

Jeg prioriterte først den grunnleggende funksjonaliteten – altså å få på plass et solid brukersystem og bankkonto. Jeg startet med å lage tre hovedtabeller i databasen: en for brukere, en for bankkontoer og en for transaksjoner. Dette ga meg en god struktur som resten av systemet kunne bygges på.

Etter å ha fått tabellene på plass, satte jeg opp API-kommunikasjonen mellom Next.js og Node.js. På den måten kunne frontend prate med backend, og jeg kunne lage funksjoner for å sette inn og ta ut penger fra kontoene. Jeg implementerte også autentisering ved hjelp av session IDs, slik at systemet kunne sjekke om brukeren var logget inn.

Når de mest essensielle funksjonene var klare, gikk jeg løs på bonusoppgavene. Jeg lagde grafer for å vise kontoaktivitet og brukte transaksjonstabellene for å gi en visuell oversikt over bevegelsene på kontoen. Selv om jeg dessverre ikke rakk å få på plass renteutregning, mener jeg at de visuelle elementene gir en bedre brukeropplevelse.

For lagring av data valgte jeg PostgreSQL. Jeg syntes dette var et supert valg fordi PostgreSQL er kjapt på skriving, har et bra verktøy som pgAdmin for visualisering, og er helt gratis. Det gir en strukturert og skalerbar løsning som jeg kan bygge videre på senere.

Når det kommer til brukeropplevelse og navigasjon, lagde jeg en topbar med en logo som alltid tar deg tilbake til hovedmenyen. I tillegg har jeg en hamburger-meny for enkel tilgang til alle sidene, og på hver side er det en pil som sender deg tilbake til hovedmenyen. Jeg tenkte at færre tastetrykk gjør opplevelsen bedre, så jeg prøvde virkelig å holde navigasjonen så enkel som mulig.

I starten vurderte jeg å bruke Python, men følte raskt at det ikke passet til en nettside. Jeg er mye mer komfortabel med Next.js og Node.js i webutvikling, og disse teknologiene gir mange enkle muligheter for APIer. Jeg fikk også inspirasjon fra teknologien brukt av Spotify, noe som motiverte meg til å satse på denne stacken. Selv om jeg drømmer om å lage en desktop-app med C++ en dag, var det naturlig å gå for det jeg kan best i dette prosjektet.

En utfordring jeg støtte på, var problemer med å oppdatere state på hjemmesiden. useState oppførte seg ikke helt som forventet, så jeg måtte sjekke om det allerede var data der før jeg la til nye verdier. Hvis det fantes en eksisterende verdi, la jeg til den nye informasjonen, hvis ikke, satte jeg state til null. Om jeg skulle gjort det om igjen, ville jeg nok fokusert enda mer på å lage en sømløs registrerings- og innloggingsprosess slik at man ikke trenger å gjøre begge deler separat.

Alt i alt lærte jeg masse av dette prosjektet. Jeg fikk god erfaring med både backend og frontend, og jeg ser hvor viktig det er å ha et ryddig, intuitivt system for en bra brukeropplevelse. Jeg syntes også det var morsomt å prøve det å lage strukturert og bra kode.