# Eksamensoppgave utvikling vår 2024

[Søk i eksamensoppgaver](https://sokeresultat.udir.no/eksamensoppgaver.html?query=ITK2003)

**Situasjonsbeskrivelse**

Som et ledd i integrerings- og inkluderingstiltak ønsker Bergefjord kommune å tilby en chattetjeneste der innbyggerne kan registrere seg og delta i chattegrupper. Tanken er å ha en digital språkkafé der man kan lære seg et språk ved å chatte med andre.

Tjenesten er gratis og skal være tilgjengelig for alle over 16 år.

**Du lager dine egne begrunnede forutsetninger og avgrensninger slik du selv ser det naturlig, for å besvare oppgavene. Det som ikke står i oppgavene, velger du selv.**

**Oppgave 1: Databasemodell**

Lag en databasemodell som dekker informasjonsbehovet til chatteappen i tråd med situasjonsbeskrivelsen. Brukere skal ha en profil der man kan registrere hvilke språk man er interessert i, enten som mentor eller elev.

Når man logger seg på, kan man som bruker se aktive chatterom og eventuelt opprette nye chatterom hvis man ikke finner et som passer.

Beskriv modellen, med aktuelle attributter, datatyper og relasjoner. Gjør rede for de forutsetningene du finner nødvendig.

**Løsningsforslag oppgave 1:**

Forutsetningene jeg tok når jeg laget løsningen er:

* jeg holdt løsningen enklest mulig
* Innbyggere har jeg ikke noe mer profildata utenom navn og ID
* I chatterom har jeg bare et navn og et språk, ikke administratorer og etc.
* Chattemelding har ikke et språk, den hører til et rom som har et språk.
* En innbygger kan ønske å være mentor i flere språk og også være elev i flere språk.
* Innllogging/autentisering blir håndtert av en annen tjeneste.

Jeg ser for meg totalt 6 tabeller som en løsning.

**Innbygger:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Navn | Varchar (10) |  | N |  |

**Chatterom:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Navn | Varchar (10) |  | N |  |
| Språk | Integer | Ja | N | Språk:Id |

**Språk:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Navn | Varchar (10) |  | N |  |

**Chattemelding:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Dato | Datetime |  | N |  |
| Avsender | Integer | N | N | Innbygger:Id |
| Chatterom | Integer | N | N | Chatterom:Id |
| Melding | Varchar(2024) | N | N |  |

**Mentor:**

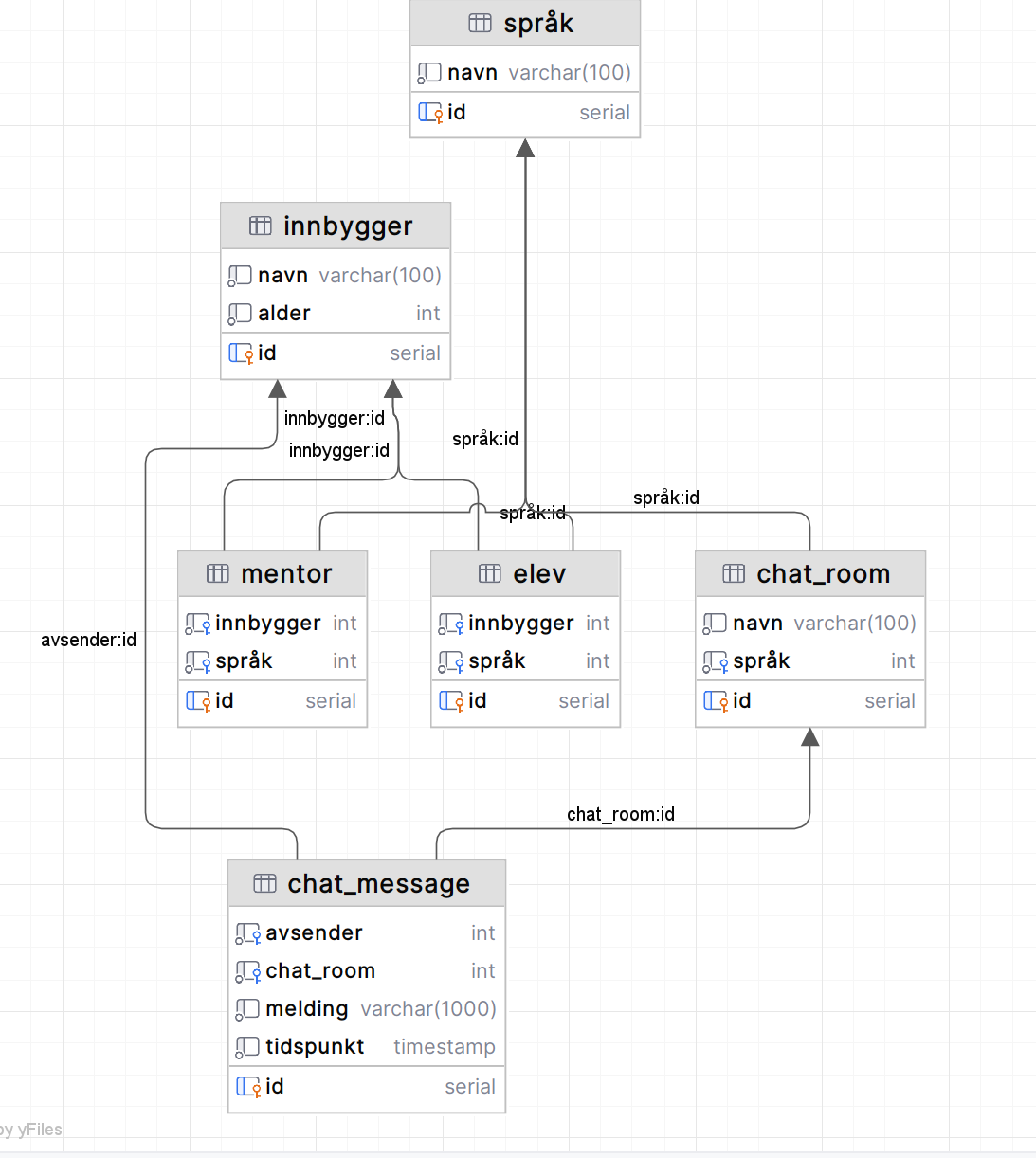
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Innbygger | Integer | N | N | Innbygger:Id |
| Språk | Integer | N | N | Språk:Id |

**Elev:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kolonnenavn | Datatypen | Unik? | Null? | FK? |
| ID | Integer | Ja | N |  |
| Innbygger | Integer | N | N | Innbygger:Id |
| Språk | Integer | N | N | Språk:Id |

Her er SQL kloden for å lage databasen:

create table innbygger (  
 id integer primary key,  
 navn varchar(*100*) not null  
);  
  
create table språk (  
 id integer primary key,  
 navn varchar(*100*) not null  
);  
  
create table chat\_room (  
 id integer primary key,  
 navn varchar(*100*) not null,  
 språk int not null references språk(id)  
);  
  
create table chat\_message (  
 id integer primary key,  
 avsender int not null references innbygger(id),  
 chat\_room int not null references chat\_room(id),  
 melding varchar(*1000*) not null,  
 tidspunkt timestamp not null  
);  
  
create table mentor (  
 id integer primary key,  
 innbygger int not null references innbygger(id),  
 språk int not null references språk(id)  
);  
create table elev (  
 id integer primary key,  
 innbygger int not null references innbygger(id),  
 språk int not null references språk(id)  
);



**Oppgave 2: Skjermbilde**

**Lag ett av følgende to skjermbilder til chatteappen som beskrevet i situasjonsbeskrivelsen ved hjelp av et markeringsspråk (f.eks. HTML eller CSS).**

**1  Lag skjermbildet som vises etter at man har logget inn, hvor man ser tilgjengelige chatterom og eventuelt kan opprette et nytt.**

**eller**

**2  Lag skjermbildet der brukeren registrerer profilen sin.**

**Løsning 1:**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>ChatteApp</title>  
  
 <style>  
 body {  
 font-family: Arial, sans-serif;  
 }  
  
 .user-name {  
 font-weight: bold;  
 color: darkmagenta;  
 }  
  
 h1 {  
 text-align: center;  
 color: darkmagenta;  
 }  
  
 fieldset legend {  
 color: darkmagenta;  
 }  
  
 .profile {  
 text-align: center;  
 }  
  
 .chat-rooms {  
 margin: 0 auto;  
 width: 50%;  
 }  
  
 .chat-rooms ul {  
 list-style-type: none;  
 padding: 0;  
 }  
  
 .chat-rooms li {  
 background-color: #f0f0f0;  
 border: 1px solid #ccc;  
 margin-bottom: 5px;  
 padding: 5px;  
 }  
  
 .new-chat-room {  
 margin: 0 auto;  
 width: 50%;  
 }  
  
 .new-chat-room fieldset {  
 border: 1px solid #ccc;  
 padding: 10px;  
 }  
  
 .new-chat-room legend {  
 font-weight: bold;  
 }  
  
 .new-chat-room label {  
 display: block;  
 margin-bottom: 5px;  
 }  
  
 .new-chat-room input[type="text"],  
 .new-chat-room select {  
 width: 100%;  
 }  
  
 .new-chat-room button {  
 margin-top: 5px;  
 }  
  
 </style>  
</head>  
<body>  
  
<h1>ChatteApp</h1>  
  
<section class="profile">  
<p>  
 Du er logget inn som <span class="user-name">Innbygger Innbyggersen</span>.  
</p>  
<button>Logg ut</button>  
</section>  
  
<section class="chat-rooms">  
<ul>  
 <li>Spansk - matlaging</li>  
 <li>Italiensk - vinkurs</li>  
 <li>Spansk for nybegynnere</li>  
</ul>  
</section>  
  
<section class="new-chat-room">  
<form>  
 <fieldset>  
 <legend>Opprett nytt chatterom</legend>  
 <label>  
 Navn på chatterom:  
 <input type="text" name="room-name">  
 </label>  
 <label>  
 Språk:  
 <select name="language">  
 <option value="spansk">Spansk</option>  
 <option value="italiensk">Italiensk</option>  
 <option value="engelsk">Engelsk</option>  
 </select>  
 </label>  
 <button type="submit">Opprett</button>  
 </fieldset>  
</form>  
</section>  
  
</body>  
</html>

**Oppgave 3: Utvikling**

**Beskriv en mulig løsning for hvordan du henter og viser data fra databasen i chatteappen.**

**Lag enten**

**1) et konkret kodeeksempel til skjermbildet som viser tilgjengelige chatterom**

**eller**

**2) pseudokode til skjermbildet som som viser tilgjengelige chatterom**

**Svar 1:**

Forutsetning:

Vi antar at vi har satt opp en database server, og en server som eksponerer databasen ved hjelp av REST (JSON over http).

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>ChatteApp</title>  
  
 <style>  
 body {  
 font-family: Arial, sans-serif;  
 }  
  
 .user-name {  
 font-weight: bold;  
 color: darkmagenta;  
 }  
  
 h1 {  
 text-align: center;  
 color: darkmagenta;  
 }  
  
 fieldset legend {  
 color: darkmagenta;  
 }  
  
 .profile {  
 text-align: center;  
 }  
  
 .chat-rooms {  
 margin: 0 auto;  
 width: 50%;  
 }  
  
 .chat-rooms ul {  
 list-style-type: none;  
 padding: 0;  
 }  
  
 .chat-rooms li {  
 background-color: #f0f0f0;  
 border: 1px solid #ccc;  
 margin-bottom: 5px;  
 padding: 5px;  
 }  
  
 .new-chat-room {  
 margin: 0 auto;  
 width: 50%;  
 }  
  
 .new-chat-room fieldset {  
 border: 1px solid #ccc;  
 padding: 10px;  
 }  
  
 .new-chat-room legend {  
 font-weight: bold;  
 }  
  
 .new-chat-room label {  
 display: block;  
 margin-bottom: 5px;  
 }  
  
 .new-chat-room input[type="text"],  
 .new-chat-room select {  
 width: 100%;  
 }  
  
 .new-chat-room button {  
 margin-top: 5px;  
 }  
  
 </style>  
  
 <script>  
  
 function *fetchChatRooms*() {  
 const url = 'chat-rooms.json';  
 *fetch*(url)  
 .then(response => response.json())  
 .then(chatRooms => {  
 const chatRoomList = document.getElementById('chat-room-list');

chatRoomList.innerHTML = '';  
 for (let room of chatRooms) {  
 const li = document.createElement('li');  
 li.textContent = room.navn;  
 chatRoomList.appendChild(li);  
 }  
 });  
 }  
  
 document.addEventListener('DOMContentLoaded', *fetchChatRooms*);  
  
  
  
 </script>  
</head>  
<body>  
  
<h1>ChatteApp</h1>  
  
<section class="profile">  
<p>  
 Du er logget inn som <span class="user-name">Innbygger Innbyggersen</span>.  
</p>  
<button>Logg ut</button>  
</section>  
  
<section class="chat-rooms">  
<ul id="chat-room-list">  
</ul>  
</section>  
  
<section class="new-chat-room">  
<form>  
 <fieldset>  
 <legend>Opprett nytt chatterom</legend>  
 <label>  
 Navn på chatterom:  
 <input name="room-name" type="text">  
 </label>  
 <label>  
 Språk:  
 <select name="language">  
 <option value="spansk">Spansk</option>  
 <option value="italiensk">Italiensk</option>  
 <option value="engelsk">Engelsk</option>  
 </select>  
 </label>  
 <button type="submit">Opprett</button>  
 </fieldset>  
</form>  
</section>  
  
</body>  
</html>

**chat-rooms.json fila ser slik ut:**

[  
 {  
 "id": 1,  
 "navn": "Spansk - matlaging"  
 },  
 {  
 "id": 2,  
 "navn": "Norsk - grammatikk"  
 },  
 {  
 "id": 3,  
 "navn": "Engelsk - skriving"  
 }  
]

**Selve websiden:**

Et bilde som inneholder tekst, skjermbilde, programvare, nummer

Automatisk generert beskrivelse

**Oppgave 4: Digitale trusler og sårbarheter**

**Du blir bedt om å gjøre en kort analyse av sårbarheter og digitale trusler til denne chatteappen. Kan du gi noen eksempler på hvordan du kan sikre at chatteappen har innebygd sikkerhet?**

* Som nevnt forutsetter jeg at autentisering og eventuelt tilgangskontroll blir håndtert av andre tjenester.
* Beskyttelse mot SQL injeksjon må håndteres på server siden for eks. ved å bruke bindingsvariabler.
* Beskyttelse mot XSS og andre klient angrep bør håndteres ved å bruke et rammeverk for eks. Angular eller React. I mitt kodeeksempel burde jeg har brukt html-escaping av bruker generert tekst, for eks. navn på chatt rom.
* Jeg forutsetter at serverne er beskyttet mot nettverksbaserte angrep ved hjelp av for eks. brannmurer, rate-limiting etc.