Tampere University of Applied Sciences

Verkkofoorumi

React-sovellus

Daniil Kovalev

OPINNÄYTETYÖ
Toukokuu 2024

Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

KOVALEV, DANIIL:
Verkkofoorumi
React-sovellus

Opinnäytetyö 21 sivua, liitteitä 0 sivua
Toukokuu 2024

Opinnäytetyön tavoitteena oli luoda React-pohjainen foorumisivusto lisättäväksi tekijän portfolioon. Työ sisältää tausta-, käyttöliittymä- ja tietokantakomponentteja fullstack-sovellukselle.

Opinnäytetyössä käytettiin React-, node-, express-, MySQL2-, JWT-, MUI- ja bcrypt-kirjastoja. Foorumisivusto rakennettiin siten, että kuka tahansa käyttäjä, jolla on tai ei ole tiliä, voi tehdä hakuja mistä tahansa aiheesta sekä lukea vastauksia luotuihin aiheisiin. Kirjautuessaan sisään tai luotuaan tilin käyttäjät voivat kuitenkin luoda uusia aiheita, jotka näkyvät sivutuilla aihesivuilla kaikille, jättää vastauksia valitsemiinsa aiheisiin, tykätä tai poistaa antamiaan tykkäyksiä mistä tahansa vastauksesta sekä muokata omia vastauksiaan.

Työn tuloksena on toimiva sovellus, joka täyttää tuotekehityskelpoisuuden vähimmäisvaatimukset. Jatkokehitysehdotuksena sovellusta voisi laajentaa lisäämällä siihen esimerkiksi raporttitoiminto, käyttäjän mahdollisuus vaihtaa tai nollata salasana sekä profiilikuvan vaihtaminen pois paikkamerkkikuvasta.

Asiasanat: react, foorumisivusto, fullstack

ABSTRACT

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

KOVALEV, DANIIL:
Web Forum
React App

Bachelor's thesis 21 pages, appendices 0 pages
May 2024

The object of this thesis was to create a React framework based forum webpage to expand the creator’s portfolio. The thesis project was done by a single person over five months and contained the requisite backend, frontend and database for a full stack application.

The thesis was carried out using React, node, express, MySQL2, JWT, MUI and bcrypt libraries to build the application’s functionality. The thesis was created in such a way that any user, regardless of whether they owned an account, could search for, and interact with topics of their choosing, and read the replies to any chosen topic. Upon authentication or account creation however, the application allowed the user to create topics of their own, as well as reply to topics, both like and remove their given likes from any replies of their choosing and edit their own replies to topics.

The application could be expanded to include reporting functionality, as well as give the users the ability to change their passwords, delete their accounts, or change their profile pictures to something other than the placeholder image.

Key words: react, forum, fullstack

SISÄLLYS

LYHENTEET JA TERMIT

TAMK	Tampereen ammattikorkeakoulu
op	opintopiste
backend	tietojen käyttökerros ohjelmistoarkkitehtuurissa
frontend	verkkosivuston graafinen käyttöliittymä
db	tietokanta
React	JavaScript kirjasto
JS	JavaScript ohjelmointi kieli
CSS	tyylisivujen kieli
SQL	vakiokieli tietokannan luomiseen ja käsittelyyn
JWT	JSON Web Token
MUI	Material UI

1 JOHDANTO

Idea verkkopohjaisista keskustelufoorumeista sai alkunsa Usenet-järjestelmää käyttävistä uutisryhmistä. Vuonna 1979 kehitetty Usenet toimi ilmoitustaulujärjestelmänä, ja UNIX-koneet tukivat sitä. Teknologian kehittyessä keskustelufoorumit kehitettiin toimimaan verkossa UNIX-pohjaisen järjestelmän sijaan.

Nykyään verkkofoorumi on suosittu ja laajalti käytetty sovellusmuoto, jota käytetään useista syistä keskustelupalstasta, kuten nimestä voi päätellä, tapaan, jolla ihmiset jakavat taide- tai videopelistrategioita Internetissä ja sitten valmiustilassa validointia varten. Verkkofoorumit ovat tavallaan yhtä paljon sosiaalista mediaa kuin esimerkiksi instagram tai x. (Formerly twitter)

Verkkofoorumi on loistava tapa pitää yhteyttä, ja vaikka WarThunder-foorumi (WarThunder forum) saa jatkuvasti salattuja armeijan tiedostovuotoja, koska ihmiset tappelevat jatkuvasti WarThunder-verkkopelin (WarThunder-verkkopeli) vääristyneistä teknisistä tiedoista, verkkofoorumi on loistava tapa seurustella ja seurustella. muodostaa ystävyyssuhteita.

Verkkofoorumi palvelee myös heitä, jotka osallistuvat teknisiin keskusteluihin tai esimerkiksi roolipeleihin. Esimerkkeinä ohjelmoijien Stack Overflow (Stackoverflow forum) tai SpaceBattles (SpaceBattles forum), jossa on keskustelupalstoja politiikasta mediaan, luovia keskustelupalstoja fanifiktiosta ja alkuperäisistä romaaneista taiteeseen, sekä Questit, jotka ovat pohjimmiltaan vanhan koulun tekstipohjaisia seikkailupelejä, joita ajavat ja kirjoittavat oikeat ihmiset pelaajien demokraattisten äänten perusteella.

Lyhyesti sanottuna verkkofoorumit ovat sosiaalisen median niche-muoto, joka palvelee erityisiä markkinarakotapauksia käyttäjän tarpeiden mukaan. Kaikki verkkofoorumit eivät ole samanlaisia, ja jotkin ovat joko keskittyneet erikoisesti omaan markkinarakoonsa tai ovat yleisempiä monipuolisempaa käyttöä varten.

Projektin aiheena valittiin verkkofoorumi siksi, että tunnen verkkofoorumit ja halusin nähdä, kuinka paljon työtä vaatisi saada alkeellinen perusmalli verkkofoorumista, joka toimii itse osoituksena taidoistani lisättäväksi portfoliooni.

Sellaisenaan tämä raportti sisältää tietoa tietokannan, taustajärjestelmän ja käyttöliittymän rakenteesta, projektissa käytetyt teknologiat, kuten esimerkiksi express, jota käytettiin sovelluksen backendin rakentamiseen sekä siitä, miltä projekti näyttää käyttäjälle.

2 TAVOITE JA TARKOITUS

Tämän opinnäytetyön tavoitteena oli saada aikaiseksi react pohjainen fullstack verkkoforumi. Opinnäytetyössä on myös käytetty node.js ja express kirjastoja backend kehitykseen, material UI kirjastoa frontend kehitykseen ja muita kirjastoja, kuten JWT kirjasto backend:in salasanan salaukseen ja selainten sessio valvontaan.

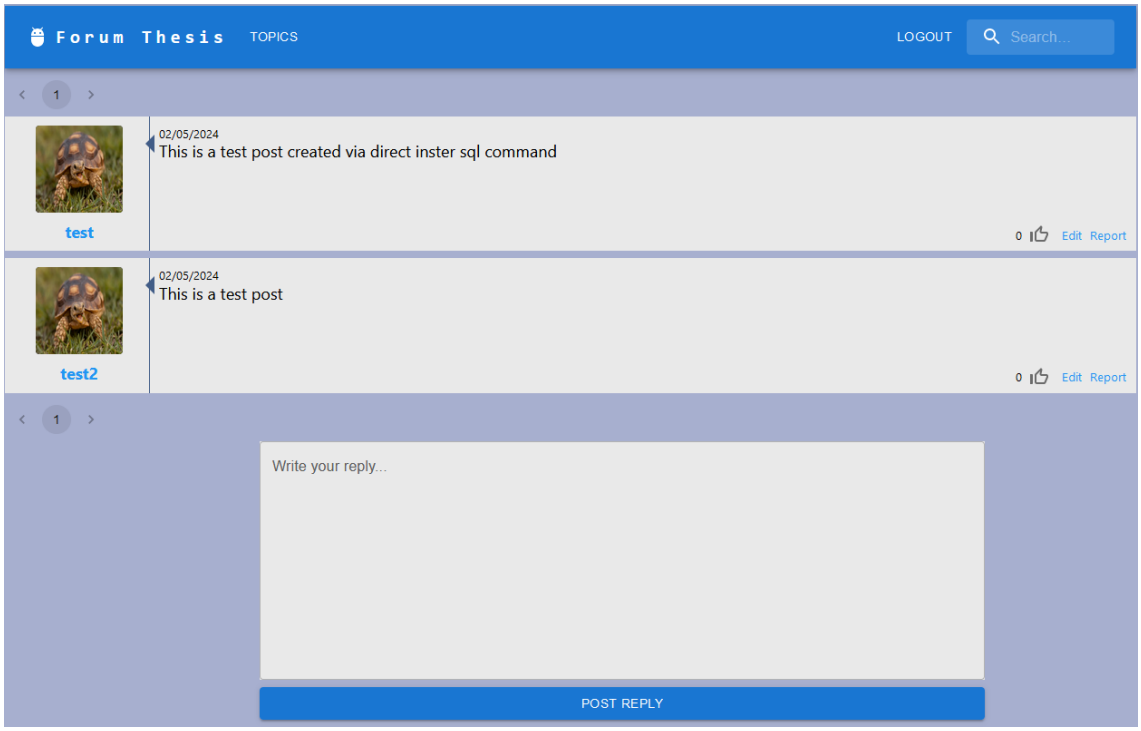
Tämä tavoite oli toteutettu jakamalla projekti backend, frontend ja tietokanta osiin. Backend osuus koostuu kaikista serveri koodista ja mysql tietokannan koodista. Frontend osuus koostuu käyttöliittymäkomponenteista, jotka käyttävät backend komponentteja react hookien kautta antamaan staattiselle sivulle toiminnan.

Projektin tietokanta on jaettu kolmeen osaan:

- ”users”-taulukko, joka sisältää käyttäjien tiedot
- ”posts”-taulukko, joka aloittaa forumin ketjut
- ”topics”-taulukko, joka sisältää forumin aiheet

Projekti käyttää ”topics” ”-taulukkoa pääsivulla otsikoina, joita käyttäjät voi painaa päästääkseen lukemaan kesusteluja ja vastaamaan niihin jätettyihin viesteihin.

Kuva 1. Kotisivu



Kuva 2. Valitun aiheen näkymä.

3 TEKNOLOGIAT

3.1 Backend

3.1.1 Node.js

Projekti käyttää Nodea, joka on monialustainen, avoimen lähdekoodin JavaScript-ajoympäristö, joka voi toimia Windowsissa, Linuxissa, Unixissa, macOS:ssä ja muissa käyttöjärjestelmissä JavaScript-koodin suorittamiseen verkkoselaimen ulkopuolella, mikä mahdollistaa JavaScriptin kirjoittamisen komentorivityökalujen ja -työkalujen kirjoittamiseen. palvelinpuolen komentosarjaan. Node.js toimii V8 JavaScript -moottorilla ja suorittaa JavaScript-koodin verkkoselaimen ulkopuolella. (NodeJS documentation)

3.1.2 Express

Express.js tai yksinkertaisesti Express on taustaverkkosovelluskehys RESTful-sovellusliittymien rakentamiseen Node.js:n avulla. Se on julkaistu ilmaisena avoimen lähdekoodin ohjelmistona MIT-lisenssillä. Se on suunniteltu verkkosovellusten ja API:iden rakentamiseen. Sitä on kutsuttu Node.js:n de facto vakiopalvelinkehykseksi. (Express homepage)

Projektissa Expressiä käytetään kevyenä ja joustavana reitityskehyksenä, jota on tarkoitus täydentää Express-väliohjelmistomoduuleilla. Tämä mahdollistaa tausta-API:n luomisen, jota käytetään sovelluksen käyttöliittymässä.

3.1.3 MySQL2

MySQL on avoimen lähdekoodin relaatiotietokannan hallintajärjestelmä ja MySQL2 on jatkoa MySQL-Nativelle, joka itsessään korvaa MySQL Client kirjaston. MySQL2-tiimi työskentelee yhdessä Node MySQL -tiimin kanssa jaetun koodin erottamiseksi ja siirtämiseksi mysqljs-organisaation alle.

Protokollan jäsentäjäkoodi kirjoitettiin uudelleen alusta ja api muutettiin vastaamaan suosittua Node MySQL:ää. MySQL2 on enimmäkseen API-yhteensopiva Node MySQL:n kanssa ja tukee useimpia ominaisuuksia. Kuten esim:

- Nopeampi / parempi suorituskyky
- Laajennettu tuki koodaukselle ja lajittelulle
- Valmistetut lausunnot
- Poolaaminen

(MySQL2 repository)

3.1.4 Bcrypt

Bcrypt on kryptografinen hajautustoiminto, joka on suunniteltu salasanojen hajauttamiseen ja turvalliseen tallentamiseen sovellusten taustajärjestelmään tavalla, joka on vähemmän herkkä sanakirjapohjaisille kyberhyökkäyksille. Sen loivat vuonna 1999 Niels Provos ja David Mazières käyttämällä Blowfishin salausalgoritmia perustanaan. (Provos, Honeyman ja Mazières, 1999)

Sovelluksessa bcryptiä käytettiin parantamaan salasanan tiivistystä ja vertaamaan tiivistettyä salasanaa tietokantaan salasanan syöttämiseksi, kun käyttäjä kirjautuu sisään. Sen lisäksi, että bcrypt sisältää suolan (satunnaisdata, joka syötetään lisäsyötteenä yksisuuntaiseen toimintoon, joka tiivistää tietoja) suojatakseen sateenkaaritaulukko (esilaskettu taulukko kryptografisen hajautusfunktion tulosten tallentamiseen välimuistiin) hyökkäyksiä vastaan, bcrypt on mukautuva toiminto: ajan mittaan iteraatioiden määrää voidaan lisätä hidastaakseen sitä, joten se kestää raa'an voiman etsintähyökkäyksiä jopa laskentatehon kasvaessa. (Bcrypt repository)

3.1.5 JSONwebtoken

JSON-verkkotunnukset (JWT) ovat standardoitu tapa lähettää tietoja turvallisesti kahden osapuolen välillä. Ne sisältävät JSON-muotoon koodattuja tietoja (vaatimuksia). Nämä väitteet auttavat jakamaan tiettyjä tietoja osapuolten

kesken. JWT on pohjimmiltaan mekanismi joidenkin JSON-tietojen aitouden tarkistamiseksi.

Sovellus käyttää jsonwebtokenia tietojen turvalliseen siirtämiseen osapuolten välillä JSON-objektina, mikä helpottaa tiedonsiirtoa tausta- ja käyttöliittymän välillä todennusta ja istuntotunnisteen luomista varten. Tunnus luodaan käyttämällä yksityistä salaisuutta tai julkista/yksityistä avainta. JWT luottaa muihin JSON-pohjaisiin standardeihin: JSON Web Signature ja JSON Web Encryption. (JSONwebToken repository)

3.2 Frontend

3.2.1 React

React on ilmainen ja avoimen lähdekoodin käyttöliittymän JavaScript-kirjasto komponentteihin perustuvien käyttöliittymien rakentamiseen. Sitä ylläpitävät Meta ja yksittäisten kehittäjien ja yritysten yhteisö. Reactilla voidaan kehittää yksisivuisia, mobiili- tai palvelimella renderöityjä sovelluksia kehyksillä, kuten Next.js. (React homepage)

Sovelluksen käyttöliittymä luotiin käyttämällä React-kehystä, koska se sisältää paljon sisäänrakennettuja kirjastoja, jotka ovat hyödyllisiä sovelluksen rakentamisessa. Yksi niistä on sisäänrakennettu react-queries -kirjasto, joka mahdollistaa API-kutsut.

Muita käytettyjä reaktiokirjastoja olivat React-router-dom -kirjasto, joka mahdollistaa reitityksen sovelluksen käyttöliittymässä, sekä perus React-kirjasto, joka mahdollistaa useState-, useCallback-, useEffect- ja muut vastaavat toiminnot tietojen muokkaamisen tai hakemisen helpottamiseksi.

3.2.2 Material UI (MUI)

Material UI on avoimen lähdekoodin React-komponenttikirjasto, joka toteuttaa Googlen materiaalisuunnittelun. Se on kattava ja sitä voidaan käyttää tuotannossa heti. MUI on React-ekosysteemin suurin käyttöliittymäyhteisö. Se on melkein yhtä vanha kuin itse React, ja se alkoi materiaalisuunnitteluohjeiden React-toteutuksena vuonna 2014. (Material UI documentation)

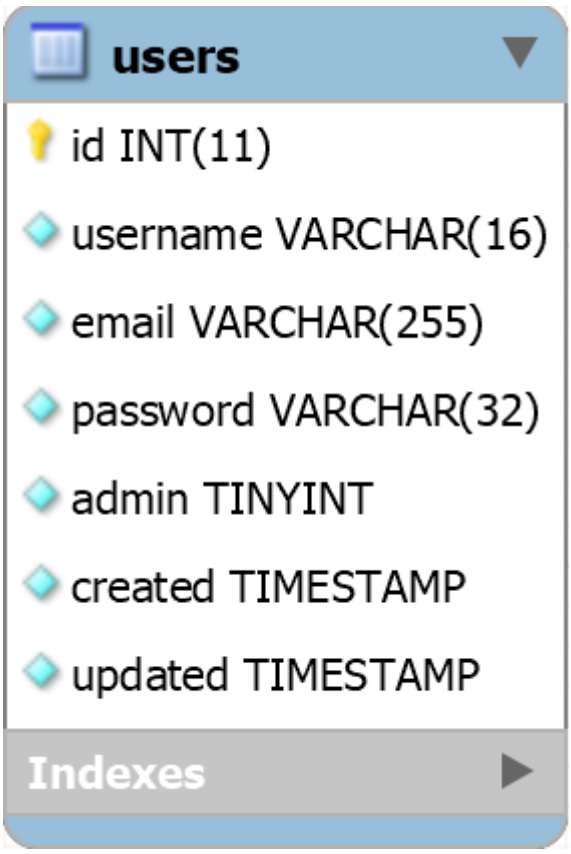
4 TOTEUTUS

4.1 Tietokanta

Projektin tietokanta on luotu mysql2:lla ja sisältää käyttäjätietoja, käyttäjien ja ylläpitäjien luomia aiheita ja niihin liittyviä keskustelupalstoja.

4.1.1 Users

Tietokanta oli jaettu muutamaan taulukkoon, josta ”users” on yksi. Users taulukon rakenne siältää pääavaimen ”id”, käyttäjien nimet ”username”, käyttäjien sähköpostit ”email”, käyttäjien salasanat ”password”, käyttäjien mahdolliset järjestelmänvalvojan oikeudet ”admin” ja käyttäjän luomisen ja muokkaamisen ajat ”created” ja ”updated” kenttiä.



KUVA 3. Users taulukon rakenne.

Oletusarvoisesti mikään taulukon sarakkeista ei voi olla tyhjä. Pääavain "id" automaattisesti kasvaa jokaisen uuden käyttäjän kanssa, käyttäjän tiedot on perus muuttuvia merkkikenttiä, järjestelmänvalvojan oikeudet on yksinkertainen boolean, joka tarkistetaab backend:in puolella ja tilin luonti- ja muokkausajat on nykyistä aikaleimaa.

4.1.2 Posts

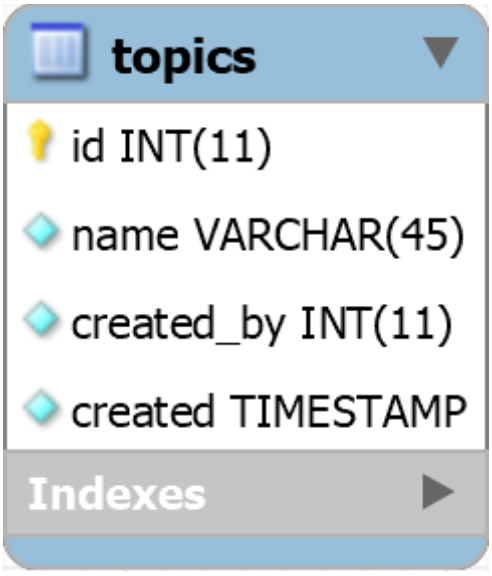
Tietokannan toinen taulukko, on ”posts”. ”Posts” taulukon rakenne siältää pääavaimen ”id”, aiheen id ”topic_id”, viestin luojan käyttäjä nimi ”created_by”, viestin teksti sisältö ”body”, viestin tykkäykset ”likes” ja viestin luomisen ja muokkaamisen ajat ”created” ja ”updated” kenttiä.

KUVA 4. Posts taulukon rakenne.

Samoin kuin käyttäjät taulukollakin oletusarvoisesti viestit taulukolla mikään sarakkeista ei voi olla tyhjä. Pääavain "id" automaattisesti kasvaa jokaisen uuden käyttäjän kanssa, viestin luojan käyttäjä nimi on perus muuttuva merkkikenttä, viestin teksti sisältö on ”text” kenttää, sitten on JSON kenttä joka sisältöö kaikkien käyttäjien id:t jota ovat tykkännyt viestin ja viestien luonti- ja muokkausajat on nykyistä aikaleimaa.

4.1.3 Topics

Tietokanta sisältää "topics"-taulukon, jota käytetään pääasiassa otsikoille varsinaisissa keskustelupalstoissa ja otsikkona aihesivulla, jota käyttäjät voivat painaa saadakseen uudelleenohjauksen valittuun aiheeseen liittyvään keskustelupalstaan. "Topics"-taulukko sisältää, numeerisen id:n, merkikkikentän nimi, aiheen luojan numeerinen id ja ajan jolloin aihe oli luotu.



KUVA 5. Topics taulukon rakenne.

4.2 Backend

Projektin backend hoitaa foorumisivuston toiminnan kannalta olennaiset palvelinpuolen tehtävät, kuten esimerkiksi toiminnot käyttäjien luomien tilitietojen lähettämiseksi sql-tietokantaan.

Itse taustaohjelma on jaettu muutamaan osaan, app.js-tiedostoon, joka käsittelee koko taustajärjestelmän expressin kautta sovelluksen käyttämien reittien käsittelemiseksi. Nämä reitit määritellään vastaavissa reitittimissä "routes"-kansiossa, jotka saavat toimintoja ohjaimista, jotta reitit todella toimii. “Controllers”-kansiosta löytyvät ohjaintiedostot käyttävät malleja, jotka löytyvät “models”-kansiosta, joissa on sql-komentoja tietojen lähettämiseksi tietokantaan.

Backend sisältää myös "db"-kansion, jossa on pool.js joka sisältää tarvittavat mysql-poolitiedot kätevästi yhdessä paikassa, joten se voidaan tuoda kaikkiin sitä tarvitseviin tiedostoihin sen sijaan, että se tehdään uudelleen jokaisessa tiedostossa.

4.2.1 Mallit

Mallit-kansio sisältää moduuleja, joissa käytetään SQL-komentoja tietokannan yhdistämiseksi projektiin. Kaikki komennot noudattavat yksinkertaista kaavaa. Esimerkkinä "findAll" funktio: Funktio käyttää Promise rakentajaa, joka saa yhteyden tietokantaan poolin kautta, joka sitten sisältää kaikki asiaankuuluvat tietokantatiedot. Kun yhteys on muodostettu, SQL-komento, tässä tapauksessa yksinkertainen SELECT * FROM table, lähettää tietokantaan query()-funktion kautta SQL komennon. Kyseessä oleva "table" on luonnollisesti kyseessä olevaan malliin liittyvä taulukko. Posts mallilla se on posts taulukko, topics mallilla se on topics taulukko ja users mallilla users taulukko.

Esitetty esimerkki on hieman yksinkertainen, koska se on yleinen hae kaikki tyyppinen sql, jossa ei ole muuttujia taulukon nimen lisäksi. Muut koodin sql-komennot ovat monimutkaisempia ja vaativat vähintään yhden tietopisteen määrittämään mitä haetaan tai mitä tietokantaan lisätään. Edellisen esimerkin mukaisesti käytämme toista posts taulukon funktiota, ja tällä kertaa se sisältää komennon tietojen lisäämiseksi tietokantaan.

Posts.js-mallitiedostossa on "save"-toiminto. Se toimii samalla tavalla kuin "findAll"-toiminto paitsi, että tämä ottaa tietojoukon ja lähettää sen tietokantaan

"INSERT INTO posts SET ?" sql-komennon avulla. Projektin mallikoodin komennoissa käytetään valmiita lausekkeita parametroiduilla kyselyillä, kuten alla olevan koodissa näkyvä kysymysmerkki osoittaa. Tämä tehdään, koska tietokanta erottaa aina koodin ja tiedon riippumatta siitä mitä käyttäjä syöttää. Valmistetut lausunnot varmistavat myös, että hyökkääjä ei pysty muuttamaan komennon tarkoitusta, vaikka hyökkääjä olisi lisännyt SQL-komentoja.

```
save: (post) => new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
      if (err)
        return reject(err);

      const createQuery = 'INSERT INTO posts SET ?';
      connection.query(createQuery, post, (err, result) => {
        connection.release();
        if (err) {
          return reject(err);
        }
        resolve(result);
      });
    });
  })
```

Kaikki mallit on luotu käyttämällä tätä perussuunnitelmaa. Uusi lupaus, jossa on yhteyspooli, joka lähettää kyselyn mainitun yhteyden kautta ja palauttaa sitten tuloksen lupauksen ratkaisun kautta. Itse SQL komento vaihtuu per tarvittu toiminta, mutta ne kaikki toimii tätä muotoa.

4.2.2 Ohjaimet

Projektiohjaimet käyttävät malleja sql-toiminnallisuuteen ja itse ovat melko yksinkertaisia toteuttaa. Jatkamme posts taulukon käyttöä myös ohjaimille. Posts malleissa on yhdenksän ohjainta, sisältäen ohjaimia:
- viestien datan hakuun (getPosts, getPostsById, getPostsByTopicId, getPostCount)
- viestien luomiseen (createPost)

- viestien poistamiseen (deletePost, deletePostByTopicId)
- viestien muokkaamiseen. (editPost, addLike)

4.2.3 Reitit

Projektin reitit toimii express kirjaston ”router” toiminnallisuuden kautta. Nämä reitit on tehty yksinkertaisen mallin mukaan: kaikki reitit käyttää router:in get, patch, post ja delete omaisuuksia, jotka sisältää polun, ja ohjaimen.

```
router.get('/', getPosts);
router.get('/:id', getPostsById);
router.get('/byTopic/:id', getPostsByTopicId);
router.get('/countByTopic/:id', getPostCount);

router.use(verifyToken);

router.post('/create', createPost);
router.patch('/edit/:id', editPost);
router.patch('/like/:id', addLike);
router.delete('/delete/:id', deletePostById);
router.delete('/del-by-topic/:id', deletePostByTopicId)
```

Projektin ”topics” ja ”posts” reitit käyttävät verifyToken väliohjelmostoa, joka tarkistaa, että käyttäjä on kirjautunut sisään.

4.2.4 Väliohjelmisto

Projektin taustalla on verifyToken väliohjelmisto, joka nimensä mukaan tarkistaa, että taustaohjelmassa vastaanotettu tunnus vastaa sisäänkirjautumisen yhteydessä luotua JWT-salattua tunnusta.

4.2.5 Mysqlpool

MySQL-yhteyspooli toimii asiakaspuolella varmistaakseen, että käyttäjä ei jatkuvasti muodosta yhteyttä MySQL-palvelimeen tai katkaisee yhteyttä siitä. Sitä mukaan projektissa käytetty MySQL pool ottaa tietokanna tietoja joiden avulla luodaan yhteys projektin tasuta ja tietokanta luo yhteyden.

4.3 Frontend

Projektin käyttöliittymä on rakennettu React-kehystyöllä, mikä mahdollistaa reaktiivisen lopputuotteen käyttöliittymään.

4.3.1 Navigointi

Projektin käyttöliittymän navigointi tapahtuu React-navigointi- ja reittikomponenteilla. Itse navigointi tapahtuu app.js-tiedostossa, joka koostuu BrowserRouter-komponentista ja Routes-komponentista. BrowserRouter tallentaa nykyisen sijainnin selaimen osoitepalkkiin puhtaiden URL-osoitteiden avulla ja navigoi selaimen sisäänrakennetun historiapinon avulla. Reitit-komponentti renderöi käyttöliittymän, jos sijainti vastaa nykyistä reittipolkua, jotka on tallennettu alareitteihin. Nämä on kääritty QueryClientProvideriin, joka käyttää React Contextia QueryClientin jakamiseen koko sovelluksessa.

4.3.2 Navigointipalkki

Projektin navigointipalkki oli luotu Material UI -kirjastolla ja sisältää projektin nimen, jota napsautettuna ohjaa kotisivulle, sisään-/uloskirjautumispainikkeen, joka muuttuu sen mukaan, onko käyttäjä kirjattu sisään vai ei ja ohjaa todennussivulle tai kirjaa käyttäjän ulos, jos hän on kirjautunut sisään, sekä hakupalkin, joka vastaanottaa tekstinsyötön ja etsii tietokannasta aiheita, jotka sisältävät tekstikenttään kirjoitetut sanat.

Kuva 6. Palkki

4.3.3 Kotisivu

Projektin kotisivu on Topis-sivu, joka sisältää sivutusnäkymän kaikista tietokannan aihemerkinnöistä, jotka on jäsennelty näkymään luettelona korteilla, jotka sisältävät aiheen tunnuksen, aiheen nimen, aiheen luojan, aiheen vastausten määrän ja päivämäärän aiheen luomisesta. Topics-kotisivulla on myös aiheen luontipainike, joka näkyy vain niille käyttäjille, joilla on tili ja jotka ovat kirjautuneet sivustolle. Luo aihe -painike näyttää modaalin, joka kysyy aiheen nimeä. Kaikki muut aiheen luomiseen liittyvät tiedot, kuten esimerkiksi käyttäjän käyttäjätunnus, saadaan taustaprosesseilla, kuten todennuskontekstia käyttämällä.

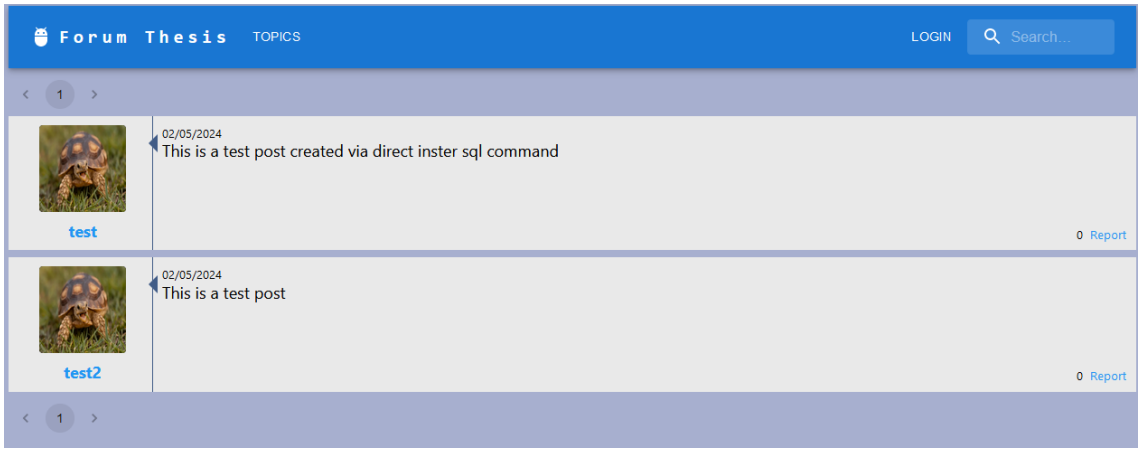
Kuva 7. Topics/Kotisivu

Kuva 8. Kotisivu kirjautuneena sisään.

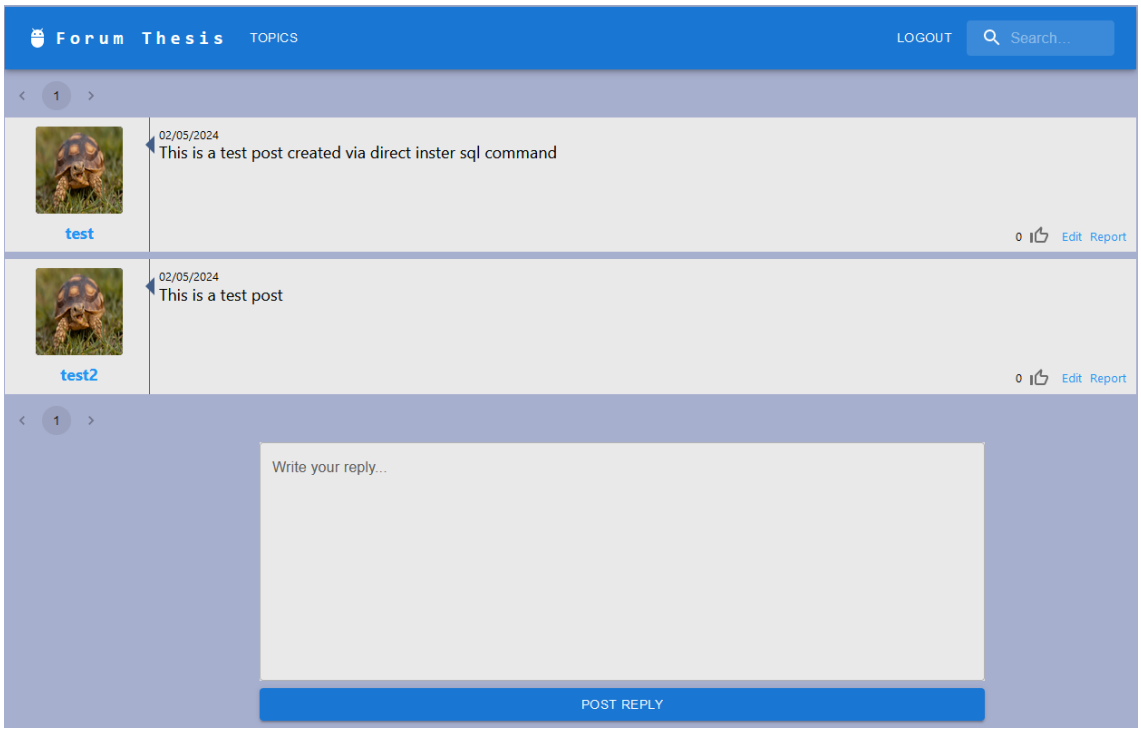
4.3.4 Keskustelusivu

Viesti tai keskustelusivu sisältää jokaisen yksittäisen viestin, jossa on valitun aiheen numeerinen aihetunnus, sivutettuna 25 merkintään per sivu. Sekä

kommenttilaatikko, joka itsessään on erillinen komponentti, josta kerrotaan tarkemmin myöhemmin.



Kuva 9. Kirjautumatta sisään.



Kuva 10. Kirjaudutta sisään.

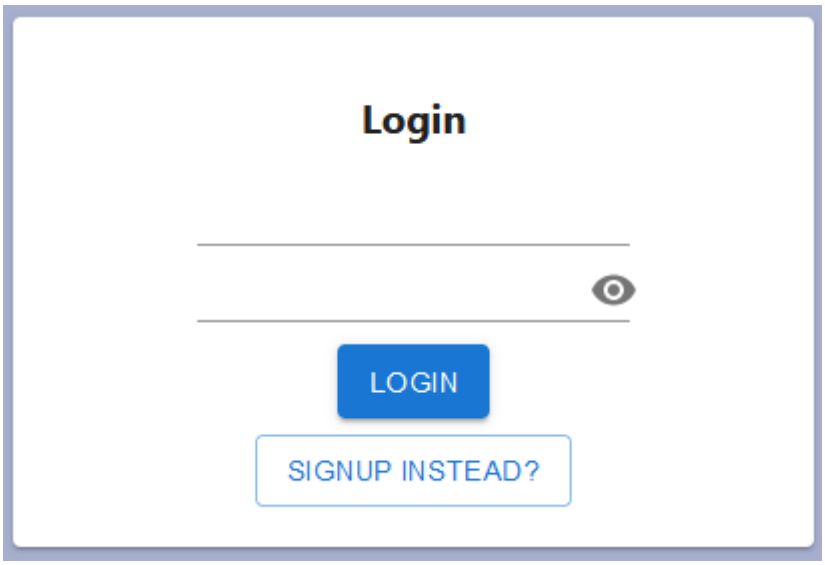
4.3.5 Kirjautuminen

Todennussivuilla on kortteja, jotka vaihtelevat valitun tilan mukaan, sisäänkirjautumistila edellyttää, että käyttäjä syöttää käyttäjätunnuksen, sähköpostiosoitteen ja salasanan tilin luomiseksi. Kun käyttäjä on napsauttanut

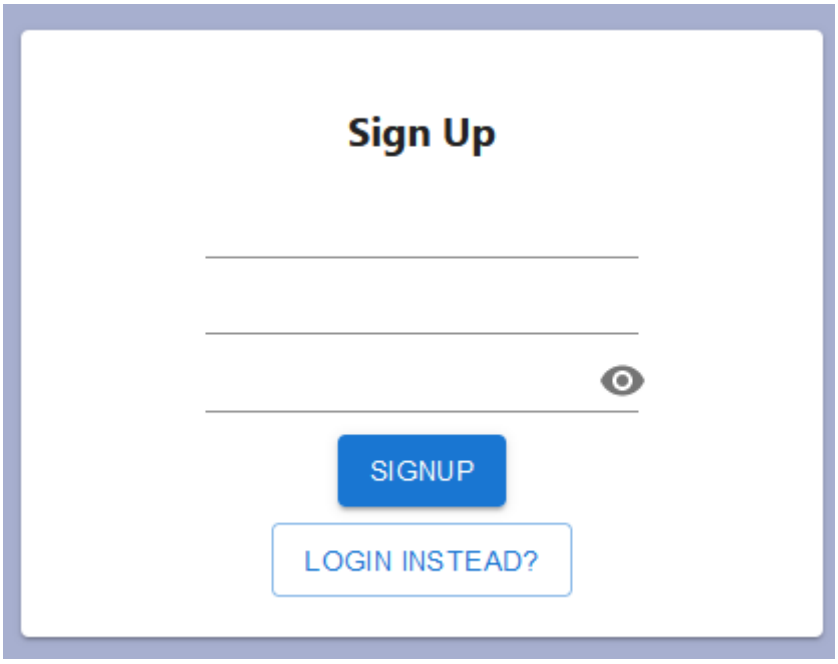
rekisteröitymispainiketta, se kirjautuu automaattisesti sisään. sivu vaatii käyttäjän sähköpostiosoitteen ja salasanan, joita käytetään sitten kirjautumiseen sisään. Molemmat tapaukset ovat vuorovaikutuksessa tietokannan kanssa ja tarkistavat annetun sähköpostiosoitteen.

Jos sähköpostiosoite vastaa jo järjestelmässä olevaa sähköpostiosoitetta rekisteröitymisen yhteydessä, sivusto näyttää virheilmoituksen, samoin jos vastaavaa sähköpostiosoitetta ei löydy kirjautumisruudusta, tulee virheilmoitus. Mikäli annetut tiedot vastaavat sähköpostiosoitetta ja tietokannassa olevaa salattua salasanaa, niin käyttäjä kirjautuu sisään. Sisäänkirjautumisen aikana luodaan tunti voimassa oleva token.

Tämän tunnuksen avulla tarkistetaan, onko käyttäjä kirjautunut sisään ja siten onko tiettyjä toimintoja käyttäjän käytettävissä. Oletuksena luoduilla käyttäjillä ei ole järjestelmänvalvojan oikeuksia, ja tämän opinnäytetyön kirjoittamishetkellä ainoa tapa luoda admin-tili on manuaalinen koodin suorittaminen manuaalisesti taustajärjestelmässä.



Kuva 11. Kirjautuminen.



Kuva 12. Tilin luonti.

4.3.6 Kommenttilaatikko komponentti

Kommenttilaatikkokomponentti käyttää Material UI -kirjastoa tekstikentän renderöintii, joka vastaanottaessaan käyttäjän syötteen tulostaa tekstikenttään käyttäjän kirjoitetun tekstin sujuvan käyttökokemuksen takaamiseksi ja päivittää text-nimisen tilan arvon setText set state -toiminnolla. Sitten kun käyttäjä lähettää tekstikenttään kirjoitetun tekstin, tiedot lähetetään replySubmitHandlerille, joka kutsuu createPostMutationin, joka kutsuu APIn createPost funktion käyttäen topic_id:n arvoa, käyttäjän syöttämää tekstiä, joka on asetettu arvoon textRef, valtuutetun käyttäjän tunnus, tykkäysmäärälle oletus arvo (0) sekä käyttäjän kirjautumisen aikana luotu tunnus. Tekstin arvo nollataan sitten tekstikentän tyhjentämiseksi.

Kuva 13. Kommenttilaatikko.

5 POHDINTA

Projekti oli samanaikaisesti sekä yksinkertaisempi että monimutkaisempi kuin odotin sen olevan. Backend oli melko yksinkertaista saada valmiiksi, ja vaikka jotkin sql-kyselyt vaikeuttivat päästä töihin, tässä ei ollut todellisia haasteita. Suurimmat haasteet olivat projektin frontend-puolella. Minun piti sopeutua uudempaan versioon Router-toiminnosta react-router-dom-kirjastosta sekä opetella Material UI lennossa, mikä oli välillä turhauttavaakin kokemus. Opin paljon tietojen kyselystä Reactissa ja opin välttämään piilossa olevat melko ilmeiset virheet. Opin myös osioiden haetut tiedot sivutusta varten MUI:n avulla, mikä oli mielenkiintoinen ja erittäin turhauttava kokemus.

Sain melkein kaikki halutut persu toiminnot tehtyä, ainuut mikä alkuperäisesti halutuista toiminnoista puuttuu ainoastaa forumin viestien ilmoittaminen. Itse funktionaalisuus on jo olemassa koodisaa, sekä kaikki admin sivut joissa ne ilmoitukset näkyy, mutta itse viestien ilmoittaminen ei toimi ihan oikein, ja taustalla on pieniä ongelmia sen kanssa. Muuten kaikki muut toiminnot on tehty. Käyttäjä voi lukea kaikki viestit valitun aiheen viestiketjusta ja voi hakea aiheita haku palkista. Samoin käyttäjä voi luoda tilin ja saatua tili luotua voi luoda omat aiheet, lähettää viestejä, tykätä viestit joista hän pitää ja muokata omat viestit.

Aion jatkaa tämän projektin kehittämistä opinnäytetyön valmistuttua, sillä mielestäni siinä on vielä paljon parantamisen varaa. Ajattelin lisätä profiilikuvia käyttäjätaulukkoon oletus kilpikonna png:n käyttämisen sijaan. Sekä varsinaisen profiilisivun lisääminen, joka sisältäisi joitain profiilinhallintatoimintoja, kuten salasanan vaihtaminen, profiilikuvan vaihtaminen ja tilin poistaminen.

Projektin tekninen toteutus löytyy github arkistosta (Tekninen toteutus)

LÄHTEET

OpenJS Foundation. n.d. NodeJS documentation. Verkkosivu. Viitattu 14.5.2024. https://nodejs.org/docs/latest/api/

OpenJS Foundation. n.d. Express homepage. Verkkosivu. Viitattu 14.5.2024. https://expressjs.com/

Sidorov, A. MySQL2 repository. Verkkosivu. Viitattu 14.5.2024. https://sidorares.github.io/node-mysql2/docs/documentation

Campbell, N. Bcrypt repository. Verkkosivu. Viitattu 14.5.2024. https://github.com/kelektiv/node.bcrypt.js/tree/master#readme

Provos, N., Honeyman, P. ja Mazières, D., 1999. A Future-Adaptable Password Scheme. Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference, Monterey, California, USA, June 6–11, 1999. Saatavilla: https://www.usenix.org/legacy/events/usenix99/provos/provos.pdf [Viitattu 20.5.2024].

Auth0, Inc. n.d. JSONwebToken repository. Verkkosivu. Viitattu 14.5.2024. https://github.com/auth0/node-jsonwebtoken#readme

Meta. n.d. React homepage. Verkkosivu. Viitattu 14.5.2024. https://react.dev/

MUI. n.d. Material UI documentation. Verkkosivu. Viitattu 14.5.2024. https://mui.com/material-ui/getting-started/

Gaijin Entertainment. n.d. WarThunder forum. Verkkosivu. Viitattu 20.5.2024. https://forum.warthunder.com/

Gaijin Entertainment. n.d. WarThunder-verkkopeli. Verkkosivu. Viitattu 20.5.2024. https://warthunder.com/fi

Stack Overflow. n.d. Stackoverflow forum. Verkkosivu. Viitattu 20.5.2024. https://stackoverflow.com/

The Observer. n.d. SpaceBattles forum. Verkkosivu. Viitattu 20.5.2024. https://forums.spacebattles.com/

X Corp. n.d. Formerly twitter. Verkkosivu. Viitattu 20.5.2024. https://x.com

Kovalev, D. Tekninen toteutus. Verkkosivu. Viitattu 20.5.2024. https://github.com/BoltVolta/thesis-web-forum