**Ryhmä Project**

**Projektipäällikkö:** Akonpelto Jooa

**Vastuut:**

Kuka haluaa tehdä mitä (kirjottakaa nimi jonnekin tehtävän kohtaan)

**Lainaukset** - (userTeam):

Näytä lista lainauksista. Jokaisella lainauksella tulee olla ainakin teksti ja kirjoittaja. - **Jooa**

**Lainauksen lisääminen** - (adminTeam):

Mahdollisuus lisätä uusia lainauksia sovellukseen admin-tunnuksella admin-paneelin lomakkeen kautta.

**Sakke Minkkinen**

**Lainauksen tietojen näyttäminen** - (userTeam): Käyttäjä voi tarkastella yksittäisen lainauksen tietoja. (esim. Näytä lainauksen teksti ja kirjoittaja) - **Jooa**

**Responsiivisuus** - (userTeam, adminTeam): Sovellus toimii hyvin eri näyttökokoluokissa, mukaan lukien mobiililaitteilla. **Verneri Pelho, Miska**

**Yksinkertainen käyttöliittymä** - (userTeam, adminTeam):

Käyttöliittymä on helppokäyttöinen ja intuitiivinen.

Vältä tarpeetonta monimutkaisuutta. Selkeä ja houkutteleva käyttöliittymä helpottaa lainausten selailua.

**Miska**

**Perustoiminnallisuudet** - (userTeam):

Käyttäjä voi selata lainauksia hakusanoilla tai suodattaa niitä esimerkiksi kirjoittajan tai aiheen perusteella. Kirjautuminen ja käyttäjätilit: Käyttäjät voivat luoda käyttäjätilin (GR-tunnus), joka mahdollistaa lainauksen. **Julia**

**Koodin ylläpito ja selkeys** - (userTeam, adminTeam): Koodin tulisi olla selkeää ja helposti ymmärrettävää. Pilko koodi pieniin komponentteihin ja hyödynnä Reactin ominaisuuksia. **Miska** (user-puoli)

**Perusvisuaalinen suunnittelu** - (userTeam,adminTeam):

Käytä yksinkertaista visuaalista suunnittelua, joka houkuttelee sovelluksen käyttäjiä. **Julia**

**Dokumentaatio** - (userTeam, adminTeam): Luo yksinkertainen dokumentaatio, joka selittää sovelluksen toiminnan ja kuinka sitä käytetään. **Verneri Pelho, Miska**

**Testaus** - (userTeam, adminTeam): Testaa sovelluksen perustoiminnallisuudet varmistaaksesi, että ne toimivat odotetusti. - **Jooa, Verneri Pelho**

**Tietoturva** - (userTeam, adminTeam): Varmista tietoturva käyttäjätietojen osalta. **Julia**

**Tallennus ja tietokanta** - (userTeam, adminTeam): Lainaukset ja muut tiedot on tallennettava johonkin tietokantaan tai tiedostoon, JSON-muotoon. Kannattaa tehdä kaksi erillistä JSON-tiedostoa. Ensimmäinen sisältää lainaukset ja niiden tiedot, kuten teksti ja kirjoittaja. Toinen sisältää käyttäjätilien tiedot, kuten käyttäjänimi ja salasana. - **Jooa**

**Vastuu Järjestys:**

kaikille oma vastuu alue ja sopikaa tehtävistä tiimin muiden jäsenten kanssa

Lainaukset:

Lainaukset, lainausten lisääminen ja lainausten näyttäminen

Kirjautuminen ja käyttäjätilit:

Kirjautuminen ja käyttäjä tilit ja se admin kirjautumis- juttu (?)

Tallennus ja tietokanta:

Talletus ja tietokanta ja tietoturva

QR-koodi:

**AIKATAULU:**

aluksi viikkotasolla (vko1, vko2, vko3, vko4), mieti sen jälkeen vähän tarkempi aikataulu päivätasolla

**Viikko 1:**

**Komponenttien luominen:**

* Luo komponentti lainauslistalle, joka näyttää kaikki lainaukset.
* Jokainen lainaus voi olla oma komponenttinsa, joka sisältää lainauksen tekstin ja kuvat ja tiedot mahdollisesta kirjoittajasta.

**JSON-tiedoston käsittely:**

* Luo JSON-tiedosto, joka sisältää lainaukset ja niiden tiedot.
* Käytä tilaa (state) komponenteissa hallitaksesi lainauksia.

**Viikko 2:**

**Lainauslistan näyttäminen:**

* Näytä JSON-tiedostosta ladatut lainaukset lainauslistalla.
* Voit käyttää .map()-funktiota luodaksesi jokaiselle lainaukselle oman komponentin.

**Admin-käyttäjän lisäämät lainaukset:**

* Luo lomake, jolla käyttäjä voi lisätä uusia ja poistaa käytöstä lainauksia.
* Kun käyttäjä lähettää lomakkeen, lisätään uusi lainaus JSON-tiedostoon tai poistetaan.

**Viikko 3:**

**Hakutoiminnot:**

* Lisää hakutoiminto, jolla käyttäjä voi etsiä lainauksia avainsanojen perusteella tai jotenkin muuten.
* Huomio saldo, jos kaikki on lainassa niin tuotetta ei voi lainata.

**Käyttäjäkokemus ja ulkoasu:**

* Suunnittele käyttäjäystävällinen käyttöliittymä, käytä CSS:ää ulkoasun parantamiseen.

**Viikko 4:**

**Hakutoiminnot:**

* Lisää hakutoiminto, jolla käyttäjä voi etsiä lainauksia avainsanojen perusteella tai jotenkin muuten.
* Huomio saldo, jos kaikki on lainassa niin tuotetta ei voi lainata.

**Julkaisu:**

* Kun sovellus on valmis, voit julkaista sen verkossa esimerkiksi GitHub Pagesilla.

**Riskit:**

* JSON-tietokanta
* Kirjautuminen ja käyttäjätilit
* Käyttäjätietojen salaus tietokannassa
* koko QR-koodi systeemi
* [Käyttäjien tietojen tallentaminen seuraavaa kirjautumista varten](https://medium.com/@sushinpv/how-to-store-data-securely-into-local-storage-using-react-secure-storage-83626919be13)
* Henkilöriskit (sairaudet, poissaolot)
* aikataulu

**Implementointi:**

**JSON-tietokanta:**

Tietokanta jaetaan 2 erilliseen tiedostoon (Käyttäjänimet ja salasanat) ja (Lainauksien tiedot)

**JSON**-esimerkki **rakenne**:

**Käyttäjänimet ja salasanat:**

{

"gr123456" : {

"salasana":"Salasana123!"

},

"gr789101" : {

"salasana":"Salasana456!"

}

}

**Lainauksien tiedot:**

{

"gr123456" : {

"lainaukset" :{

"tuotteet": [1,2,43] //tuotteen ID

}

}

}

**Kirjautuminen ja käyttäjätilit:**

GR-tunnuksella käyttäjänimi, eli GR-Alkuinen ja kuusi (?) numeroa perään. regex **^gr\d{6}$**  
Salasanassa pitää olla perusturvavaatimukset (Väh. 8-merkkia, yksi symboli tms. Jne.)

**Käyttäjätietojen salaus tietokannassa:**

Tiedot pitää salata tekniikalla, esim. **argon2** open suosima.

**QR-koodi systeemi:**

Skannaat puhelimella QR-koodin. QR-koodi sisältää *nettisivun-nimen.com/home*/*lainaus-ID-2* tai jotain samankaltaista, jonka avulla netti sivu pitää tunnistaa minkä tuotteen käyttäjä haluaa lainata ja tarkistaa onko sitä lainattavissa. Jos sitä ei ole lainattavissa, sivun pitää tarkistaa tuotteen vapautumisen pvm.

Tilanteessa, jos käyttäjä ei ole kirjautunut. Pitää käyttäjä välittää kirjautumissivulle, jonka jälkeen tuote asetetaan käyttäjän lainaus listaan

**Layout suunnitelma:**

Kaikki tekee layout suunnitelmaa ja antaa omia ideoita siihen.

Layout aloitetaan, kun on saatu koodin perusasiat tehtyä.