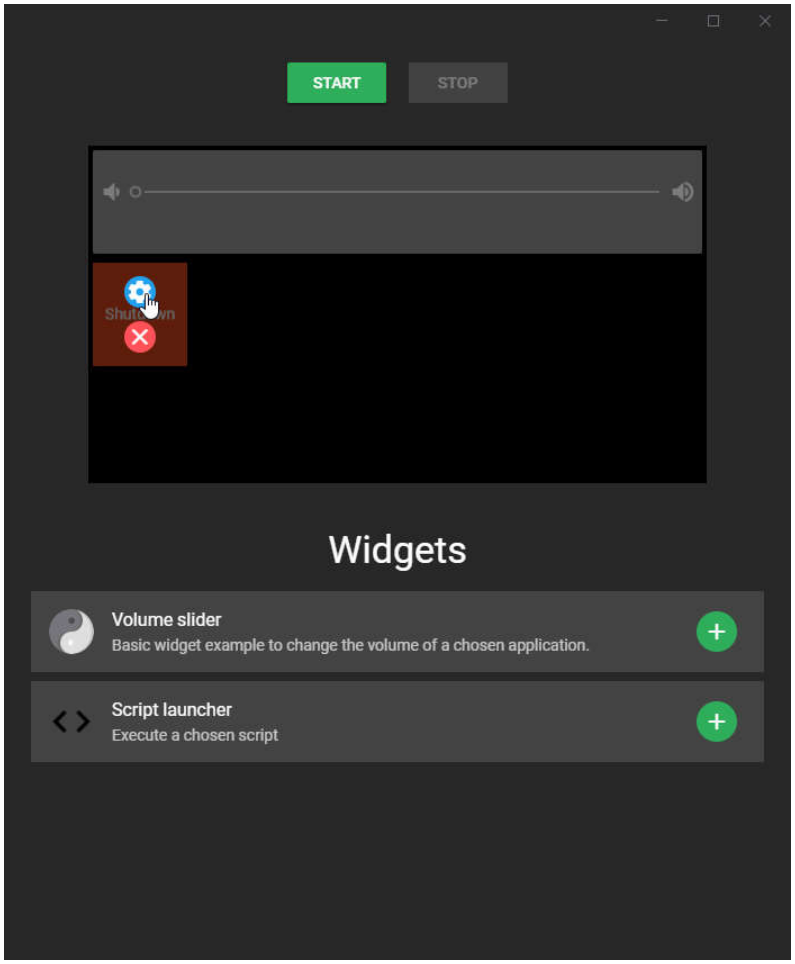
Sovelluksen työpöytäkäyttöliittymässä tarvitsee olla mahdollista valita haluttu ethernet osoite, josta verkkosivukäyttöliittymää tarjoillaan, sekä valita halutut pikanäppäimet ja asettaa niille mahdolliset asetukset.

Ethernet-osoitteen valinnalle luotiin oma näkymä, joka näytetään ensimmäisenä kun sovellus avataan (kuva 2).

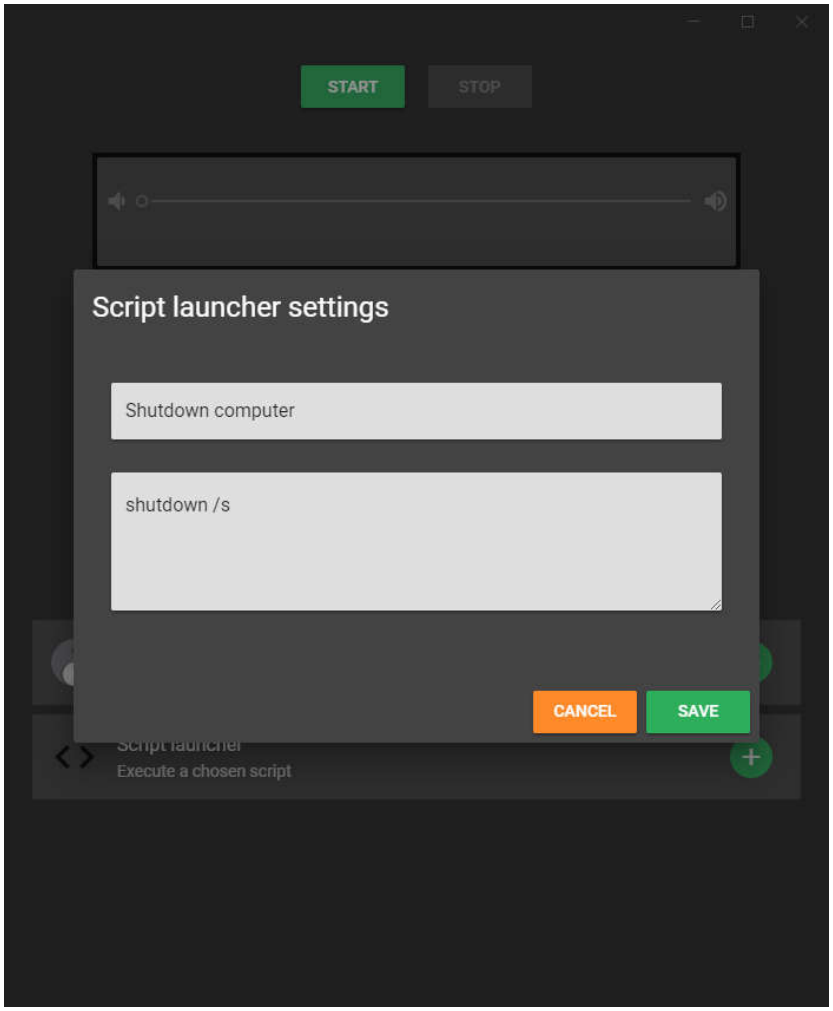


Kuva 2. Ethernet-osoitteen valitseminen.

Pikanäppäinten konfiguroinnille on oma näkymä, jossa näppäimiä voidaan lisätä sekä poistaa, ja vaihtaa näppäinten asetuksia (kuva 4). Myös pikanäppäinten taustatoiminta voidaan käynnistää samasta näkymästä ylhäältä ”Start” -painikkeesta. Näkymän keskellä on myös esikatselunäkymä, miltä mobiilikäyttöliittymä tulee valituilla näppäimillä näyttämään (kuva 3).



Kuva 3. Pikanäppäinten lisäys ja esikatselu.

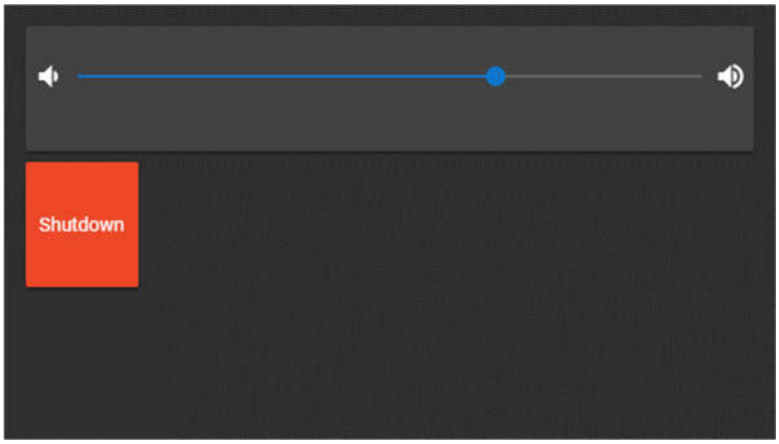


Kuva 4. Pikanäppäimen asetusikkuna.

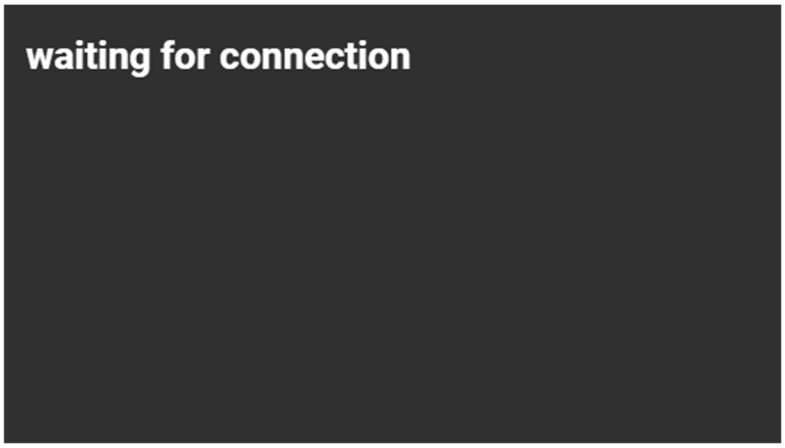
Kun näppäinten taustatoiminnallisuus käynnistetään, estetään pikanäppäinten lisäys ja muokkaus. Tämä siksi, että näppäinten lisäys ja muokkaus vaatii verkkokäyttöliittymän uudelleenluonnin, joka tapahtuu ”Start” -painiketta painettaessa.

3.1.2 Verkkosivukäyttöliittymä

Verkkosivukäyttöliittymän ei tarvitse kuin näyttää pikanäppäimet, jotka ovat valittuna (kuva 5). Verkkosivu ottaa yhteyden työpöytäsovellukseen Socket.io -kirjaston avulla, jonka kautta verkkosivu saa tiedon, mitä pikanäppäimiä käyttöliittymässä tulee näyttää. Jos työpöytäsovellukseen ei saada yhteyttä, näytetään tilaa kuvaava näkymä käyttäjälle (kuva 6).



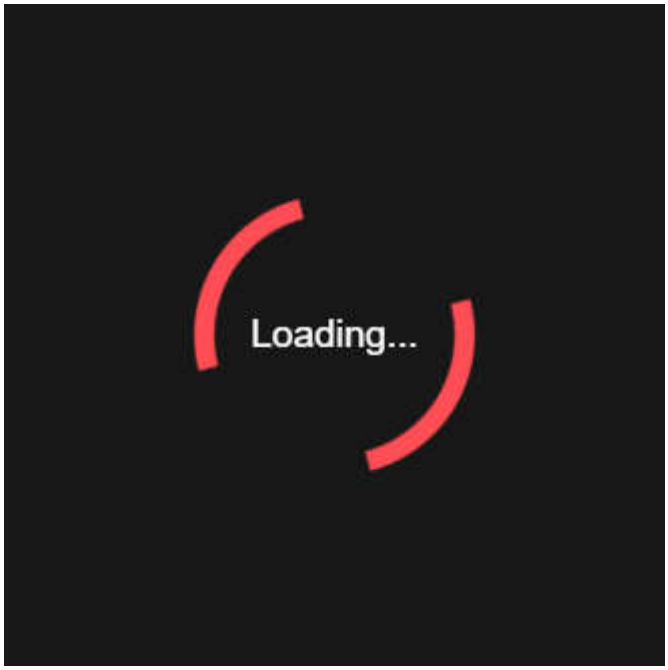
Kuva 5. Verkkosivukäyttöliittymä mobiililaitteella.



Kuva 6. Mobiilinäkymä kun työpöytäsovellukseen ei saada yhteyttä.

3.2 Taustatoiminta

Jotta pikanäppäinkomponenttien käyttö ja kehitys olisi mahdollisimman vaivatonta, pikanäppäinten lataus tehdään dynaamisesti. Tämän vuoksi molemmat käyttöliittymät joudutaan kääntämään uudestaan aina kun sovellus käynnistetään, joka hidastaa käynnistystä huomattavasti. Electron -taustaprosessi näyttää kuitenkin pienen latausikkunan tämän prosessin aikana.



Kuva 7. Lataus indikaattori

Kommunikointiin sovelluksen käyttöliittymän ja taustaprosessin välillä käytetään Electronin omia ipcMain ja ipcRenderer -rajapintoja. Näiden avulla mahdollistetaan esimerkiksi verkkosivukäyttöliittymän käynnistys työpöytäkäyttöliittymästä napin painalluksella.

Verkkosivukäyttöliittymän palvelemiseen käytetään Express -kirjastoa, joka palvelee käännettyä verkkosivukäyttöliittymää aiemmin valitusta verkko-osoitteessa. Verkkosivukäyttöliittymä keskustelee työpöytäprosessin kanssa Socket.io -kirjaston avulla.

4 PARANNUSMAHDOLLISUUDET

Koska harvan projektin tekemistä voi loputtomasti jatkaa, jäi tässäkin tapauksessa paljon tilaa parannuksille.

4.1 Pilvipalvelu

Verkkosivukäyttöliittymän palveleminen suoraan pilvipalvelun kautta saattaisi tehdä sovelluksen käytöstä merkittävästi helpompaa. Jotta tämä olisi mahdollista, pitäisi sovelluksella olla keskitetty kauppapaikka pikänäppäimille.

4.2 Asennus ja päivitys

Itsenäinen asennusohjelma sekä automaattipäivitykset nostaisivat myös sovelluksen käytettävyyttä. Tämän mahdollistamiseksi pitäisi sovelluksella olla kuitenkin oma palvelin, jonka kautta päivityksiä voitaisiin jakaa.

5 LOPUKSI

Työn tavoitteena oli luoda työpöydällä toimiva pikanäppäinsovellus käyttäen web-teknologioita. Tarkoituksena oli helpottaa tietokoneen jokapäiväistä käyttöä pikanäppäimillä ilman pakollisia laitehankintoja.

Työ tehtiin pääsääntöisesti JavaScript-ohjelmointikielellä käyttäen apuna erinäisiä kirjastoja kuten esimerkiksi Node.js, Electron, Vue ja Express. Kyseiset kirjastot osoittautuivat erittäin päteviksi, ja työ saatiin niiden avulla valmiiksi.

Sovellukselle jäi paljon tilaa jatkokehitykselle, kuten keskitetty kauppapaikka pikanäppäimille tai itsenäinen päivitys. Tästä huolimatta sovellus on osoittautunut hyödylliseksi jo omassa arkielämässä.

LÄHTEET

[1] An Introduction to JavaScript. Viitattu 11.11.2019. https://javascript.info/intro

[2] The status of JavaScript outside of the browser: 2018 & beyond. Viitattu 26.02.2019. https://hackernoon.com/the-status-of-javascript-outside-of-the-browser-2018-beyond-ee0b79ee059f

[3] Introduction to Node.js. Viitattu 10.03.2019. https://nodejs.dev/

[4] 6 Main Reasons Why Node.js Has Become a Standard for Enterprise-Level Organizations. Viitattu 17.02.2020. https://www.monterail.com/blog/nodejs-development-enterprises

[5] About Electron. Viitattu 10.03.2019. https://electronjs.org/docs/tutorial/about

[6] What is Electron JS? Viitattu 17.02.2020. https://brainhub.eu/blog/what-is-electron-js/

[7] What Socket.IO is? Viitattu 17.02.2020 https://socket.io/docs/

[8] Angular vs React vs Vue: Which Framework to Choose in 2020. Viitattu 19.3.2020 https://www.codeinwp.com/blog/angular-vs-vue-vs-react/

[9] Why Vuetify? Viitattu 26.3.2020 https://vuetifyjs.com/en/introduction/why-vuetify/#what-39-s-the-difference