

Projekti I suunnitelma – Tiltless

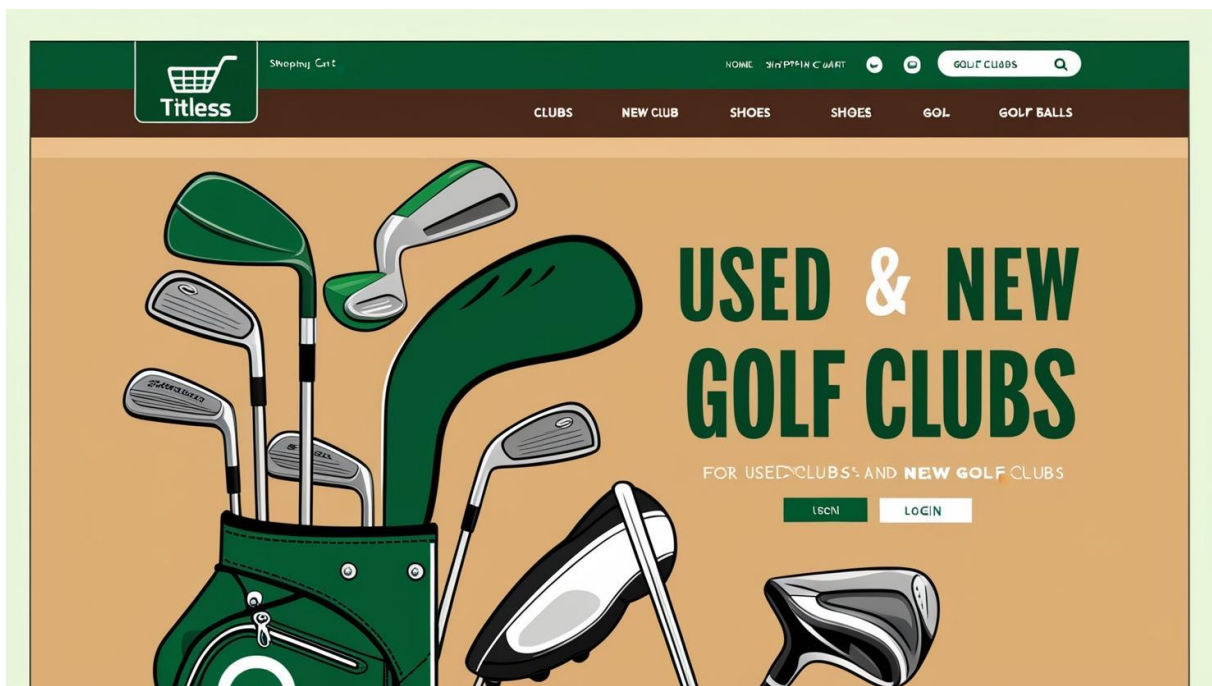
Kuvaus

Tarkoituksena rakentaa nettisivut verkkokaupalle. Kyseessä on Tiltless (omakeksimä nimi) niminen kauppa, joka myy golfvarusteita, niin käytettyjä mailoja ja varusteita, kuin myös uusia tietyiltä merkeiltä. Esimerkkinä golfmailat, pallot, kengät jne.

Tallentaakseen ja ostaakseen tuotteita käyttäjän on kirjaututtava sisään. Tuotteita voi selata ilman kirjautumista. Verkkosivulle ei luoda oikeita maksu- tai tilaustoimintoja, mutta teen silti ostokorin kirjautunutta käyttäjää kohden, jotta verkkokauppaa on helpompi mallintaa.

Havaintokuva

Seuraavana on havaintokuva verkkosivun etusivusta. Todellinen lopullinen versio tulee olemaan varmasti erilainen, mutta tyyli on sama. Tekstit ja kuvat varmasti tulevat olemaan erilaiset, sillä kaupan nimikin on kirjoitettu väärin tässä. Syynä havainnollistuskuvan virheille on, että se on tehty Canvan tekoälyllä, mutta tässä vaiheessa se kuvaa tarpeeksi hyvin verkkokaupan etusivua.



Käyttöliittymän toiminnallinen kuvaus

- Etusivu:
 - Tervehdysviesti/ -kuva
 - Linkki kirjautumissivulle
 - Linkki tuotteiden selaussivulle (esikatselu kategorioista ja voi jo valita suodatettavan kategorian etusivulla)
 - Yhteystiedot
- Kirjautumissivu

- . Lomake sähköpostille ja salasanelle
 - . Linkki rekisteröitymissivulle (lähes samanlainen kuin kirjautumissivu)
 - . Virheviestit esim. väärä salasana tai käyttäjätunnus
- Tuotteiden selaussivu
 - . Kategoriavalikko
 - . Järjestelyvalikko (esim. hinta alin-ylin, hinta ylin-alin)
 - . Hakukenttä nimille
 - . Itse tuotekortit, joissa kuva, nimi, hinta, lomake valittavalle määrälle ja ”Lisää ostokoriin”-nappi
- Ostokori-sivu (näkyv kirjautuneelle)
 - . Lista ostokorin tuotteita
 - . Nappi tilauksen tallentamisesta
 - . Nappi tuotteiden poistamiselle
- Tilaukset-sivu
 - . Lista eri tilauksista ja niihin kuuluvista tuotteista (ei oikea tilaus tai esim. maksutoiminnallisuuksia)

Tavoitteet

Tulen projektin aikana varmasti oppimaan FullStack-kehitystä, vahvistamaan taitojani käytettävillä teknologioilla sekä ymmärtämään paremmin ohjelmistokehityksen projektitoimintaa. Tämä tulee ohjelmoinnin harjoittelun jälkeen olemaan ensimmäinen itse toteutettu projekti, joten tulee varmasti opittua paljon.

Web-ohjelmoinnin kurssilla saanut hyvin pohjaa aiheelle, joten haluan vielä laajentaa osaamistani paljon projektin muodossa. On myös kiva päästä suunnittelemaan visuaalista puolta nettisivuista enemmän.

Tekninen puoli

Kielet

- JavaScript, CS, HTML

Kirjastot

- Express
- React
- Mongoose
- Axios
- bcrypt
- jsonwebtoken
- Jest
- Dotenv

Tietokanta

- MongoDB

Ympäristöt

- VisualStudioCode
- Postman
- npm
- GitHub

Teknisen toteutuksen havainnollistus

1. Projektin rakenne ja teknologiat

- **Frontend** (React + CSS)
 - Sivuston käyttöliittymä: tuotteiden selaus, kirjautuminen, ostoskori
 - Axiosilla HTTP-pyynnöt backendille
 - React Router sivunavigointiin
 - Tilanhallinta: React Context / Redux (jos tarpeen)
- **Backend** (Express + MongoDB)
 - REST API tuotteille ja käyttäjille
 - Käyttäjääutentikaatio: bcrypt + jsonwebtoken
 - MongoDB tietokanta (Mongoose skeemat)
 - Ostoskorin tallennus kirjautuneelle käyttäjälle
- **Tietokanta** (MongoDB + Mongoose)
 - Kokoelmat ja niiden väliset riippuvuudet:

```
Users
{
  _id: ObjectId,
  email: String (unique),
  passwordHash: String,
  name: String,
  address: String,
  cart: [{ productId: ObjectId, quantity: Number }], // Ostoskorin sisältö
  orders: [ObjectId] // Viittaukset käyttäjän tilauksiin
}

Products
{
  _id: ObjectId,
  name: String,
  category: String (esim. "mailat", "pallot"),
  price: Number,
  description: String,
  brand: String,
  size: String (vain kengille),
  stock: Number,
  imageUrl: String
}

Orders
{
  _id: ObjectId,
  userId: ObjectId, // Viittaus tilaavaan käyttäjään
  products: [{ productId: ObjectId, quantity: Number }], // Tilauksen sisältö
  totalPrice: Number,
  status: String (esim. "pending", "shipped", "delivered"),
  date: Date
}
```

Ostokori tallennetaan käyttäjän tietoihin Users.cart

Käyttäjillä on viittaus tilauksiinsa

Tilaukset eli Orders viittaa käyttäjään userId:llä ja sisältävät tuotteet

Toiminnallinen suunnitelma

- Aloitusivu: Navigointivalikko (esim. tuotekategoriat), Sivun tervehdys viesti, sisäänkirjautumispainike
- Tietokanta: Tietokannasta tiedon hakeminen ja tiedon muokkaaminen
- Sisäänkirjautuminen: Käyttäjä voi kirjautua omilla tunnuksillaan ja luoda tarvittaessa uudet tunnukset. Uloskirjautuminen
- Ostokori: Sisään kirjautunut voi lisätä ostokoriin tuotteita, pyydetään kirjautumaan sisään
- Tuotteet: tietoja voidaan tarkastella, kengät ainoa missä koon voi valita, mailoihin voi valita rajattua customointia, ostokoriin voi lisätä useita tuotteta.

REST API:n endpointit ja dataformaatit

Autentikaatio (api/auth)

- POST /api/auth/register – Luo uusi käyttäjä
- { "email": "user@example.com", "password": "salasana123", "name": "Matti Meikäläinen" }
- POST /api/auth/login – Sisäänkirjautuminen (palauttaa JWT-tokenin)

Tuotteet

- GET /api/products – Listaa kaikki tuotteet (mahdolliset suodattimet lisäksi)
- GET /api/products/:id – Hae yksittäinen tuote

Ostokori

- GET /api/cart – Listaa kirjautuneen käyttäjän ostokorin (vaatii JWT-tokenin)
- POST /api/cart – Lisätään tuote käyttäjän ostokoriin
- { "productId": "123abc", "quantity": 2 }

Tilaukset

- POST /api/orders – Luo uusi tilaus ostokorista, vaatii kirjautumisen
- GET /api/orders - Näyttää käyttäjän tekemät tilaukset

Projektissa käytettävä versionhallinta

Tulen käyttämään projektissa GitHubia version hallinnassa. Se on tullut yliopistokurssien aikana tutuksi ja todettu hyväksi.

Aikataulu

Tavoitteen projektin valmistumiselle olisi viimeistään huhtikuun loppuun mennessä. Kirjoitan samaan aikaan kandidutkielmaa + muut kurssit samaan aikaan, joten teen projektia aina kun muilta kouluhommilta ehdin. Tavoittelen 15-20 h viikkoaikataulua ja arvioin koko projektin vievän noin. 150 h(tavoitteena 5op).

Viikkotason aikataulu:

Viikko	Tehtävä
1	Tekninen suunnittelu, projektirakenne, Git-repo
2	Tietokantarakenne ja käyttäjäautentikaatio
3	Backendin REST API (tuotteet, käyttäjät)
4	Frontend: React-pohjan luonti, navigaatio, tuotelistaus
5	Tuotesivut, ostokori ja tilausten toteutus
6	Käyttäjätilit ja ostokorin tallennus tietokantaan
7	Testaus (Jest, Postman), virheenkäsittely
8	Viimeistely, dokumentointi, käyttöliittymän parantelu

Aikataulukin voi mahdollisesti muuttua matkan varrella sen suhteen, missä järjestyksellä lähdän liikkeelle, mutta tässä alustava suunnitelma. Tämän suunnitelman pohjalta alan projektia tekemään ja dokumentoin työ joka kerralla, kun sitä teen, niin lopuksi näkee miten työ poikkesi suunnitelmasta.