**Next.js Dashboard**

Sisällysluettelo

[1 Yleistä tietoa projektista 3](#_Toc197019110)

[2 Käytettyjä tekniikoita ja erikoisuuksia 3](#_Toc197019111)

[3 Sivuston / ohjelmiston design ja rakenne 3](#_Toc197019112)

[4 Oma arvio työstä ja oman osaamisen kehittymisestä 3](#_Toc197019113)

[5 Palaute opettajalle kurssista sekä itse opetuksesta tähän saakka 4](#_Toc197019114)

[6 Linkit 4](#_Toc197019115)

[6.1 Sovellukseni verkko-osoite 4](#_Toc197019116)

[6.2 GitHub-repositorion verkko-osoite 4](#_Toc197019117)

[6.3 Projektin videoesityksen verkko-osoite 4](#_Toc197019118)

[7 Lähteet 5](#_Toc197019119)

# Yleistä tietoa projektista

Rakensin opintojakson *Full Stack -sovelluskehitys TO00BS65-3007* loppuprojektina *dashboardin*. Dashboardin käyttäjä voi tarkastella, lisätä, muokata ja poistaa laskutustietoja. Dashboard koostuu julkisista etu- ja kirjautumissivuista, sekä käyttäjäkirjautumisen takaa saavutettavista koti-, laskut- ja asiakkaat -sivuista.

Dashboard on *full stack* -sovellus: käyttäjän antamien syötteiden perusteella käyttöliittymä pyytää tarvittavia tietoja tai lähettää haluttuja tietoja palvelimelta/palvelimelle, joka ottaa edelleen yhteyden tietokantaan ja tekee siellä tarvittavat toimenpiteet.

Dashboard on rakennettu seuraten 16-osaista *Next.js*:n *App Router* -tutoriaalia.

# Käytettyjä tekniikoita ja erikoisuuksia

Rakensin projektin käyttäen Next.js -teknologiaa. Next.js on verkkokehitysviitekehys (*web development framework*), joka on rakennettu *React*-verkkokehitysviitekehyksen päälle.

Koin dashboardin rakentamisen aikana mielenkiintoisimmiksi uudet React Server Components ja React Suspense -komponentit (*suspend*, lykätä tai viivästyttää). Erityisesti Suspense kiinnitti huomioni, sillä en ollut koskaan kuullut siitä. Lyhyesti, kyseisellä rajapinnalla voidaan määrittää fallback, skeleton, (varmistus) näytettäväksi sillä välin, kun tiettyä komponenttia ladataan. Näin käyttäjäkokemuksesta saadaan jatkuva, skeletonin nähdessään käyttäjä ymmärtää heti, että sivulle ollaan nyt lataamassa jotakin hänelle näytettäväksi.

# Sovelluksen design ja rakenne

Halusin, että sovellukseni olisi …

# Oma arvio työstä ja oman osaamisen kehittymisestä

Mielestäni onnistuin toteuttamaan kaikki projektissa vaaditut ominaisuudet sekä luomaan sovelluksen ympärille kaikki halutut lisäsisällöt: loin sovelluksesta kertovan videoesityksen sekä kirjoitin siitä kattavan projektiraportin.

Projektin koodin kommentoin kaikkein oleellisimmilta osin. Tämän tein lähinnä omaksi hyödykseni, sillä tahdon säilyttää tulevaisuuteenkin mahdollisuuden palata tarkastelemaan koodia, ja ymmärtää mitä siitä tapahtuu. Olen jo tässä vaiheessa pyrkinyt ymmärtämään mahdollisimman paljon koodin toiminnasta. Kokonaisuus on tosin niin laaja, että jokaista yksityiskohtaa en siitä vielä ymmärrä. Kaikki pääperiaatteet ovat kuitenkin hallinnassani.

Olen ehdottomasti oppinut projektin aikana uutta. React oli minulle melko tuttu jo entuudestaan, mutta Next.js ainoastaan otsikkotasolla. Opin projektin aikana uusia termejä ja käsitteitä sekä sain tarkempaa tietoa jo tuntemistani: *staattinen* ja *dynaaminen renderöinti* sekä *partial rendering*/*prerendering*, *network*/*request waterfall*, *object-related mapping*, *streaming*, *Route Groups*, *UI fallback* ja *UI skeleton*, *debouncing*, *Server Components* ja *Server Actions* sekä *Suspense* ja monet muut. *NextAuth.js* -käyttäjäkirjautumiskokonaisuuden luominen jäi erityisen mielenkiintoisena asiana mieleeni.

Kävin tutoriaalin läpi kahdesti. Ensimmäisellä kerralla rakensin applikaation tutoriaalin ohjeiden mukaisesti, samalla dokumentaatiota ajatuksella lukien. Toisella kerralla silmäilin tekstin uudestaan lävitse; tuolla kerralla kommentoin oman koodini tutoriaalista löytämieni avainkohtien perusteella. Vasta tällä toisella kerralla ymmärsin suurimman osan tutoriaalin sisältämistä asioista. Yhdeksi haastavimmista osioista ymmärtää koin osion 12, jossa sovellukseen lisättiin laskujen lisäys-, muokkaus- ja poistotoiminnot.

Projektin aikana oli kaiken kaikkiaan antoisaa tutustua Reactin uusimpiin käyttäjäkokemuksen huippuunsa hioviin rajapintoihin kuten Suspense, ja tutoriaalin kattavien kuvausten avulla huomata, kuinka paljon automatiikkaa, pellin alla tapahtuvaa magiikkaa, Next.js sisältää.

Kertomani ja kokemani sekä projektin osaamisvaatimusten perusteella antaisin itselleni projektin suorittamisesta täydet pisteet.

# Palaute opettajalle opintojaksosta sekä opetuksesta

Opintojakson opetussisällöt ovat olleet mielestäni oleelliset opintojakson nimeen (full stack) nähden. Olemme käyneet läpi

# Linkit

## Sovellukseni verkko-osoite

<https://nextjs-dashboard-azure-gamma-85.vercel.app>

## GitHub-repositorion verkko-osoite

<https://github.com/jonidaniel/nextjs-dashboard>

## Projektin videoesityksen verkko-osoite

asd

# Lähteet

Sähköiset

Projektin tehtävänanto (käyttäjäkirjautumisen takana):

<https://canvas.laurea.fi/courses/11658/assignments/209750>

App Router -tutoriaali:

<https://nextjs.org/learn/dashboard-app>