



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Topi Järvinen

# WEB-KÄYTTÖLIITTYMÄN TOTEUTUS REACTILLA

Liiketalous  
2023

## TIIVISTELMÄ

Tekijä	Topi Järvinen
Opinnäytetyön nimi	Web-käyttöliittymän toteutus Reactilla
Vuosi	2023
Kieli	suomi
Sivumäärä	25
Ohjaaja	Päivi Rajala

---

ReactJS on suosittu JavaScript-kirjasto, jota käytetään käyttöliittymän rakentamiseen verkkosovelluksissa. Opinnäytetyön tavoitteena on syventyä Reactin keskeisiin ominaisuuksiin ja toteuttaa web-käyttöliittymä.

Teorian osassa käsitellään frontend-tekniikan ja Reactin keskeisiä teknologioita ja ominaisuuksia. Käsiteltäviin teknologioihin sisältyy DOM, HTML, CSS, JavaScript, versionhallintajärjestelmät, sekä ReactJS ja sen keskeiset ominaisuudet.

Käytännön osassa toteutetaan Moonfang, videopeliohje- ja uutissivusto, keskittyen käyttöliittymään. Opinnäytetyössä kuvaillaan sivun toteutuksen eri vaiheita ja Reactin ominaisuuksia kehitystyössä. Työssä hyödynnetään Reactin koukkuja (hookseja), komponenttipohjaisuutta sekä erikseen asennettavia React-kirjastoja, React Routersia navigointia varten, ja React-popups on käytössä popup-ikkunoita varten.

Lopputuloksena syntyi teos frontend-kehityksen ja Reactin keskeisistä teknologioista ja ominaisuuksista ja verkkosivusto Moonfang toimivalla käyttöliittymällä.

## ABSTRACT

Author	Topi Järvinen
Title	Implementation of the web user interface with React
Year	2023
Language	Finnish
Pages	25
Name of Supervisor	Päivi Rajala

---

ReactJS is a popular JavaScript library used for building user interfaces for web applications. The objective of the study was to delve into the key features of React and implement a web user interface.

The theoretical section covers essential frontend technologies and React's key technologies and features. The technologies discussed include DOM, HTML, CSS, JavaScript, version control systems, as well as ReactJS and its core capabilities.

In the practical section, a website called Moonfang, focusing on video game guides and news, is implemented with a focus on the user interface. The thesis describes the various stages of the project's development and the use of React's features in the development process. The work utilizes React hooks, component-based architecture, and separately installed React libraries, such as React Router for navigation and React-popups for popup windows.

The result is a comprehensive work on frontend development and React's key technologies and features, along with the functional Moonfang website with a user-friendly user interface.

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	7
1.1	Työn tausta .....	7
1.2	Työn tavoite .....	7
2	FRONTEND.....	8
2.1	DOM.....	8
2.2	HTML.....	8
2.3	CSS.....	9
2.3.1	CSS-kehykset (frameworks) .....	9
2.4	JavaScript .....	9
2.4.1	ECMAScript (ES) .....	11
2.4.2	JavaScriptin kehitysympäristöt .....	11
2.5	Versionhallintajärjestelmät.....	12
2.5.1	Git.....	13
2.5.2	Git-versionhallinnan käyttö kehityksessä .....	13
3	VERKKOSOVELLUSKEHITYS REACTILLA .....	15
3.1	React .....	15
3.2	Vite.....	15
3.3	Keskeiset verkkokehitysympäristön ominaisuudet .....	16
3.3.1	Virtuaalinen DOM .....	16
3.3.2	Komponentit .....	16
3.3.3	Deklaratiivinen ohjelmointi.....	16
3.3.4	JSX.....	17
4	VERKKOSIVUSTOPROJEKTIN SUUNNITTELU JA TOTEUTUS .....	18
4.1	Suunnittelu.....	18
4.2	Viten ja Reactin käyttöönotto.....	19

4.3	Verkkosivun toteutus.....	19
5	VALMIS KÄYTTÖLIITTYMÄ.....	24
6	YHTEENVETO .....	26
	LÄHTEET .....	27

## KUVALUETTELO

<b>Kuva 1.</b> Moonfang-logot	19
<b>Kuva 2.</b> App.jsx-komponentti	20
<b>Kuva 3.</b> useState-koukku (hook)	21
<b>Kuva 4.</b> handleClick-funktio	21
<b>Kuva 5.</b> Sivupalkin submenu, josta aukaistaan SingInModal-popup	22
<b>Kuva 6.</b> React Routerin Link-elementit navigointipalkissa	23
<b>Kuva 7.</b> Valmis sivusto desktop-näkymässä	24
<b>Kuva 8.</b> Valmis sivusto mobiilinäkymässä	25

# 1 JOHDANTO

## 1.1 Työn tausta

Web-kehitys jaetaan yleensä kahteen osaan, frontend-kehitykseen ja backend-kehitykseen. Frontend-kehitys tarkoittaa verkkosivujen ja -applikaatioiden käyttöliittymän kehittämistä. Tähän käytettävää teknologiaa ovat HTML, CSS ja JavaScript, sekä yleensä työhön sopiva JavaScript-kirjasto. Frontend-kehityksen tarkoituksena on tarjota käyttäjälle tapa vuorovaikuttaa sovelluksen tai sivuston kanssa.

Backend-kehityksellä tarkoitetaan työskentelyä kaiken kanssa, mikä ei ole näkyvää verkkosivulla. Työhön voi kuulua esimerkiksi tietokannat, API:t ja serverit.

Kehittäjää, joka työskentelee kummankin, frontendin ja backendin kanssa, kutsutaan yleensä fullstack-kehittäjäksi. Tämä tarkoittaa, että heillä on laaja tietämys ja taitotaso koko web-kehitysprosessissa. Fullstack-kehittäjä pystyy rakentamaan kokonaisii verkkosivustoja tai sovelluksia alusta loppuun ja ymmärtää, miten sekä käyttöliittymä että taustajärjestelmä toimivat yhdessä.

## 1.2 Työn tavoite

Tämä opinnäytetyö on empiirinen laadullinen tapaustutkimus, joka on toteutettu projektina. Työn tavoite on luoda web-käyttöliittymä videopelisivustolleni nimeltään Moonfang. Opinnäytetyön teoriaosassa keskitytään frontend-kehitykseen, ja käytettävään teknologiaan kuuluvat HTML, CSS, JavaScript, git sekä Vite + ReactJS.

Opinnäytetyön käytännön osuudessa suunnitellaan ja toteutetaan web-käyttöliittymä Moonfang-videopelisivustolle. Videopelisivusto suunnittelussa tarkastellaan aiemmin tehtyjä videopelisivustojen käyttöliittymiä. Moonfang-videopelisivuston kehittämisessä käytetään verkkosivustojen rakentamiseen soveltuvia tekniikoita: HTML, CSS, JavaScript ja React.

## 2 FRONTEND

Tässä luvussa kuvataan frontend-kehityksen keskeisiä käsitteitä ja teknologiaa.

### 2.1 DOM

DOM eli Document Object Model on ohjelmointialusta web-dokumenteille. Se esittää sivun elementit, kuten tekstit, kuvat ja linkit, hierarkkisina objekteina, joita voi muokata ja hallita.

Kun verkkosivu latautuu selaimessa, selain luo DOM-mallin sivusta, joka on kuin "virtuaalinen versio" sivusta. Tämä mahdollistaa JavaScriptin tai muiden ohjelmointikielten käytön sivun osien muokkaamiseen, lisäämiseen, poistamiseen ja päivittämiseen ilman tarvetta ladata koko sivua uudelleen. (Mozilla 2023.)

### 2.2 HTML

HTML eli HyperText Markup Language on merkintäkieli, jolla määritetään ja luodaan verkkosivun rakenne. Rakenne luodaan HTML-elementeillä. HTML-elementtejä ovat esimerkiksi otsikot, kappaleet, linkit, kuvat ja taulukot. Näiden elementtien avulla sivu ja sen sisältö voidaan järjestää selkeästi ja loogisesti. (Mozilla 2023.)

HTML-tagit ovat perusyksiköitä, joita käytetään HTML-kielessä elementtien määrittämiseen verkkosivuilla. Jokainen HTML-elementti alkaa avaus-tagilla ja päättyy sulkevaan tagiin, ja näiden tagien väliin sijoitetaan elementtiin liittyvä sisältö tai määrittelyt. HTML-tagit ovat keskeinen osa sivun rakentamista ja auttavat selainta tulkitsemaan, miten sivun eri osat pitäisi näyttää ja toimia. (Mozilla 2023.)

Esimerkki HTML-elementistä ja tageista:

```
<h1>Esimerkkiotsikko opinnäytetyöhön</h1>
```

`<h1>` on avaus-tag, joka aloittaa otsikkoelementin. `</h1>` on sulkeva tag, joka päättää otsikkoelementin.



HTML on perusta kaikille verkkosivuille, ja se toimii yhdessä muiden teknologioiden, kuten CSS:n ja JavaScriptin kanssa, jotta verkkosivut voivat tarjota monipuolisempia ja interaktiivisempia kokemuksia käyttäjille. (Mozilla 2023.)

## **2.3 CSS**

CSS eli Cascading Style Sheets on tyyliohjejärjestelmä, jota käytetään muokkaamaan verkkosivun visuaalista ulkonäköä.

Sivuston tyylin muokkaaminen tapahtuu CSS-tiedostossa, jossa muokataan HTML-elementtien ominaisuuksia, esimerkiksi niiden väriä, kokoa, tai sijaintia. Viittaus muokattaviin HTML-elementteihin voi tapahtua elementtien nimen, luokan nimen tai id-tunnisteen avulla. CSS-tiedosto liitetään HTML-tiedostoon, yleensä <head> tagin sisälle. (Mozilla 2023.)

### **2.3.1 CSS-kehykset (frameworks)**

CSS-kehykset ovat valmiita tyylitiedostoja ja komponentteja, jotka helpottavat verkkosivujen ja sovellusten ulkoasun suunnittelua ja toteutusta. Ne tarjoavat ennalta määriteltyjä tyylisääntöjä, rakenteita ja komponentteja, jotka voit ottaa käyttöön projektissasi. Esimerkkejä suosituista CSS-kehyksistä ovat Bootstrap ja Tailwind CSS.

Jokaisella näistä CSS-kehyksellä on omat vahvuutensa ja heikkoutensa, ja valinta riippuu projektin vaatimuksista ja henkilökohtaisista mieltymyksistäsi.

## **2.4 JavaScript**

JavaScript on ohjelmointikieli, jota käytetään laajasti web-kehityksessä interaktiivisten ja dynaamisten verkkosivujen ja -sovellusten luomiseen. Se toimii yhdessä HTML:n (HyperText Markup Language) ja CSS:n (Cascading Style Sheets) kanssa

tarjoten mahdollisuuden lisätä monenlaisia toiminnallisuuksia ja interaktiivisuutta verkkosivuille. (Mozilla 2023.)

Tässä on joitakin keskeisiä käyttötapoja ja ominaisuuksia, joita JavaScript tarjoaa web-kehityksessä:

**-Dynaaminen sisältö:** JavaScript mahdollistaa verkkosivujen päivittämisen ilman sivun uudelleen lataamista. Tätä käytetään esimerkiksi reaaliaikaisissa päivityksissä, kuten chat-sovelluksissa tai live-uutisvirroissa.

**-Käyttäjävuorovaikutus:** JavaScriptilla voidaan toteuttaa käyttäjän kanssa vuorovaikutuksessa olevia elementtejä, kuten lomakkeita, painikkeita, valikkoja ja pop-up-ikkunoita. Tämä parantaa käyttäjäkokemusta.

**-Validointi:** JavaScriptiä voidaan käyttää lomakkeiden validointiin, jotta varmistetaan, että käyttäjät syöttävät oikeanlaisia tietoja lomakkeisiin ennen niiden lähettämistä.

**-Ajax:** JavaScriptin avulla voidaan tehdä HTTP-pyyntöjä palvelimelle taustalla ilman sivun uudelleen lataamista. Tämä mahdollistaa dynaamisen sisällön hakemisen ja päivittämisen.

**-DOM-manipulaatio:** JavaScript voi muuttaa verkkosivun Document Object Model (DOM) -rakennetta, mikä tarkoittaa, että se voi lisätä, poistaa ja muokata sivun elementtejä ja niiden sisältöä.

**-Animaatiot:** JavaScriptilla voi luoda animaatioita ja liikkeitä verkkosivuilla, mikä voi tehdä sivuista visuaalisesti houkuttelevampia ja informatiivisempia.

**-Käyttäjätietojen tallennus:** JavaScript voi tallentaa tietoja käyttäjän selaimen paikalliseen tallennustilaan, kuten evästeisiin tai Web Storageen. Tämä on hyödyllistä esimerkiksi istunnon ylläpitämisessä tai asetusvalintojen tallentamisessa.

**-Kolmannen osapuolen kirjastot ja kehykset:** JavaScriptilla voi hyödyntää monia valmiita kirjastoja ja kehyksiä, kuten jQuery, React, Angular ja Vue.js, jotka tekevät web-kehityksestä nopeampaa ja helpompaa tarjoamalla valmiita ratkaisuja moniin yleisiin tehtäviin.

**-Frontend ja backend kehitys:** JavaScriptia voidaan käyttää sekä front-end (selainpuolen) että back-end (palvelinpuolen) kehityksessä. Front-endissä se liittyy verkkosivujen käyttäjäliittymään, kun taas back-endissä se voi toimia palvelimen puolella esimerkiksi Node.js-ympäristössä.

JavaScript on olennainen osa modernia web-kehitystä, ja se tarjoaa mahdollisuuden lisätä interaktiivisuutta, parantaa käyttäjäkokemusta ja tehdä verkkosivuista ja -sovelluksista monipuolisempia ja dynaamisempia.

#### 2.4.1 ECMAScript (ES)

JavaScript luotiin Netscapella verkon alkuaikoina, ja teknisesti "JavaScript" on Sun Microsystemsin (nykyisin Oracle) lisensoima tavaramerkki, jota käytetään kuvaamaan Netscapen (nykyisin Mozillan) kielen toteutusta. Netscape toimitti kielen standardointia varten ECMA:lle (European Computer Manufacturer's Association), ja tavaramerkkiongelmien vuoksi kielen standardoitu versio jäi jumiin hankalaan nimeen "ECMAScript". Käytännössä kaikki kutsuvat kieltä vain JavaScriptiksi. (Flanagan 2020.)

#### 2.4.2 JavaScriptin kehitysympäristöt

JavaScript-kehitystä voidaan toteuttaa monissa erilaisissa kehitysympäristöissä. Yksi suosituimmista kehitysympäristöistä JavaScript-kehittäjille on Visual Studio Code, joka on ilmainen, avoimen lähdekoodin koodieditori. Se tarjoaa monipuolisen ja tehokkaan työkalun JavaScript-kehitykseen.

Lisätietoja Visual Studio Codesta ja joistakin muista yleisistä JavaScript-kehitysympäristöistä:

**-Visual Studio Code (VS Code):** Visual Studio Code on kehitysympäristö, joka on erittäin suosittu JavaScript-kehittäjien keskuudessa. Se on kehitetty Microsoftin toimesta ja se on saatavilla Windowsille, macOS:lle ja Linuxille.

VS Code tukee laajennuksia, mikä tekee siitä erittäin mukautettavan ja monipuolisen. Siihen voi lisätä laajennuksia, jotka auttavat työskentelyssä, kuten IntelliSense (koodin automaattinen täydennys), vianmääritystyökaluja ja pakettien hallintaa. VS Code tarjoaa myös integroidun git-tuen versionhallintaan ja laajan yhteisön tukemana se tarjoaa monia opetusohjelmia ja laajennuksia, jotka helpottavat JavaScript-kehitystä. (Visual Studio Code n.d.)

**-WebStorm:** WebStorm on toinen suosittu JavaScript-kehitysympäristö, joka on erikoistunut web-kehitykseen. Se tarjoaa voimakkaan koodieditorin, vianmääritystyökalut ja integroidun git-tuen. WebStormilla on monia ominaisuuksia, jotka auttavat JavaScript-kehittäjiä, kuten live preview, refaktorointi, ja automaattinen koodianalyysi. (JetBrains n.d.)

**-Sublime Text:** Sublime Text on kevyt ja nopea koodieditori, joka on suosittu monien ohjelmoijien keskuudessa. Se on saatavilla useille alustoille ja tarjoaa laajan valikoiman laajennuksia JavaScript-kehitystä varten. (Sublime Text n.d.)

## 2.5 Versionhallintajärjestelmät

Versionhallintajärjestelmä on ohjelmisto, joka tallentaa ja hallitsee projektin tiedostojen ja koodin muutoksia ajan kuluessa. Sen tarkoitus on mahdollistaa useiden kehittäjien tai tiimien yhteistyö samassa projektissa, hallita versioita (erilaisia projektin tiloja) ja tarjota tehokkaita työkaluja ongelmien ratkaisemiseen ja konfliktien hallintaan.

### 2.5.1 Git

Git on hajautettu versionhallintajärjestelmä, joka on suunniteltu seuraamaan ja hallitsemaan muutoksia ohjelmistoprojekteissa. Se tarjoaa kehittäjille tavan tallentaa ja seurata projektin koodia ajan kuluessa. (git n.d.)

Tärkeimmät ominaisuudet ovat:

**-Versiohallinta:** Git tallentaa projektin kaikki versiot ajan myötä. Tämä mahdollistaa erilaisten versioiden vertailun ja palauttamisen tarpeen mukaan.

**-Haarat (Branches):** Git mahdollistaa haarojen luomisen, mikä tarkoittaa, että voi työskennellä erillään pääkehityslinjasta ilman, että vaikuttaa muihin kehittäjiin.

**-Yhdistäminen (Merge):** Kehittäjä voi tehdä oman osuuden valmiiksi erillään ja yhdistää sen lopuksi pääkehityslinjaan.

**-Hajautettu järjestelmä:** Git toimii hajautetusti, mikä tarkoittaa, että jokaisella kehittäjällä on oma paikallinen kopio projektista, ja muutokset voidaan synkronoida muiden kanssa tarvittaessa.

### 2.5.2 Git-versionhallinnan käyttö kehityksessä

**-Git-projektin alustaminen:**

Luodaan tyhjä Git-repositorio (tai kloonataan olemassa oleva) projektille. Tämä tehdään komennolla `git init` (tyhjä repo) tai `git clone` (kloonaus).

**-Työskentely Git-repositoriossa:**

Komennoilla `git add` lisätään muutokset seurantaan ja `git commit` tallentaa ne paikallisesti.

**-Harojen (branch) käyttö:**

Gitissä voi luoda haaroja, jotka ovat erillisiä koodin haaroja. Yleensä on hyvä käyttää haaroja ominaisuuksien, korjausten tai kehityksen eri osille. Haaran voi luoda komennolla `git branch` ja vaihtaa siihen komennolla `git checkout`.

**-Koodin yhdistäminen (merge) ja konfliktien ratkaisu:**

Kun haaroissa on tehty muutoksia, ne voi yhdistää päähaaraan komennolla `git merge`. Jos koodissa on ristiriitoja, ne on ratkaistava ne manuaalisesti.

**-Etärepositoriot (remote repositories):**

Usein projektit tallennetaan etärepositorioihin, kuten GitHub, GitLab tai Bitbucket. Paikallisen git-repositorion voi liittää etärepoon komennolla `git remote add`.

**-Päivitysten hakeminen ja lähettäminen etärepositorioon:**

Paikallisen repositorion voi päivittää hakemalla muutokset etärepositoriosta komennolla `git pull`. Paikalliset muutokset voi lähettää etärepositorioon komennolla `git push`.

### 3 VERKKOSOVELLUSKEHITYS REACTILLA

#### 3.1 React

React on Facebookin kehittämä avoimen lähdekoodin JavaScript-kirjasto, jota hyödynnetään käyttöliittymien toteutuksessa.

JavaScript-kirjaston pragmaattinen määritelmä on, että se on kokoelma toimintoja, joita voit käyttää koodissasi saavuttaaksesi tuloksia, jotka normaalisti vaativat enemmän koodia ja työtä sinulta. Kirjastot auttavat sinua kirjoittamaan parempaa ja lyhyempää koodia ja mahdollistavat tiettyjen ominaisuuksien nopeamman käyttöönoton. Lisäksi, koska voit keskittyä "ydinliiketoimintalogiikkaan", et vain etene nopeammin, vaan myös todennäköisesti tuotat parempaa koodia, koska sinun ei tarvitse keksiä pyörää uudelleen ongelmissa, jotka muut ovat aiemmin ratkaisseet. ([Schwarzmüller 2022.](#))

React on suosittu valinta modernien verkkosovellusten kehittämiseen, ja se on saanut laajan kehittäjäyhteisön tuen ja monia kolmannen osapuolen kirjastoja ja komponentteja. Se on käytössä monissa suurissa verkkopalveluissa, kuten Facebook itse, Instagram, Airbnb ja monet muut.

#### 3.2 Vite

Vite on nopea ja moderni kehitysympäristö JavaScript-sovelluksille, erityisesti suunniteltu React-sovellusten kehittämiseen. Se eroaa perinteisistä kehitysympäristöistä, kuten Webpackista tai Create React Appista, koska se pyrkii tarjoamaan huomattavasti nopeamman kehityskokemuksen. (Vite n.d)

Vite on suunniteltu erityisesti React-sovelluksille. Se sisältää valmiin React-pohjaisen projektirakenteen ja mahdollistaa React-sovellusten nopean käynnistämisen.

### **3.3 Keskeiset verkkokehitysympäristön ominaisuudet**

#### **3.3.1 Virtuaalinen DOM**

React on kirjasto, joka on suunniteltu päivittämään selaimen DOM puolestamme. Meidän ei enää tarvitse huolehtia tehokkaiden SPA-alueiden rakentamiseen liittyvistä monimutkaisista kysymyksistä, koska React voi tehdä sen puolestamme. Reactin avulla emme ole suoraan vuorovaikutuksessa DOM-sovellusliittymän kanssa. Sen sijaan annamme ohjeet siitä, mitä haluamme Reactin rakentavan, ja React huolehtii sen luovien elementtien hahmontamisesta ja yhteensovittamisesta. (Banks, Porcello 2020.)

#### **3.3.2 Komponentit**

Reactin keskeinen käsite on ns. komponenttien käyttö. Komponentit ovat uudelleenkäytettäviä rakennuspalikoita, jotka yhdistetään lopullisen käyttöliittymän muodostamiseksi. Esimerkiksi perussivusto voi koostua otsikosta, joka sisältää navigointipalkin ja pääosion, joka sisältää todennuslomakkeen. ([Schwarz Müller 2022.](#))

#### **3.3.3 Deklaratiivinen ohjelmointi**

Deklaratiivisellä ohjelmoinnilla tarkoitetaan ohjelmointityyliä, jossa sovellukset on rakennettu tavalla, joka asettaa etusijalle sen määrittelyn, mitä pitäisi tapahtua, eikä sitä, miten sen pitäisi tapahtua. (Banks, Porcello 2020.)

Esimerkiksi SQL-kysely on deklaratiiivinen, koska se määrittelee, mitä tietokannasta halutaan hakea, mutta ei sitä, miten haku toteutetaan.

Deklaratiivinen ohjelmointi tarjoaa useita etuja, kuten parannetun koodin luettavuuden, ylläpidettävyyden ja mahdollisuuden optimoida suorituskykyä taustalla. Se voi myös tehdä koodista vähemmän alttiin virheille, koska kehittäjän ei tarvitse hallita monimutkaisia tilanmuutoksia ja sivuvaikutuksia suoraan. Sen sijaan ohjelmistoa voi helpommin ymmärtää ja testata. (Bertram 2023.)



### 3.3.4 JSX

JSX tulee sanoista JavaScript XML ja se on erittäin hyödyllinen työkalu React-kehittäjille. JSX on JavaScript-kielen laajennus, joka tarjoaa tavan jäsentää komponenttien renderöintiä käyttämällä HTML:n kaltaista syntaksia. JSX antaa meille mahdollisuuden kirjoittaa HTML-elementtejä Javascriptissä ja sijoittaa ne DOM:iin muuntamalla HTML-tunnisteet React-elementeiksi ilman muita menetelmiä, kuten `createElement()` tai `appendChild()`. Tämä Javascriptin ja HTML:n yhdistelmä johtaa tehokkaampiin sovelluksiin tehostetulla suorituskyvyllä. (Arancio 2021.)

## 4 VERKKOSIVUSTOPROJEKTIN SUUNNITTELU JA TOTEUTUS

Tässä luvussa kuvataan käytännön osan suunnittelua ja toteutusta. Projektina luotiin videopeliohje- ja uutissivusto Moonfang. Sivuston toteutuksessa on keskitytty käyttöliittymään, eikä niinkään sisältöön.

Projektin toteutuksessa käytettiin Visual Studio Codea, Vite-kehityspalvelinta, Reactia, Node.js:ää, npm paketinhallintaa ja git-versionhallintaa. Kaikki CSS-tyylit on luotu itse ilman CSS-kehyksien käyttöä.

Sivuston navigointia varten asennettiin erillinen React Router kirjasto, joka on oma erillinen kirjasto, joka on suunniteltu erityisesti React-sovellusten reitityksen hallintaan. Se ei ole osa Reactia itsessään, mutta se on erittäin yleisesti käytetty ja suosittu kirjasto React-sovellusten reitityksen toteuttamiseen.

Pop-up ikkunoiden toteutukseen asennettiin Reactjs-popup, React-kirjasto, joka tarjoaa helpon tavan luoda ja hallita erilaisia pop-up-ikkunoita React-sovelluksissa. Kirjasto mahdollistaa erilaisten modaalien, ilmoitusten, valintapohjien ja muiden pop-up-tyylisten käyttöliittymäelementtien luomisen ja hallinnan React-sovelluksissa. Sitä käytettiin sisäänkirjautumisikkunaa varten.

### 4.1 Suunnittelu

Aloitin verkkosivuni suunnittelun tarkastelemalla muita videopelisivustoja ja niiden käyttöliittymiä. Seuraavaksi lähdin luomaan suunnitelmaa omasta käyttöliittymästäni Figmalla. Figma on suosittu verkkopohjainen suunnittelu- ja prototyypityökalu, joka on suunniteltu erityisesti käytettäväksi käyttöliittymän suunnittelussa.

Päädyn antamaan sivustolleni nimen Moonfang, ja loin sille logot ja sivulla käytettävät ikonit Canvalla.



**Kuva 1.** Moonfang-logot

## 4.2 Viten ja Reactin käyttöönotto

Aluksi pitää olla asennettuna Node.js ja npm-paketinhallinta. Seuraavaksi asennettiin Viten npm:n avulla. Kun Vite on asennettu, uusi projekti voidaan luoda komennolla `npm create vite@latest`. Tämän jälkeen syötetään projektin nimi ja valitaan käytettävä JavaScript-kirjasto. Tässä projektissa valittiin React. Siirrytään cd komennolla projektikansioon ja viimeistellään komennolla `npm install`. Lopuksi käynnistetään projekti komennolla `npm run dev`.

## 4.3 Verkkosivun toteutus

Toteutus aloitettiin luomalla komponentit käyttöliittymän perusrakennetta varten. Näihin kuuluivat navigointipalkki sivun yläreunaan, sekä vasempaan sivuun ja footer sivun alareunaan. App.jsx (kuva 2) komponenttiin luotiin elementit maindisplay ja maincontainer ja muokattiin niistä CSS:llä sopivan kokoiset. Vaihtoehtoisesti niistäkin olisi voinut luoda omat erilliset komponentit.

```

src > App.jsx > App
1  import { useState } from 'react'
2  import Sidebar from '../components/Sidebar'
3  import Navbar from '../components/Navbar'
4  import Footer from '../components/Footer'
5  import Homepage from '../components/Homepage'
6  import NewsPage from '../components/NewsPage'
7  import BaldursSite from '../components/BaldursSite'
8  import DiabloSite from '../components/DiabloSite'
9  import Diablo2Site from '../components/Diablo2Site'
10 import EldenSite from '../components/EldenSite'
11 import ForumsSite from '../components/ForumsSite'
12 import './App.css'
13
14 import { BrowserRouter as Router, Routes, Route }
15     | from 'react-router-dom'
16
17 function App() {
18     const [count, setCount] = useState(0)
19
20     return (
21         <div className="maincontainer">
22             <Sidebar/>
23             <div className="maindisplay">
24                 <Navbar/>
25                 <Routes>
26                     <Route path="/moonfang/" element={<Homepage/>} />
27                     <Route path="/gamenews" element={<NewsPage/>} />
28                     <Route path="/baldursgate3" element={<BaldursSite/>} />
29                     <Route path="/diablo4" element={<DiabloSite/>} />
30                     <Route path="/diablo2" element={<Diablo2Site/>} />
31                     <Route path="/eldenring" element={<EldenSite/>} />
32                     <Route path="/forums" element={<ForumsSite/>} />
33                 </Routes>
34                 <Footer/>
35             </div>
36         </div>
37     )
38 }
39
40
41
42 export default App

```

**Kuva 2.** App.jsx-komponentti

Navigointipalkeista löytyy linkit sivuston eri osioihin, kuten etusivulle, eri pelien sivuille, tai uutissivulle. Sivureunan navigointipalkin voi suurentaa, jolloin kuvaiko-

nien lisäksi näkyviin tulee linkkien nimet ja alaosaan tulee näkyviin sosiaalisen median linkit. Tämä toteutettiin onClick-funktiolla ja Reactin useState-koukulla (hook) (kuvat 3 ja 4). Kun käyttäjä klikkaa sivupalkin viivaikonia, vaihtuu sivupalkin CSS-luokka. Sidebar2-luokka on asetettu leveämmäksi.

```
const [style, setStyle] = useState("sidebar");
```

**Kuva 3.** useState-koukku (hook)

```
function handleClick() {  
  if (style === "sidebar") {  
    setStyle("sidebar2");  
  }  
  else {  
    setStyle("sidebar");  
  }  
  setShow(!show);  
}
```

**Kuva 4.** handleClick-funktio

Seuraavaksi luotiin React-popup-kirjaston avulla popup-ikkuna sisäänkirjautumista varten. Sivustolle ei pysty kirjautumaan, mutta toiminto lisätty käyttöliittymään. Popup-ikkuna aukeaa onClick-funktion ja Reactin useState-koukun avulla, kun käyttäjä klikkaa sisäänkirjautumislinkkiä.

```
<div className="submenu1">
  <a onClick={() => setOpen(true)}>SIGN IN</a>
  <Link> SIGN UP</Link>
</div>)}
<SignInModal setOpen={setOpen} open={open}/>
```

**Kuva 5.** Sivupalkin submenu, josta aukaistaan SingInModal-popup

Etusivulle luotiin sisältöä. Uutisehdotuksia voi navigoida oikealle tai vasemmalle. Uutislinkit laitettiin gridin (ruudukon) sisälle ja viereisillä nuolipainikkeilla voi siirtyä onClick-funktion avulla oikealle tai vasemmalle. Etusivulla on myös peliohje-ehdotus, sekä lomake uutiskirjeen tilausta varten.

Sivuston linkeistä pystyy myös navigoimaan uutissivulle, pelien omille sivuille, sekä foruumisivulle, niillä ei ole muuta sisältöä kuin otsikko.

Sivustolla navigoiminen toteutettiin React Router-kirjaston avulla. App.jsx komponenttiin lisättiin Routes-elementti, jonka sisällä määritettiin Route-komponentit (Kuva Nro). React Router tarjoaa erilaisia komponentteja, kuten Link, NavLink ja Redirect, jotka auttavat navigoinnissa ja linkkien luomisessa. Link-komponentti mahdollistaa käyttäjän siirtymisen eri reittien välillä, niitä käytettiin projektin linkkeinä.

```
<section className="navsection">  
  <Link to="/gamenews">NEWS</Link>  
  <Link to="/baldursgate3"> BALDUR'S GATE III</Link>  
  <Link to="/diablo4"> DIABLO IV</Link>  
  <Link to="/diablo2">DIABLO II</Link>  
  <Link to="/eldenring">ELDEN RING</Link>  
  <Link to="/forums">FORUMS</Link>  
</section>
```

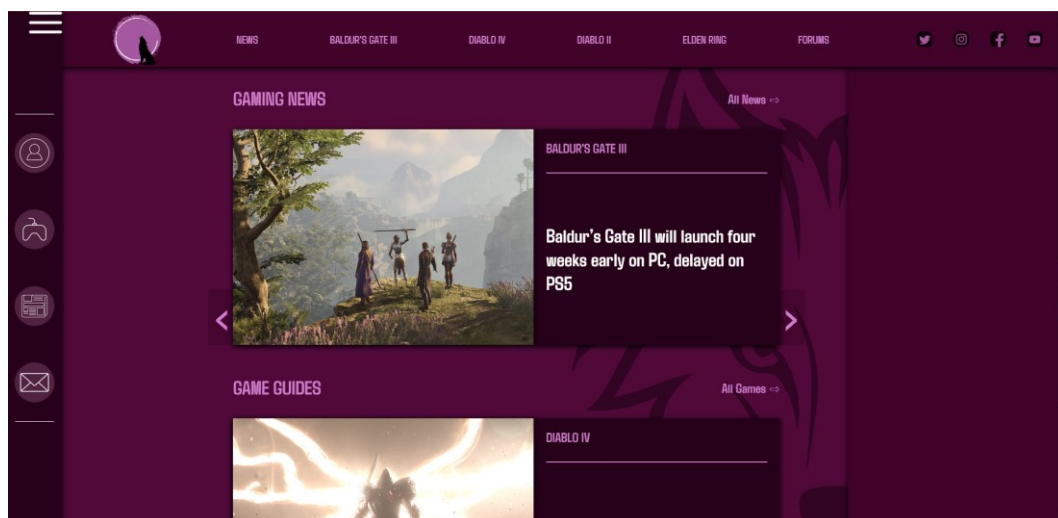
**Kuva 6.** React Routerin Link-elementit navigointipalkissa

Komponenttien CSS-tiedostoissa muokattiin Media Queryja apuna käyttäen sivustosta responsiivinen, jotta sivusto näyttää hyvältä pienilläkin näytöillä kuten mobiililaitteilla.

## 5 VALMIS KÄYTTÖLIITTYMÄ

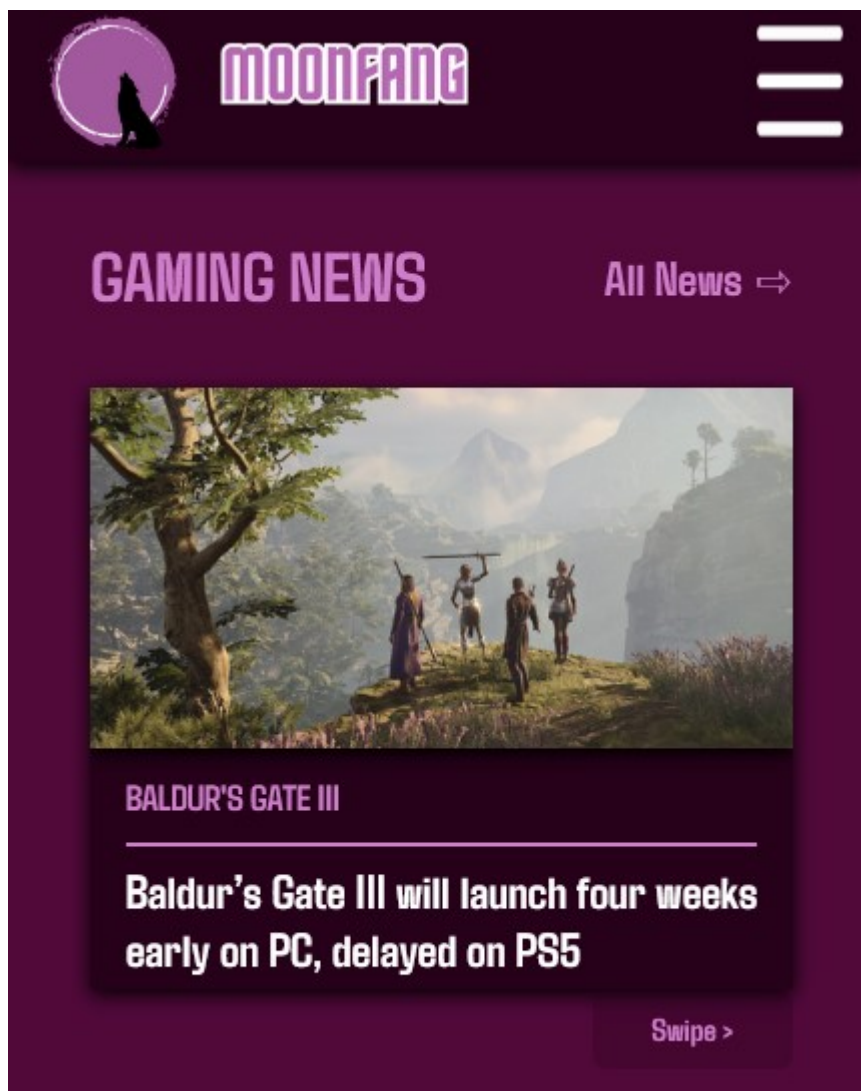
Tässä luvussa esitellään valmis verkkosivusto (kuva 7) ja sen käyttöliittymä, sekä analysoidaan, saavutettiinko projektille asetetut tavoitteet. Sivustosta tuli visuaalisesti miellyttävä ja se on täysin responsiivinen. Etusivulla, mobiililaitteilla voi käyttää kosketusnäyttöä uutisten selaamiseen.

Käyttöliittymä on helppokäyttöinen ja sivustolla navigointi sujuu nopeasti React Routerin ansiosta. Navigointia varten on vasemman reunan, sekä yläreunan navigointipalkit, joissa on linkit sivuston eri osuuksiin. Mobiililaitteilla käyttöliittymä (kuva 8) on hieman erilainen ja sisältää vain yhden laajennettavan navigointipalkin sivun yläreunassa. Kuvassa 2 nähdään kuinka komponentit muodostavat sivuston ja sen käyttöliittymän.



**Kuva 7.** Valmis sivusto desktop-näkymässä





**Kuva 8.** Valmis sivusto mobiilinäkymässä

## 6 YHTEENVETO

Projektin tavoitteena oli luoda oma verkkosivusto Moonfang React-JavaScript -kirjastoa käyttäen ja keskittyä käyttöliittymän kehitykseen. Työn tavoitteet täyttyivät ja tuloksena syntyi toimiva, helppokäyttöinen verkkosivu ja tietoa frontend-teknologiasta, keskittyen Reactin käyttöön ja ominaisuuksiin. Sivustolla on helppokäyttöinen, helposti navigoitava ja visuaalisesti miellyttävä käyttöliittymä.

Reactin käyttö on tehokasta ja sillä on helppo kehittää web-käyttöliittymiä. Työn jakaminen komponentteihin helpottaa eri osien hallintaa ja tekee kehitysprosessista selkeämmän.

Opinnäytetyö vastasi tutkimuskysymykseen, eikä sen tekemisessä ollut suurempia vaikeuksia. Työ sopii frontend-teknologiasta ja Reactista kiinnostuneille.

## LÄHTEET

Arancio, S. 2021. What is JSX? Viitattu 10.9.2023. <https://medium.com/@sjarancio/what-is-jsx-e3dda0af3490>

Banks, A. Porcello, E. 2020. Learning React, 2nd Edition. Sebastopol. O'Reilly Media, Inc.

Bertram, A. 2023. Declarative Programming. Viitattu 10.9.2023. <https://www.techtarget.com/searchitoperations/definition/declarative-programming>

Flanagan, D. 2020. JavaScript: The Definitive Guide, 7th Edition. Sebastopol. O'Reilly Media, Inc.

git. n.d. Documentation. Viitattu 12.9.2023. <https://git-scm.com/doc>

Schwarz Müller, M. 2022. React Key Concepts. Birmingham. Packt Publishing.

Sublime Text. n.d. Documentation. Viitattu 12.9.2023. <https://www.sublimetext.com/docs/>

Mozilla. 2023. Introduction to the DOM. Viitattu 14.9.2023 [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction)

Mozilla. 2023. CSS: Cascading Style Sheets. Viitattu 14.9.2023 <https://developer.mozilla.org/en-US/docs/Web/CSS>

Mozilla. 2023. HTML: HyperText Markup Language. Viitattu 14.9.2023. <https://developer.mozilla.org/en-US/docs/Web/HTML>

Visual Studio Code. n.d. Getting Started. Viitattu 13.9.2023. <https://code.visualstudio.com/docs>

Vite. n.d. Getting Started. Verkkosivu. Viitattu 11.9.2023. <https://vitejs.dev/guide/why.html>

WebStorm. n.d. Getting started with WebStorm. Viitattu 13.9.2023. <https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html>