



Aihetunnisteet osana tapahtumasovellusta

Aleksanteri Voutilainen

OPINNÄYTETYÖ
Kesäkuu 2024

Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

VOUTILAINEN, ALEKSANTERI:
Aihetunnisteet osana tapahtumasovellusta

Opinnäytetyö 32 sivua, joista liitteitä 0 sivua
Kesäkuu 2024

Opinnäytetyö käsitteli aihetunnisteiden integrointia Hubi-tapahtumasovellukseen, joka yhdistää tapahtumajärjestäjät ja käyttäjät. Työn tavoitteena oli kehittää ominaisuus, jonka avulla tapahtumajärjestäjät voivat lisätä aihetunnisteita tapahtumien luomisen yhteydessä. Tavoitteena oli parantaa käyttäjäkokemusta ja tuoda esiin uusia tapoja mainostaa tapahtumia sovelluksessa.

Kehitysprosessi kuvattiin yksityiskohtaisesti, ja siinä hyödynnettiin web-kehityksen teknologioita, kuten Next.js, PHP ja MySQL. Opinnäytetyössä tutkittiin myös aihetunnisteiden käyttöä ja niiden tarkoitusta eri alustoilla ja digitaalisissa ympäristöissä.

Lopputuloksena syntyi valmis toiminnallisuus, jossa websovelluksena toimivan admin-paneelin kautta voi lisätä ja hallinnoida aihetunnisteita tapahtumille. Aihetunnisteet tallennetaan tietokantaan, josta niitä voi jatkossa hyödyntää.

Aihetunnisteiden lisääminen Hubi-sovellukseen on tarjonnut vankan perustan tapahtumien luokittelulle ja löydettävyydelle. Seuraavana askeleena on kehittää käyttäjille kohdennettua sisältöä aihetunnisteiden perusteella, mikä parantaa heidän kokemustaan ja helpottaa tapahtumien löytämistä.

Asiasanat: web-kehitys, aihetunnisteet, ohjelmistokehitys

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Software Engineering

VOUTILAINEN, ALEKSANTERI:
Hashtags as Part of an Event Application

Bachelor's thesis 32 pages, appendices 0 pages
June 2024

This thesis addressed the integration of hashtags into the Hubi event application. The aim was to develop a feature allowing event organizers to add hashtags when creating events. The objective was to improve user experience and introduce new ways to promote events within the application.

The starting point was the existing functionality of the Hubi application, into which the new feature was integrated. The development process was detailed in the thesis, utilizing web development technologies such as Next.js, PHP, and MySQL. Additionally, the thesis examined the use of hashtags and their purpose across various platforms and digital environments.

As a result, the planned features were successfully integrated into the application. Event organizers can now add and manage hashtags for events via a web-based admin panel. Hashtags are added to a dedicated table in the MySQL database, where they can be utilized further.

The addition of hashtags to the Hubi application has provided a solid foundation for categorizing and finding events. This opens possibilities for future development, such as broader utilization of hashtags in the application's user interface to create targeted content for users.

Key words: web development, hashtags, MySQL, PHP, Next.js

SISÄLLYS

1	JOHDANTO	7
2	TARKOITUS JA TAVOITTEET	8
	2.1 Lähtökohdat	8
	2.2 Työn tavoite	8
3	AIHEEN ESITTELY.....	10
	3.1 Aihepiirin esittely / Aihetunnisteet /tagit.....	10
	3.2 Aihetunnisteet yleisesti.....	10
	3.3 Hubi sovellus.....	11
	3.3.1 Mobiilisovellus	11
	3.3.2 Admin paneeli.....	12
	3.3.3 Taustajärjestelmät	13
	3.4 Työn kannalta oleelliset teknologiat	13
	3.4.1 PHP	14
	3.4.2 MySQL	14
	3.4.3 React	14
	3.4.4 Next.js	15
	3.4.5 API.....	15
	3.5 MVC-arkkitehtuuri	16
4	KÄYTÄNNÖN TOTEUTUS	18
	4.1 Projektin eteneminen	18
	4.2 Käyttöliittymä muutokset	18
	4.2.1 Tag-komponentti.....	19
	4.2.2 AddNewTag-Komponentti	20
	4.2.3 TopTags-komponentti.....	21
	4.3 Taustajärjestelmän muutokset	21
	4.3.1 Tietokantamuutokset	22
	4.4 API	23
	4.4.1 Index.....	23
	4.4.2 Store.....	24
	4.4.3 React hook	25
	4.5 Testaus	26
	4.5.1 HTTP-metodit.....	26
	4.5.2 CRUD testaus.....	27
	4.5.3 Käyttöliittymättestaus.....	28
	4.6 Yhteenveto käytännön toteutuksesta	28
	4.7 Jatkokehitys	29

5 POHDINTA	31
LÄHTEET	32

ERITYISSANASTO tai LYHENTEET JA TERMIT (valitse jompikumpi)

Aihetunnisteet	Hashtagit, merkittyjä avainsanoja tai fraaseja, jotka alkavat ristikkomerkillä (#).
API	Application Programming Interface, joukko protokollia ja työkaluja ohjelmistojen väliseen kommunikointiin.
CRUD	Create, Read, Update, Delete, neljä perustavanlaatuaista toimintoa tietokantojen hallinnassa.
MVC	Model-View-Controller, ohjelmistokehityksen arkkitehtuurimalli.
Next.js	Reactin päälle rakennettu kehityskehys, joka optimoi suorituskäytettä ja tarjoaa valmiita työkaluja.
PHP	Hypertext Preprocessor, palvelinpuolen skriptikieli web-kehitykseen.
Props	Reactin ominaisuus, jolla komponentit voivat vastaanottaa dataa ja funktioita.
React	JavaScript-kirjasto, jota käytetään käyttöliittymien rakentamiseen.
SQL	Structured Query Language, standardoitu ohjelmointikieli tietokantojen hallintaan ja manipulointiin.

1 JOHDANTO

Tämän opinnäytetyön aiheena ovat aihetunnisteet ja niiden käyttö tapahtumasovelluksessa. Aihetunnisteet, eli hashtagit, ovat merkittyjä avainsanoja tai fraaseja, jotka alkavat usein ristikkomerkillä (#). Ne ovat tuttuja sosiaalisesta mediasta, missä niitä käytetään laajasti sisällön kategorisointiin ja hakutoimintoihin. (Tietotekniikan termitalkoot n.d.)

Tässä työssä tarkastellaan aihetunnisteiden käyttöä Hubi-nimisen sovelluksen kautta ja pyritään käytännössä kehittämään uusi ominaisuus tähän sovellukseen. Hubi on tapahtumasovellus, joka yhdistää tapahtumajärjestäjät ja käyttäjät samalle alustalle. Sovellus hyödyntää käyttäjän sijaintitietoja tarjotakseen personoituja suosituksia erilaisista tapahtumista, kuten konserteista, näyttelyistä, urheilutapahtumista ja muista kulttuuritapahtumista. (Hubiapp n.d.).

Opinnäytetyössä kehitetään ominaisuus, jonka avulla tapahtumanjärjestäjät voivat luoda aihetunnisteita tapahtuman luonnin yhteydessä. Näiden aihetunnisteiden avulla tapahtumat löytyvät sovelluksesta helpommin, ja niiden avulla voidaan kohdentaa käyttäjille personoituja suosituksia. Tavoitteena on luoda ominaisuus, joka parantaa käyttäjäkokemusta ja auttaa käyttäjiä löytämään kiinnostavia tapahtumia nopeammin.

Lisäksi työssä käsitellään aihetunnisteiden yleistä käyttöä ja merkitystä. Aihetunnisteiden avulla käyttäjät voivat tarkentaa kiinnostuksen kohteitaan. Aihetunnisteiden käyttö on vakiintunut käytäntö sosiaalisessa mediassa ja verkkokoupoissa, mikä tekee niistä tutun ja hyödyllisen ominaisuuden Hubi-sovellukselle.

2 TARKOITUS JA TAVOITTEET

Tässä luvussa esitellään opinnäytetyön aiheena ollut Hubi-sovellus sekä käydään läpi yleisellä tasolla aihetunnisteiden merkitystä. Tässä käydään myös läpi opinnäytetyön tavoite.

2.1 Lähtökohdat

Projektin lähtökohdana on toiminnassa oleva Hubi-sovellus ja sen toimivat järjestelmät. Ensimmäinen askel oli perehtyä perusteellisesti sovelluksen olemassa olevaan toiminnallisuuteen ja koodipohjaan. Tutustua sovelluksen nykyisiin ominaisuuksiin, rakenteisiin ja käytettyihin teknologioihin

Tämän pohjalta projekti eteni suunnitteluvaiheeseen, jossa määriteltiin tekniset vaatimukset uuden toiminnallisuuden toteuttamiseksi. Tässä vaiheessa tarkennettiin, miten uusi ominaisuus integroidaan sekä mobiilisovellukseen että admin-paneeliin, ja miten se toimii yhdessä olemassa olevien järjestelmien kanssa.

2.2 Työn tavoite

Opinnäytetyön tavoitteena on luoda pohja uudelle toiminnallisuudelle, jonka avulla tapahtumanjärjestäjät voivat lisätä aihetunnisteita tapahtumien luonnin yhteydessä. Esimerkiksi musiikkitapahtuman järjestäjä voi lisätä aihetunnisteet kuten musiikki, konsertti ja live. Aihetunnisteiden avulla tapahtumat voidaan kategorisoida ja näin kohdentaa tehokkaasti tapahtumia käyttäjille. Toiminnallisuuden toteutus pohjautuu aihetunnisteiden peruseräisiin, mutta se on räätälöity kyseisen sovelluksen tarpeisiin.

Opinnäytetyössä esitellään valmiin toiminnallisuuden toteutus ja sen vaikutukset sovelluksen käytettävyyteen. Uuden ominaisuuden kehitysprosessi kuvataan yksityiskohtaisesti. Tämä antaa kattavan kuvan siitä, miten toiminnallisuus rakennettiin ja miten se integroitiin osaksi Hubi-sovellusta.

Uuden toiminnallisuuden toteutuksen lisäksi opinnäytetyössä tarkastellaan aihe-
tunnisteiden käyttöä myös yleisellä tasolla ja niiden vaikutusta käyttäjäkokemuk-
seen.

3 AIHEEN ESITTELY

3.1 Aihepiirin esittely / Aihetunnisteet /tagit

Hubi on uusi tapahtumien hallinta- ja lipunmyyntisovellus, joka tarjoaa kattavat työkalut tapahtumien luomiseen, hallintaan ja markkinointiin. Se on suunniteltu sekä tapahtumanjärjestäjien että käyttäjien tarpeisiin, ja sen tavoitteena on tehdä tapahtumien järjestämisestä ja niihin osallistumisesta mahdollisimman sujuvaa ja helppoa.

Käyttäjät voivat helposti selata ja löytää tapahtumia sovelluksen kautta, joka tarjoaa monipuolisia hakutoimintoja ja suodattimia. Tapahtumia voidaan suodattaa sijainnin, ajankohdan tai kategorian perusteella, mikä tekee tapahtumien löytämisestä nopeaa ja vaivatonta.

Sovellus mahdollistaa tapahtumien luomisen helposti käyttäjäystävällisen käyttöliittymän kautta. Järjestäjät voivat määrittää tapahtuman yksityiskohdat, kuten ajankohdan, paikan, kuvauksen ja kategorian. Lisäksi he voivat hallita tapahtuman lipunmyyntiä. Tämä antaa järjestäjille täyden hallinnan tapahtumansa kaikista osa-alueista, mikä tekee prosessista tehokkaamman ja vähemmän aikaa vievän.

3.2 Aihetunnisteet yleisesti

Aihetunnisteet ovat digitaalisen sisällön hallinnan ja löydettävyyden keskeinen työkalu, joiden avulla käyttäjät voivat merkitä viestejään, kuviaan tai muita sisältöjään tietyillä aiheilla tai teemoilla. Kun käyttäjä lisää aihetunnisteen sisältöönsä, se tulee näkyväksi kyseisen aiheen tai teeman hakutuloksissa. Tämä parantaa sisällön löydettävyyttä ja auttaa käyttäjiä löytämään heille relevanttia tietoa nopeammin ja tehokkaammin.

Aihetunnisteiden käyttö on levinnyt laajasti eri digitaalisiin ympäristöihin. Niitä hyödynnetään esimerkiksi verkkosivujen hakukoneoptimoinnissa, tiedonhallintajärjestelmissä ja digitaalisessa viestinnässä. Tunnetuin käyttö tulee kuitenkin sosiaalisen median kautta, mistä aihetunnisteet ovat tuttuja kaikille.

Aihetunnisteiden avulla voidaan luoda tehokkaita hakutoimintoja ja järjestää suuria tietomääriä loogisesti ja käyttäjäystävällisesti. Tämä on perimmäinen syy, miksi aihetunnisteet halutaan tuoda myös Hubi-sovellukseen. Niiden käyttö on tuttua ja intuitiivista. Tapahtumasovelluksissa aihetunnisteet voivat toimia samankaltaisesti kuin sosiaalisessa mediassa. Kun tapahtumanjärjestäjät lisäävät tapahtumaan liittyviä aihetunnisteita, he voivat kohdentaa markkinointiviestejään tarkemmin tiettyyn kohdeyleisöön, joka todennäköisesti on kiinnostunut kyseisestä tapahtumasta.

Vaikka aihetunnisteita käytetään laajasti eri ympäristöissä, niiden peruseräatteen ovat samanlaiset alustasta riippumatta. Ne toimivat metatietoina, jotka parantavat sisällön löydettävyyttä ja auttavat käyttäjiä löytämään heille relevanttia tietoa. Menestys sosiaalisessa mediassa on osoittanut, kuinka tehokkaita tällaiset järjestelmät voivat olla, ja tämä on inspiroinut laajempaa aihetunnisteiden käyttöä muissa digitaalisissa ympäristöissä.

3.3 Hubi sovellus

Tässä osiossa käydään sovelluksen toiminta yleisesti läpi. Osion tarkoituksena on antaa lukijalle parempi käsitys sovelluksesta kokonaisuutena ja sen toiminnallisuudesta.

3.3.1 Mobiilisovellus

Mobiilisovellus on Hubi-sovelluksen keskeinen käyttöliittymä, joka tarjoaa käyttäjille tavan selata tapahtumia ja löytää heitä kiinnostavat kohteet. Sovellus on to-

teutettu React Nativella, mikä mahdollistaa sen toiminnan eri käyttöjärjestelmissä. Sovellus on saatavilla sekä Apple Storesta että Google Play -kaupasta. Kuvassa 1 on esillä mobiilisovelluksen käyttöliittymä.



Kuva 1 Kuvakaappaus mobiilisovelluksesta

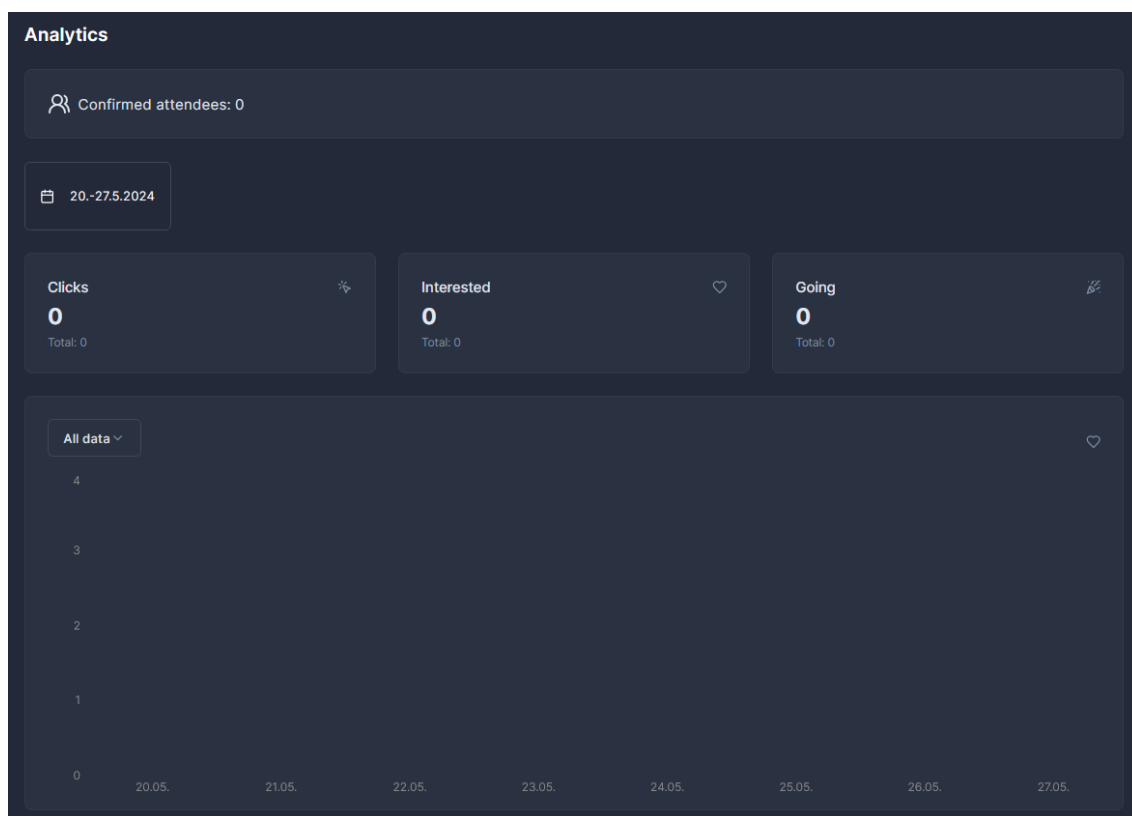
3.3.2 Admin paneeli

Admin-paneeli on yrityksille ja tapahtumanjärjestäjille tarkoitettu työkalu sovelluksen hallintaan ja tapahtumien ylläpitoon. Se on toteutettu Reactilla ja toimii web-sovelluksena.

Admin-paneelin avulla tapahtumanjärjestäjät voivat lisätä, muokata ja poistaa tapahtumia. He voivat syöttää tapahtumien tiedot, kuten nimen, kuvauksen, ajankohdan, sijainnin ja mahdolliset lipunhinnat, suoraan paneelistä. Lisäksi he voivat

määrittää tapahtumien kategoriat ja aihetunnisteet, jotka auttavat käyttäjiä löytämään tapahtumia sovelluksen kautta.

Tämän avulla voidaan myös seurata sovelluksesta kerättyä dataa. Sovellus kerää tietoa esimerkiksi tapahtumien kävijämäärästä sekä lipputuloista. Kuvassa 2 on näkymä Admin-paneelista, josta voi seurata tapahtumien dataa.



Kuva 2 Admin-paneelin Analytics näkymä testitapahtumasta

3.3.3 Taustajärjestelmät

Sovelluksen taustalla toimii PHP:llä toteutettu järjestelmä, joka käyttää MySQL-tietokantaa. Tämä taustajärjestelmä vastaa sovelluksen ydintoiminnoista, kuten tapahtumatietojen tallentamisesta, käyttäjähallinnasta ja sovelluksen suorituskyvyn varmistamisesta.

3.4 Työn kannalta oleelliset teknologiat

Tässä osiossa käsitellään opinnäytetyön kannalta keskeisimpiä teknologioita, jotka on hyödynnetty käytännön osuuden toteutuksessa. Teknologiat on valikoitu sen mukaan, miten sovellus ja sen taustajärjestelmät on alun perin rakennettu.

3.4.1 PHP

PHP on laajasti käytetty avoimen lähdekoodin ohjelmointikieli, joka on erityisesti suunniteltu web-kehitystä varten. Se toimii palvelinpuolella, tuottaen dynaamista sisältöä ja vuorovaikutteisia verkkosivustoja. PHP:n avulla voidaan rakentaa suorituskykyisiä, turvallisia ja käyttäjäystävällisiä web-sovelluksia kustannustehokkaasti ja nopeasti. (PHP n.d.)

3.4.2 MySQL

MySQL on tehokas relaatiotietokantajärjestelmä, joka tarjoaa vahvan tuen suurten tietomäärien käsittelyyn ja tallentamiseen. MySQL:n etuja ovat sen korkea suorituskyky, skaalautuvuus ja kyky käsitellä monimutkaisia kyselyitä tehokkaasti. Tietokannan indeksointi ja optimoidut hakutoiminnot mahdollistavat nopeat ja tarkat tiedonhauet, mikä on erityisen tärkeää aihetunnisteiden käytön kannalta. (MySQL n.d.)

3.4.3 React

React on suosittu avoimen lähdekoodin JavaScript-kirjasto, joka on suunniteltu rakentamaan käyttöliittymiä erityisesti yksisivuisille sovelluksille. Reactin vahvuuksia ovat sen komponenttipohjainen arkkitehtuuri ja kyky hallita tehokkaasti sovellusten tilaa ja käyttöliittymän päivityksiä. Reactin avulla kehittäjät voivat luoda uudelleenkäytettäviä komponentteja, mikä nopeuttaa kehitysprosessia ja parantaa sovellusten ylläpidettävyyttä. (React n.d.)

Hookit ovat Reactin tarjoamia funktioita, joiden avulla komponentit voivat käyttää Reactin ominaisuuksia ilman luokkapohjaisia komponentteja. Keskeisiä hookeja, joita käytettiin tässä opinnäytetyössä, ovat `useState` ja `useEffect`.

useState: Tämä hook mahdollistaa komponenttien tilan hallinnan. Sen avulla voidaan luoda ja päivittää tilamuuttujia komponentin sisällä. Tällä mahdollistettiin käyttäjän syöttämien tietojen hallinta ja käyttöliittymän tilan päivitys.

useEffect: Mahdollistaa sivuvaikutusten hallinnan React-komponenteissa. Sitä käytettiin muun muassa datan hakemiseen palvelimelta komponentin elinkaaren aikana.

useQuery: On osa React Query -kirjastoa, mikä mahdollistaa datan hakemisen palvelimelta ja sen välimuistitallennuksen. UseQuery mahdollisti tehokkaan tavan hakea ja käsitellä dataa. (Queries n.d.)

3.4.4 Next.js

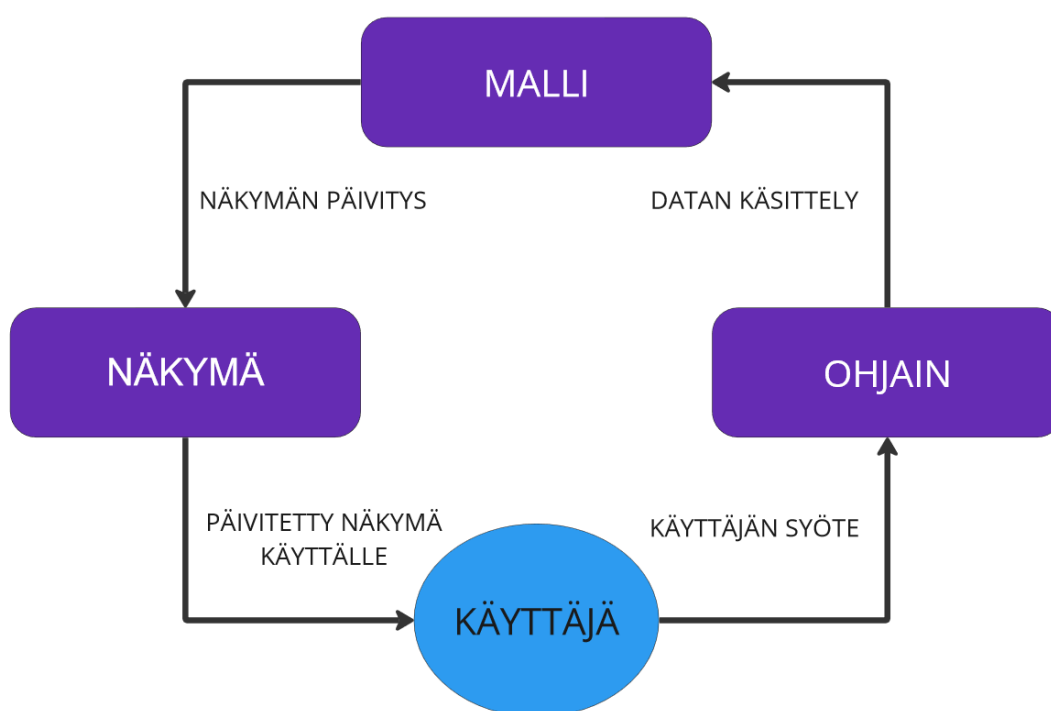
Next.js on Reactin päälle rakennettu avoimen lähdekoodin kehityskehyks. Se tarjoaa valmiiksi rakennetun reititysjärjestelmän ja työkaluja suorituskyvyn optimointiin. Näiden ominaisuuksien ansiosta Next.js on suosittu valinta kehittäjien keskuudessa, jotka haluavat rakentaa suorituskykyisiä ja helposti ylläpidettäviä web-sovelluksia. (Next.js n.d.)

3.4.5 API

Ohjelmointirajapinta eli API (Application Programming Interface) on joukko protokollia, rajapintoja ja työkaluja, joiden avulla eri ohjelmistokomponentit voivat kommunikoida keskenään. API:n avulla ohjelmat voivat vaihtaa tietoja ja kutsua toistensa toimintoja ilman, että niiden tarvitsee tietää toistensa sisäisestä toiminnasta. (Goodwin 2024.)

3.5 MVC-arkkitehtuuri

Model-View-Controller (MVC) on arkkitehtuurimalli, jota käytetään laajalti ohjelmistokehityksessä. (MVC Architecture n.d.) Hubi-sovelluksen taustajärjestelmä on rakennettu MVC-arkkitehtuurin mukaisesti, mikä helpottaa sovelluksen ylläpidettävyyttä ja skaalautuvuutta. Tässä osiossa käydään läpi, kuinka MVC-arkkitehtuuri on toteutettu Hubi-sovelluksessa. Esimerkkeinä käytetään käytännön toteutuksessa tehtyjä kohtia. Kuvassa 4 on yksinkertainen esimerkki MVC-diagrammista.



KUVA 3 MVC-diagrammi

Malli (Model) vastaa sovelluksen datasta ja liiketoimintalogiikasta. Esimerkiksi Tag-malli hallitsee aihetunnisteiden tietoja ja niiden käyttökerroista tietokannassa. Tag-mallin määritelmä sisältää kentät name ja count, joissa name tallentaa aihetunnisteen nimen ja count pitää kirjaa aihetunnisteen käyttökerroista. Lisäksi malli sisältää validointisäännöt, jotka varmistavat, että name on merkkijono ja count on kokonaisluku.

Näkymä (View) vastaa käyttäjälle näytettävästä käyttöliittymästä. Se hyödyntää tietokannasta tuotua JSON-dataa ja esittää sen käyttäjälle selkeässä muodossa. Esimerkiksi AdminTagController palauttaa JSON-muodossa aihe-tunnisteita, jotka voidaan näyttää käyttöliittymässä. Kuvassa 2 näkyy JSON-objekti, jossa on noudettu kymmenen eniten käytettyä aihe-tunnistetta.

```
Fetch most popular tags ▾ Array(10) [ {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...} ] event-form
  0: Object { id: 36, name: "sunshine", count: 798, ... }
  1: Object { id: 39, name: "vacation", count: 675, ... }
  2: Object { id: 33, name: "spring", count: 562, ... }
  3: Object { id: 46, name: "hiking", count: 517, ... }
  4: Object { id: 38, name: "flowers", count: 435, ... }
  5: Object { id: 42, name: "outdoors", count: 405, ... }
  6: Object { id: 41, name: "gardening", count: 323, ... }
  7: Object { id: 35, name: "blossom", count: 275, ... }
  8: Object { id: 34, name: "summer", count: 192, ... }
  9: Object { id: 37, name: "picnic", count: 166, ... }
  length: 10
```

KUVA 4 JSON-objekti

Ohjain (Controller) käsittelee käyttäjän pyynnöt, suorittaa liiketoimintalogiikkaa ja käyttää mallia tietojen hakemiseen ja päivittämiseen. Esimerkiksi AdminTagController hallitsee aihe-tunnisteisiin liittyviä pyyntöjä, kuten niiden lisäämistä ja päivittämistä. index-metodi hakee kymmenen käytetyintä aihe-tunnistetta tietokannasta ja palauttaa ne JSON-muodossa. store-metodi vastaanottaa listan aihe-tunnisteita, tarkistaa, ovatko ne jo olemassa, ja päivittää niiden käyttömäärää tai luo uusia aihe-tunnisteita tarvittaessa. update-metodi päivittää olemassa olevien aihe-tunnisteiden käyttömäärää.

4 KÄYTÄNNÖN TOTEUTUS

Tässä osiossa tarkastellaan yksityiskohtaisesti Hubi-sovellukseen tehtyjä muutoksia ja lisättyjä toiminnallisuuksia. Osion tarkoituksena on antaa kattava kuvaus käyttöliittymään ja taustajärjestelmään tehdyistä muutoksista, sekä selventää niiden taustalla olevia teknisiä ratkaisuja.

4.1 Projektin eteneminen

Kehitysprosessia vietiin eteenpäin suunnittelemalla aihetunnisteiden logiikka ja määrittämällä niiden tarpeellisuus. Suunnitteluvaiheessa päätettiin lisätä käyttöliittymään inputkenttä, jossa käyttäjät voivat syöttää aihetunnisteita. Vaihtoehtoisena ratkaisuna harkittiin raahaa ja pudota -valikkoa, mutta tästä olisi voinut valita vain ennalta määritetyt aihetunnisteet, mikä rajoitti joustavuutta.

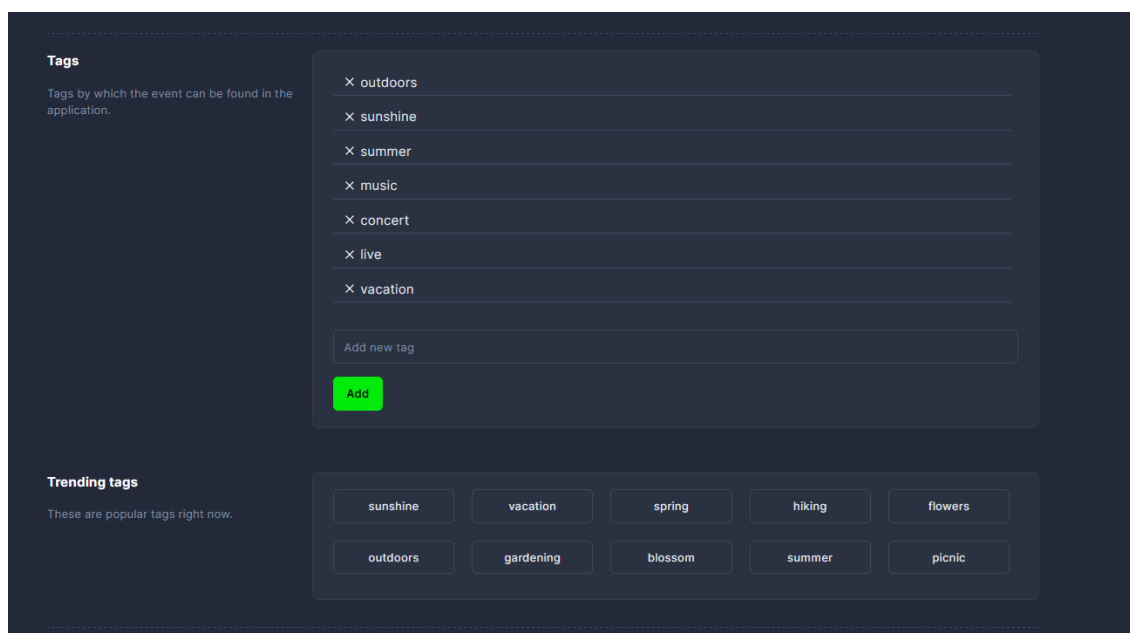
Kun aihetunnisteiden perustoiminnot saatiin valmiiksi, heräsi ajatus suosituimpien tagien lisäämisestä. Luotiin erillinen osio, joka näyttää suosituimmat aihetunnisteet. Tarkoituksena oli luoda kokonaisvaltaisempi ominaisuus. Suosituimpien aihetunnisteiden lisääminen vaati taustajärjestelmän muokkausta, jotta suosituimmat aihetunnisteet voitiin rajata tiettyyn määrään ja niiden hakeminen tietokannasta saatiin toimimaan.

Viimeisenä uudet ominaisuudet testattiin eri menetelmillä. Näin varmistettiin, että kaikki lisätyt ominaisuudet toimivat suunnitellusti ja että ne integroituivat olemassa olevaan järjestelmään.

4.2 Käyttöliittymä muutokset

Admin-paneeliin on integroitu uusia toiminnallisuuksia, joiden avulla tapahtumanjärjestäjät voivat lisätä aihetunnisteita tapahtumiensa yhteydessä sekä korostaa

suosituimpia aihetunnisteita. Näiden ominaisuuksien toteuttamiseksi käyttöliittymään lisättiin kolme keskeistä komponenttia. Kuvassa 1 on kuvakaappaus käyttöliittymästä.



KUVA 5 Käyttöliittymä aihetunnisteille

4.2.1 Tag-komponentti

Tag-komponentti tarjoaa toiminnallisuuden yksittäisen aihetunnisteen näyttämiseen ja poistamiseen. Se käyttää Reactin useState-hookia hallitsemaan hover-tilaa, joka muuttaa poistonapin väriä hiiren ollessa sen päällä. Tag-komponentti saa kaksi propsia: text ja onDelete. text-propsi on merkkijono, joka edustaa näyttävää aihetunnistetta, kun taas onDelete-propsi on funktio, joka kutsutaan, kun käyttäjä klikkaa poistonappia. Näiden propsien avulla komponentti voi näyttää eri aihetunnisteita ja suorittaa erilaisia toimintoja poistamisen yhteydessä. Kuvassa 2 on kuvakaappaus Tag-komponentin koodista.

```

const Tag = ({ text, onDelete }: { text: string; onDelete: () => void }) => {
  const [isHovered, setIsHovered] = useState(false);
  return (
    <div className="flex items-center border-b px-3 py-1 mr-2 mb-2">
      <button
        type="button"
        onClick={onDelete}
        className="mr-1"
        onMouseEnter={() => setIsHovered(true)}
        onMouseLeave={() => setIsHovered(false)}>
        <X size={16} color={isHovered ? 'red' : 'white'} />
      </button>
      <span className="mr-1">{text}</span>
    </div>
  );
};

```

KUVA 6 Tag-komponentti

4.2.2 AddNewTag-Komponentti

AddNewTag-komponentti mahdollistaa uusien aihetunnisteiden lisäämisen tapahtumiin ja olemassa olevien aihetunnisteiden hallinnan, mukaan lukien suosituimpien aihetunnisteiden korostamisen ja poistamisen.

Komponentti käyttää 'useState'-hookia hallitsemaan uuden aihetunnisteen syötettä 'newTag' ja olemassa olevia aihetunnisteita 'tags'. Käyttäjä syöttää uuden aihetunnisteen tekstikenttään, ja 'handleInputChange'-funktio päivittää 'newTag'-tilan reaaliajassa. Aihetunniste lisätään painamalla Enter-näppäintä tai klikkaamalla "Lisää" -painiketta, mikä kutsuu 'handleAddTag'-funktion. Tämä funktio tarkistaa syötetyn tekstin, lisää sen aihetunnisteiden listaan ja kutsuu 'onAddTag'-funktioita, joka on välitetty propsina.

Aihetunnisteiden poistaminen hoidetaan 'handleDelete'-funktioilla, joka poistaa aihetunnisteen joko 'tags'-tilasta tai 'toptags'-tilasta riippuen siitä, mistä se on lisätty. Olemassa olevat aihetunnisteet näytetään 'Tag'-komponentin avulla, joka mahdollistaa yksittäisten aihetunnisteiden poistamisen visuaalisten vihjeiden avulla.

Suosituimmat aihetunnisteet tallennetaan 'toptags'-tilaan ja näytetään samalla tavalla kuin käyttäjän lisäämät aihetunnisteet. 'setToptags'-funktio päivittää suosituimmat aihetunnisteet, joita voidaan poistaa 'handleDelete'-funktiolla.

4.2.3 TopTags-komponentti

'TopTags'-komponentti vastaa suosituimpien tagien hallinnasta. Tämän avulla käyttäjä voi lisätä suoraan suosituimmat tagit omaan tapahtumaan, jos ne ovat aiheellisia.

Komponentti käyttää 'useTags'-custom hookia hakemaan aihetunnisteita palvelimelta. 'useState'-hookia käytetään tallentamaan suosituimmat aihetunnisteet ('toptags'), ja 'useEffect'-hookia käytetään päivittämään tila, kun aihetunnisteet haetaan ja järjestetään niiden käyttöiheyden perusteella. Kun aihetunnisteet on haettu, ne järjestetään 'count'-sarakkeen perusteella laskevaan järjestykseen. Tämän jälkeen otetaan kymmenen suosituinta aihetunnistetta ('topTenTags'), jotka tallennetaan 'toptags'-tilaan. Tämä mahdollistaa suosituimpien aihetunnisteiden nopean ja tehokkaan hallinnan.

Kun käyttäjä klikkaa aihetunnistetta, 'onTagClick'-funktio kutsutaan, mikä lisää aihetunnisteen tapahtumaan ja päivittää 'toptags'-tilan lisäämällä kyseisen aihetunnisteen listan loppuun. Tämä toiminnallisuus tarjoaa helpon ja nopean tavan lisätä suosituimmat aihetunnisteet tapahtumiin.

4.3 Taustajärjestelmän muutokset

Hubi-sovelluksen taustajärjestelmä on toteutettu PHP:llä ja se käyttää MySQL-tietokantaa. Tämä taustajärjestelmä vastaa sovelluksen ydintoiminnoista, kuten tapahtumatietojen tallentamisesta, käyttäjähallinnasta ja sovelluksen suorituskyvyn varmistamisesta. Tässä osiossa tarkastellaan yksityiskohtaisesti Taustajärjestelmään tehtyjä muutoksia, mukaan lukien tietokantamuutokset ja API-päätepisteiden muutokset.

4.3.1 Tietokantamuutokset

Tietokantaan lisättiin uusi tags-taulu, joka sisältää sarakkeet id, name, created_at, updated_at ja count, id-kenttä toimii yksilöllisenä tunnisteena ja kasvaa automaattisesti jokaisen uuden aihetunnisteen kohdalla. name-kenttä tallentaa aihetunnisteen nimen merkkijonona, ja count-kenttä pitää kirjaa aihetunnisteen käyttökerroista. Kuvassa 2 on esitelty tags-taulu tietokannassa, jossa näkyvät sarakkeet ja niiden arvot.

```
mysql> select * from tags;
+----+-----+-----+-----+-----+
| id | name      | created_at      | updated_at | count |
+----+-----+-----+-----+-----+
| 33 | spring    | 2024-05-25 05:12:40 | NULL      | 562 |
| 34 | summer    | 2024-05-25 05:12:40 | NULL      | 192 |
| 35 | blossom   | 2024-05-25 05:12:40 | NULL      | 275 |
| 36 | sunshine  | 2024-05-25 05:12:40 | NULL      | 798 |
| 37 | picnic    | 2024-05-25 05:12:40 | NULL      | 166 |
| 38 | flowers   | 2024-05-25 05:12:40 | NULL      | 435 |
| 39 | vacation  | 2024-05-25 05:12:40 | NULL      | 675 |
| 40 | warmth    | 2024-05-25 05:12:40 | NULL      | 70 |
| 41 | gardening | 2024-05-25 05:12:40 | NULL      | 323 |
| 42 | outdoors  | 2024-05-25 05:12:40 | NULL      | 405 |
| 43 | beach     | 2024-05-25 05:12:40 | NULL      | 55 |
| 44 | green     | 2024-05-25 05:12:40 | NULL      | 61 |
| 45 | breeze    | 2024-05-25 05:12:40 | NULL      | 140 |
| 46 | hiking    | 2024-05-25 05:12:40 | NULL      | 517 |
| 47 | nature    | 2024-05-25 05:12:40 | NULL      | 163 |
+----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

KUVA 7 Tags-taulu

4.4 API

API-päätepisteitä laajennettiin tukemaan aihetunnisteiden hallintaa. Uudet päätepisteet toteutettiin AdminTagController-kontrollerissa, joka tarjoaa toiminnallisuudet aihetunnisteiden lisäämiseen, päivittämiseen ja hakemiseen.

4.4.1 Index

Index-metodi hakee kymmenen käytetyintä aihetunnistetta tietokannasta ja palauttaa ne JSON-muodossa, mikä mahdollistaa helpon integraation käyttöliittymän kanssa. Tämä toiminnallisuus on keskeinen, jotta käyttäjät voivat nähdä suosituimmat aihetunnisteet ja käyttää niitä helposti tapahtumiensa yhteydessä.

```
public function index(Request $request): JsonResponse
{
    // Fetch top used 10 tags from the database
    $tags = Tag::orderBy('count', 'desc')->limit(10)->get();

    // Return the tags as JSON response
    return response()->json($tags);
}
```

KUVA 8 Index-metodi

4.4.2 Store

Store-metodi vastaanottaa listan aihetunnisteita ja tarkistaa, ovatko ne jo olemassa tietokannassa. Jos aihetunniste on jo olemassa, sen käyttökertoja päivitetään. Jos aihetunniste ei ole olemassa, luodaan uusi aihetunniste. Tämä varmistaa, että aihetunnisteet pysyvät ajan tasalla ja että niiden käyttömäärät ovat tarkkoja.

```
public function store(Request $request): JsonResponse
{
    $tags = $request->input('tags', []);
    foreach ($tags as $tag) {
        // Check if the tag already exists
        $existingTag = Tag::where('name', $tag['name'])->first();
        if ($existingTag) {
            // If the tag exists, increment its count
            $existingTag->increment('count');
        } else {
            // If the tag doesn't exist, create a new one
            Tag::create(['name' => $tag['name'], 'count' => 1]);
        }
    }
}
```

KUVA 9 Store metodi

4.4.3 React hook

Käyttöliittymässä käytetään custom hookia aihetunnisteiden hakemiseen palvelimelta ja niiden hallintaan. Kuvassa 6 oleva hook, useTags, hyödyntää react-query-kirjastoa tietojen noutamiseen ja välimuistiin tallentamiseen. fetchTags-funktio käyttää apiClient.get-metodia noutaakseen aihetunnisteet /tags-päätepiisteestä, tarkistaa, että vastaus ei ole tyhjä, ja palauttaa sitten saadut tiedot.

```
const fetchTags = async () => {
  try {
    const res = await apiClient.get('/tags');
    if (!res || !Array.isArray(res)) {
      throw new Error('Response data is empty');
    }
    return res;
  } catch (error) {
    console.error('Error fetching tags:', error);
    throw error;
  }
};

interface Tag {
  id: number;
  name: string;
}

export default function useTags(): {
  tags: Tag[];
  isLoading: boolean;
  isFetching: boolean;
  isError: boolean;
} {
  const { data: tags = [], isLoading, isFetching, isError } = useQuery(['tags'], async () => await fetchTags());
  return {
    tags,
    isLoading,
    isFetching,
    isError,
  };
}
```

Kuva 10 React hook

4.5 Testaus

Seuraavassa osiossa kuvataan, miten aiheutunnisteiden lisäämisominaisuuden testaaminen suoritettiin. Testausprosessi voidaan jakaa useisiin vaiheisiin, jotka kattavat eri testausmenetelmät ja -tekniikat. Tässä työssä testaus tehtiin pääasiassa PHPUnit-testien avulla sekä käyttöliittymätestauksella.

4.5.1 HTTP-metodit

PHPUnit-testien avulla testattiin, että http-metodit toimivat oikein Hubi-sovelluksessa. Testit varmistivat, että kaikki HTTP-metodit palauttavat odotetut tilakoodit ja suorittavat odotetut toiminnot. Testit kattavat kaikki olennaiset HTTP-metodit ja varmistavat, että sovellus käsittelee eri pyyntöjä oikein. Kuvassa 11 on ajettu http-testi onnistuneesti läpi.

HTTP-metodit ovat:

- **GET**: Hakee resursseja palvelimelta. Tämä metodi ei muuta palvelimen tilaa ja sitä käytetään yleensä tietojen hakemiseen.
- **POST**: Lähettää dataa palvelimelle resurssin luomiseksi tai päivittämiseksi. Tämä metodi voi muuttaa palvelimen tilaa, kuten lisätä uuden tietueen tietokantaan.
- **PUT**: Päivittää olemassa olevan resurssin palvelimella kokonaisuudessaan. Se korvaa olemassa olevan resurssin lähettämällä uuden version siitä.
- **PATCH**: Päivittää osittain olemassa olevan resurssin palvelimella. Tämä metodi tekee muutoksia tiettyihin resurssin osiin sen sijaan, että korvaisi koko resurssin.
- **DELETE**: Poistaa olemassa olevan resurssin palvelimelta.

(Http request methods n.d.)

```
PASS Tests\Feature\Controllers\Admin\AdminTagControllerTest
✓ store
✓ update
✓ destroy
✓ fail store if not logged in
✓ fail update if not logged in
✓ fail destroy if not logged in
✓ fail store if not admin
✓ fail update if not admin
✓ fail destroy if not admin
✓ show
✓ index
✓ fail show if not logged in
✓ fail index if not logged in
✓ fail show if not admin
✓ fail index if not admin

Tests: 15 passed
Time: 27.00s
```

Kuva 11 http-metodien testaus

4.5.2 CRUD testaus

CRUD on lyhenne sanoista Create, Read, Update ja Delete, jotka ovat neljä perustavanlaatuaista toimintoa tietokantojen ja muiden tallennusjärjestelmien hallinnassa. (What is CRUD? n.d.)

CRUD-testauksessa varmistetaan, että nämä toiminnot toimivat oikein sovelluksessa. PHPUnit-testien avulla testattiin CRUD-metodit Hubi-sovelluksessa. Testit varmistivat, että sovellus pystyy luotettavasti suorittamaan CRUD-toiminnot tietokantaan. Kuvassa 12 on kuvakaappaus suoritetuista testeistä

```
PASS Tests\Feature\Models\TagTest
✓ create tag
✓ read tag
✓ update tag
✓ delete tag

Tests: 4 passed
Time: 15.79s
```

Kuva 12 CRUD-testaus

4.5.3 Käyttöliittymätestaus

Hubi-sovelluksen käyttöliittymätestaus suoritettiin manuaalisesti käyttäen tutkivan testauksen menetelmää (Exploratory Testing 2024).

Käyttöliittymätestauksessa käytiin huolellisesti läpi kaikki tehdyt komponentit yksi kerrallaan ja varmistettiin, että ne eivät aiheuta virheilmoituksia. Vaikka virheilmoituksia ei ilmennyt ja huomattiin, että vastausaikoja voisi vielä parantaa. Erityisesti aihetunnisteiden tuonti käyttöliittymään oli paikoin hidasta.

Testauksessa varmistettiin myös, että kaikki painikkeet ja muut käyttöliittymäelementit toimivat oikein ja suorittavat odotetut toiminnot. Lisäksi tarkasteltiin käytettävyyttä varmistaen, että käyttöliittymä on selkeä ja intuitiivinen käyttää.

4.6 Yhteenveto käytännön toteutuksesta

Aihetunnisteiden lisääminen Hubi-sovellukseen on onnistunut hyvin, ja lopputulos luo vakaan pohjan jatkokehitykselle, jossa aihetunnisteita voidaan hyödyntää entistä enemmän. Käyttöliittymään tehtyjen muutosten toteutus sujui sujuvasti ja integroitiin saumattomasti jo olemassa olevien komponenttien kanssa.

Tapahtumanjärjestäjille luotiin mahdollisuus lisätä itse aihetunnisteita tapahtumiinsa. Käyttäjä syöttää uuden aihetunnisteen tekstikenttään ja painaa "Lisää"-painiketta tai Enter-näppäintä, minkä jälkeen aihetunniste lisätään tapahtuman aihetunnisteiden listaan.

Suosittu aihe-tunnisteet tarjoavat käyttäjille hyvän käsityksen siitä, mitä aihe-tunnisteita muut käyttäjät käyttävät. Tämä ominaisuus voidaan myös laajentaa siten, että sovellus tarjoaa valmiiksi luotuja suosittuja aihe-tunnisteita esimerkiksi vuodenaikojen mukaan. Tämä lisää joustavuutta ja auttaa käyttäjiä löytämään sopivia aihe-tunnisteita tapahtumilleen, mikä parantaa tapahtumien näkyvyyttä ja löydettävyyttä.

Kaiken kaikkiaan aihe-tunnisteiden lisääminen ja suosittujen aihe-tunnisteiden hakutoiminnallisuus ovat merkittäviä parannuksia Hubi-sovelluksessa. Ne tarjoavat käyttäjille uusia työkaluja tapahtumien hallintaan ja helpottaa käytettävyyttä.

4.7 Jatkokehitys

Aihe-tunnisteiden lisääminen Hubi-sovellukseen on tarjonnut vankan perustan tapahtumien luokittelulle ja löydettävyydelle. Vaikka nykyinen ratkaisu toimii hyvin, on useita mahdollisuuksia jatkokehitykselle, jotka voivat entisestään parantaa sovelluksen toiminnallisuutta ja käyttäjäkokemusta.

Yksi merkittävä jatkokehityskohde on aihe-tunnisteiden laajempi hyödyntäminen sovelluksen käyttöliittymässä. Tällä hetkellä toiminnallisuus on rajattu vain tapahtumajärjestäjille, eikä käyttäjille ole vielä implementoitu ominaisuuksia. Seuraava askel on luoda käyttäjille kohdennettua sisältöä tagien perusteella, mikä parantaa heidän kokemustaan ja tekee tapahtumien löytämisestä helpompaa.

Aihe-tunnisteiden hallinnan jatkokehitys Hubi-sovelluksessa tarjoaa monia mahdollisuuksia sovelluksen parantamiseen. Tähän sisältyy analytiikan integrointi, joka voisi tarjota tapahtumanjärjestäjille tietoa aihe-tunnisteiden tehokkuudesta. Lisäksi kausiluonteisten ja paikallisten aihe-tunnisteiden hyödyntäminen auttaisi tapahtumia tavoittamaan kohdeyleisönsä paremmin.

Monissa tapahtumasovelluksissa aihe-tunnisteiden luominen ja hallinta voi olla osittain tai täysin automatisoituja. Sovellukset voivat käyttää tapahtuman otsik-

koa, sijaintia, kategoriaa ja ajankohtaa automaattiseen aihetunnisteiden generointiin. Esimerkiksi, jos tapahtuman otsikko sisältää sanan "konsertti", sovellus voi automaattisesti lisätä aihetunnisteen "konsertti". Tämä vähentää tapahtumanjärjestäjien työtaakkaa ja varmistaa, että tapahtumat on aina merkitty oikein ja ovat helposti löydettävissä.

5 POHDINTA

Aihetunnisteet ovat luontainen osa modernia digitaalista ympäristöä, ja niiden hyödyntäminen parantaa sovelluksen käytettävyyttä merkittävästi. Aihetunnisteiden avulla tapahtumat voidaan kategorisoida tehokkaasti, mikä auttaa käyttäjiä löytämään nopeasti heitä kiinnostavat tapahtumat. Tämä parantaa käyttäjäkokemusta ja tekee sovelluksesta intuitiivisemmän käyttää.

Opinnäytetyön keskeiset tavoitteet saavutettiin onnistuneesti. Uuden aihetunnisteominaisuuden integrointi sujui suunnitelmien mukaisesti. Projektin aikana kehitetyt komponentit ja taustajärjestelmään tehdyt muutokset toteutettiin ilman suurempia ongelmia, ja ne toimivat odotetusti. Testaus osoitti, että uusi ominaisuus on käyttökelpoinen.

Jatkokehitykselle nousi esiin paljon uusia mahdollisuuksia, jotka voivat tehdä aihetunnisteista entistä tehokkaampia ja monipuolisempia. Esimerkiksi analytiikan hyödyntäminen aihetunnisteiden käytön seuraamisessa ja automaattisen aihetunnisteiden generoinnin lisääminen voisivat tuoda merkittäviä parannuksia.

Kokonaisuudessaan projekti oli onnistunut, mutta paljon ideoita ja mahdollisuuksia jäi vielä testattavaksi. Mahdollisuuksia aihetunnisteiden käytön laajentamiseen ja parantamiseen on runsaasti, ja nämä tarjoavat hyvät lähtökohdat tuleville kehitysprojekteille. Aihetunnisteiden käyttö Hubi-sovelluksessa luo vankan perustan, jonka päälle voidaan rakentaa entistä parempia ja monipuolisempia toiminnallisuuksia, jotka tekevät sovelluksesta käyttäjäystävällisemmän ja toimivamman kokonaisuuden.

LÄHTEET

Codecademy. n.d. What is CRUD. Verkkosivu. Viitattu 30.5.2024.
<https://www.codecademy.com/article/what-is-crud>

Geeksforgeeks. 2024. Exploratory Testing. Verkkosivu. Viitattu 30.5.2024.
<https://www.geeksforgeeks.org/exploratory-testing/>

GeoPass Company Oy. n.d. Hubiapp kotisivut. Verkkosivu. Viitattu 15.5.2024.
<https://hubiapp.com/fi>

Goodwin, M. 2024. What is an API? IBM. Luettu 10.5.2024.
<https://www.ibm.com/topics/api>

Javatpoint. n.d. MVC Architecture. Verkkosivu. Viitattu 21.5.2024.
<https://www.javatpoint.com/php-mvc-architecture>

Mdn web docs. n.d. HTTP request methods. Verkkosivu. Viitattu 30.5.2024.
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

MySQL. n.d. MySQL kotisivut. Verkkosivu. Viitattu 10.5.2024.
<https://www.mysql.com/>

NEXT.js. n.d. NEXT.js kotisivut. Verkkosivu. Viitattu 10.5.2024.
<https://nextjs.org/>

PHP. n.d. PHP kotisivut. Verkkosivu. Viitattu 10.5.2024.
<https://www.php.net/>

React. n.d. React kotisivut. Verkkosivu. Viitattu 30.5.2024.
<https://react.dev/>

Sanastokeskus n.d. Tietotekniikan termitalkoot. Verkkosivu. Viitattu 30.5.2024.
https://sanastokeskus.fi/tsk/fi/termitalkoot/hakemistot-267.html?page=get_id&id=ID263&vocabulary_code=TSKTT

Tanstack. n.d. Queries. Verkkosivu. Viitattu 30.5.2024.
<https://tanstack.com/query/latest/docs/framework/react/guides/queries>