

Prosjekt: Utvikling av nettsted for Hotell Alexandria

Velkommen til prosjektet hvor dere skal utvikle et komplett nettsted for det fiktive Hotell Alexandria. Målet er å designe og implementere et brukervennlig og estetisk tiltalende nettsted som gir besøkende all nødvendig informasjon om hotellet, lar dem booke rom, og hente bookinginformasjon ved behov.

Kravspesifikasjon:

- **Fire nettsider:**
 - **Hovedsiden:** Viser kort informasjon om hotellet med et bilde.
 - **Våre rom:** Informasjon om ulike romtyper, priser og fasiliteter.
 - **Book rom:** Skjema for booking av rom (romtype, antall gjester, kontaktinformasjon).
 - **Hent bookinginfo:** Mulighet for gjester å hente bookingdetaljer ved hjelp av et referansenummer.
- **Navigasjon:**
 - Implementer navigasjonsknapper som gjør det enkelt å bevege seg mellom sidene.
- **Datahåndtering:**
 - Ved booking genereres et unikt referansenummer.
 - Bookinginformasjon lagres i en JSON-fil på servernivå.
 - Gjester kan hente sin bookinginformasjon ved å oppgi referansenummeret.
- **Design og bilder:**
 - Nettstedet skal ha et helhetlig og tiltalende design.
 - Bruk bilder for å illustrere hotellet og rommene (du kan bruke KI-verktøy for å lage bilder).
- **Teknisk oppsett:**
 - **Mappestruktur:**
 - Opprett en **public**-mappe som inneholder HTML-filer, klient-side JavaScript og CSS.
 - Inni **public**-mappen skal det være en **images**-mappe der alle bilder lagres.
 - Server-side kode skal håndtere lagring og henting av bookinginformasjon.
- **Arbeidsprosess:**
 - Del opp arbeidet i 10 steg for å strukturere utviklingsprosessen (følg veiledningen gitt tidligere).
 - Målet er at dere skal lære å strukturere et webprosjekt uten å få direkte kodehjelp.

Dette prosjektet gir dere praktisk erfaring med webutvikling ved hjelp av Node.js, og lar dere anvende kunnskap om både frontend og backend i en reell situasjon. Lykke til med utviklingen av Hotell Alexandria sitt nettsted!

Mappestruktur

Start med å opprette følgende mappestruktur:

```
prosjektmappe/  
├─ server.js  
├─ bookings.json  
├─ public/  
│   ├─ index.html  
│   ├─ rom.html  
│   ├─ book.html  
│   ├─ hent.html  
│   ├─ styles.css  
│   ├─ script.js  
│   └─ images/  
│       └─ (her legger du inn bildene)
```

Steg 1: Oppsett av prosjekt og installasjon av nødvendige pakker

- Opprett en ny prosjektmappe og naviger til den i terminalen.
- Initialiser prosjektet med `npm init` for å lage en `package.json`-fil.
- Installer Express (eller et annet rammeverk du ønsker) med `npm install express`.

Steg 2: Opprett serverfilen (`server.js`)

- Sett opp en enkel Node.js-server ved hjelp av Express.
- Konfigurer serveren til å servere statiske filer fra `public`-mappen.
- Start serveren og test at den kjører på ønsket port.

Steg 3: Lag hovedsiden (`index.html`)

- I `public`-mappen, opprett `index.html`.
- Legg til grunnleggende HTML-struktur.
- Skriv kort informasjon om Hotell Alexandria.
- Legg til et bilde av hotellet fra `images`-mappen.

Steg 4: Lag navigasjonsmeny

- Opprett en navigasjonsmeny på `index.html` med lenker til:
 - Hovedsiden (`index.html`)
 - Våre rom (`rom.html`)
 - Book rom (`book.html`)
 - Hent bookinginfo (`hent.html`)
- Kopier navigasjonsmenyen til de andre HTML-filene for konsistens.

Steg 5: Lag siden for "Våre rom" (`rom.html`)

- Opprett `rom.html` i `public`-mappen.
- Beskriv ulike romtyper, fasiliteter og priser.
- Inkluder bilder av de forskjellige romtypene fra `images`-mappen.

Steg 6: Lag bookingsiden (`book.html`)

- Opprett `book.html` i `public`-mappen.
- Lag et skjema der gjester kan:
 - Velge type rom.
 - Angi antall gjester.
 - Oppgi kontaktinformasjon.
- Sett `action="/book"` og `method="post"` på `<form>`-elementet.
- Legg til en knapp for å sende inn bookingforespørselen.

Steg 7: Implementer bookingfunksjonalitet på serversiden

- I `server.js`, opprett en POST-route på `/book` som mottar dataene fra skjemaet.
- Generer et unikt referansenummer for hver booking (f.eks. ved hjelp av en teller eller en tilfeldig generert streng).
- Lagre bookinginformasjonen sammen med referansenummeret i `bookings.json`.
- Etter at bookingen er lagret, send en bekreftelse tilbake til brukeren som viser referansenummeret.

Steg 8: Lag siden for å hente bookinginfo (`hent.html`)

- Opprett `hent.html` i `public`-mappen.
- Lag et skjema der gjester kan skrive inn sitt referansenummer.
- Bruk `script.js` til å håndtere skjemaets innsending ved å lytte til `submit`-hendelsen.
- Når skjemaet sendes inn, bruk `fetch` API til å sende en GET-forespørsel til serveren med referansenummeret (f.eks. `/bookinginfo?ref=REFERANSENUMMER`).
- På serversiden (`server.js`), opprett en GET-route på `/bookinginfo` som henter bookinginformasjonen basert på referansenummeret fra `bookings.json`.
- Returner bookinginformasjonen som JSON.
- I `script.js`, motta responsen og vis bookinginformasjonen på siden.

Steg 9: Design og stilsetting

- Opprett en `styles.css`-fil i `public`-mappen.
- Legg til CSS for å gi nettstedet et helhetlig og attraktivt design.
- Sørg for at designet er responsivt og fungerer godt på ulike enheter.

Steg 10: Test og feilsøk

- Test alle funksjonaliteter på nettstedet:
 - Navigasjon mellom sider.
 - Innsending av booking og mottak av referansenummer.
 - Henting av bookinginformasjon med referansenummer.
- Kontroller at bookingdata lagres og hentes korrekt fra `bookings.json`.
- Gjør nødvendige justeringer basert på testing.

Ekstra tips:

- **Bilder:** Bruk AI-verktøy for å generere bilder av hotellet og rommene. Lagre disse i `public/images`-mappen.
- **Validering:** Legg til validering på skjemaene for å sikre at brukeren fyller inn nødvendige felt.

- **Sikkerhet:** Pass på å håndtere data på en sikker måte, spesielt når du leser fra og skriver til `bookings.json`.
- **Utvidelser:** Vurder å legge til ekstra funksjoner som avbestilling av booking eller en administrasjonsside for hotellet.

Dette prosjektet gir deg muligheten til å arbeide med både frontend og backend ved hjelp av Node.js. Ved å følge disse stegene kan du strukturere arbeidet ditt effektivt og bygge et funksjonelt nettsted for Hotell Alexandria.