

ORRDTLILRRD

Overcoming reading challenges





Hvad er
"Ord-Til-Ord"

- Applikationen "Ord Til Ord" er en applikation der er designet til at støtte børn med læsevanskeligheder gennem skræddersyede historier, der fremmer daglig læsetræning.
- Ved at omdanne børns daglige oplevelser til engagerende tekster, tilpasser den læringsindholdet personligt og gør læsningen relevant og motiverende.
- Samtidig fungerer den som et bindeled mellem barnet og voksne, hvilket gør det muligt for dem at bidrage til barnets læseudvikling på en interaktiv måde.



Karakteristika og symptomer

- **Aflæsning af ord:** Børn med læsevanskeligheder kan kæmpe med at genkende ord hurtigt og præcist. De kan læse langsomt og tøvende, og ofte misforstå eller springe ord over.
- **Forståelse:** De kan have svært ved at forstå det læste, hvilket skyldes udfordringer med ordforråd, sætningsstruktur og at skabe sammenhæng mellem tekstdele.
- **Stavning:** Stavning kan være inkonsekvent, og børnene kan have vanskeligheder med at huske og anvende stavereregler.
- **Læse flyd:** Læsning kan mangle flyd og kan virke hakket, hvilket gør det svært at forstå og huske det læste.



Påvirkning

- **Selvværd:** Børn kan udvikle et lavt selvværd på grund af frustration og følelsen af at være bagud i forhold til jævnaldrende.
- **Motivation:** Vedvarende udfordringer kan føre til manglende motivation for skolearbejde og læring generelt.
- **Sociale færdigheder:** Læsevanskeligheder kan begrænse børns evne til at deltage i aktiviteter, der kræver læsning, hvilket kan påvirke sociale interaktioner og venskaber.
- **Hverdagsopgaver:** Almindelige opgaver som at læse skilte, menuer eller instruktioner kan blive udfordrende og frustrerende.



Forståelsen af disse aspekter er central for udviklingen af støtteværktøjer som "Ord Til Ord"-applikationen, der har til formål at imødekomme og afhjælpe nogle af de udfordringer, som børn med læsevanskeligheder møder.



Målgruppe

- **Indskoling:** aldersgruppen 6-9 år (DK: 250.000 børn)
- **Diagnosticering:** Dysleksi (3-7%)



Håndtering af personlige oplysninger

- **Sensitivitet vedrørende børns data:** Børn er en særligt sårbar gruppe, hvilket gør deres data yderst følsomme. Det er kritisk at overveje etikken i at indsamle og behandle børns personlige oplysninger.
- **Overholdelse af lovgivningen:** Applikationen skal overholde den gældende databeskyttelseslovgivning, herunder GDPR, som sætter strenge rammer for, hvordan personoplysninger skal håndteres.
- **Forældres eller værgers samtykke:** Det er ikke kun et lovkrav, men også et etisk ansvar, at indhente klart samtykke fra forældre eller værger, før børns persondata indsamles eller behandles.

ORRDTLILRO

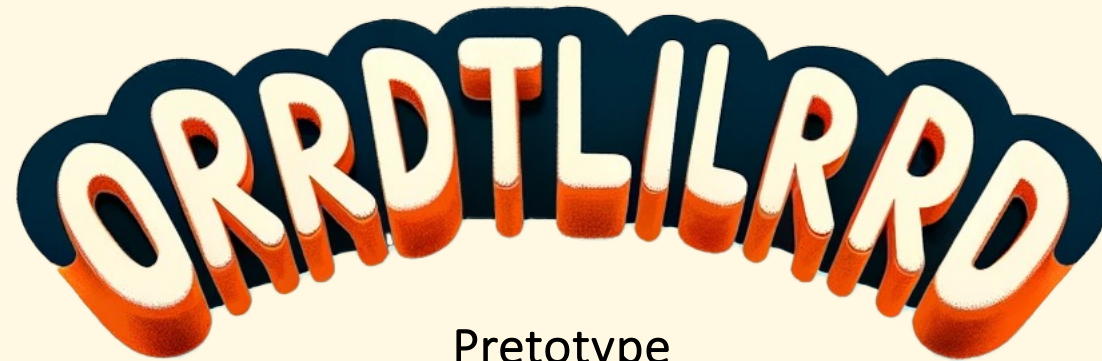
Risikominimering

- **Sensitivitet vedrørende børns data:** Børn er en særligt sårbar gruppe, hvilket gør deres data yderst følsomme. Det er kritisk at overveje etikken i at indsamle og behandle børns personlige oplysninger.
- **Overholdelse af lovgivningen:** Applikationen skal overholde den gældende databeskyttelseslovgivning, herunder GDPR, som sætter strenge rammer for, hvordan personoplysninger skal håndteres.
- **Forældres eller værgers samtykke:** Det er ikke kun et lovkrav, men også et etisk ansvar, at indhente klart samtykke fra forældre eller værger, før børns persondata indsamles eller behandles.



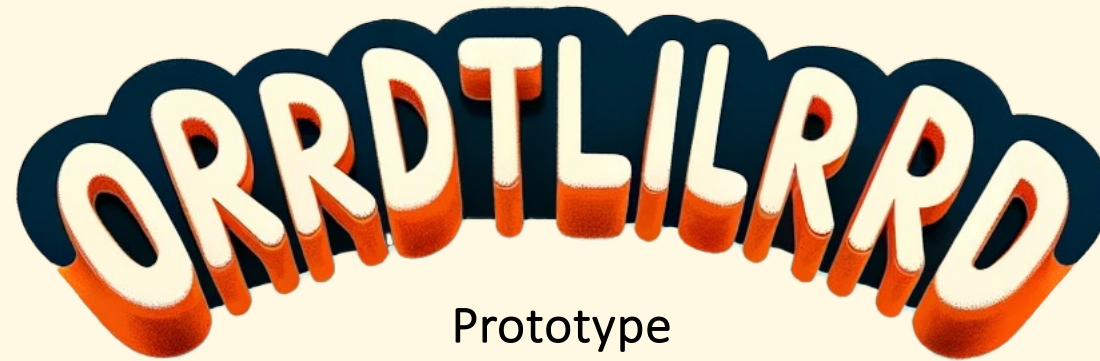
Risiko

Ved omhyggeligt at adressere og mindske risici forbundet med håndtering af personlige oplysninger, kan vi sikre at applikationen beskytter brugernes privatliv, opbygger tillid blandt brugerne og overholder lovgivningen, hvilket er afgørende for succesfuld implementering og accept af teknologien.



Pretotype

1. **Dialog:** Forældre diskuterer med barnet for at identificere interesser oplevelser, stemninger, hvilket danner grundlag for historiens indhold.
2. **Historiegenerering:** De indsamlede input kobles med de allerede barnets profil (kendte ord og læseniveau) indtastes i ChatGPT, som genererer en personlig historie, der bliver printet ud.
3. **Barnet læser:** Barnet læser den udprintede historie.




Prototype autentificering



Start

AI Miniprojekt 3. semester (Dat22A)
Velkommen til Ord Til Ord
Udført af Martin Grosen Petersen

Start



Username


Email

Password

Registrer dig

Login here

Registrering (Sign-up)



Username

Password

Login

Register here

Login (Sign-In)



Prototype Dashboard

Den nuværende prototype af applikationen præsenterer et enkelt "Dashboard", der indikerer intentionen om at udvikle yderligere funktioner i en fremtidig MVP-version.



Hjem Interesser Ønsker Ord Historier Humør

Dig Ud

Historier

3

Nye Ord

45

Dit Humør

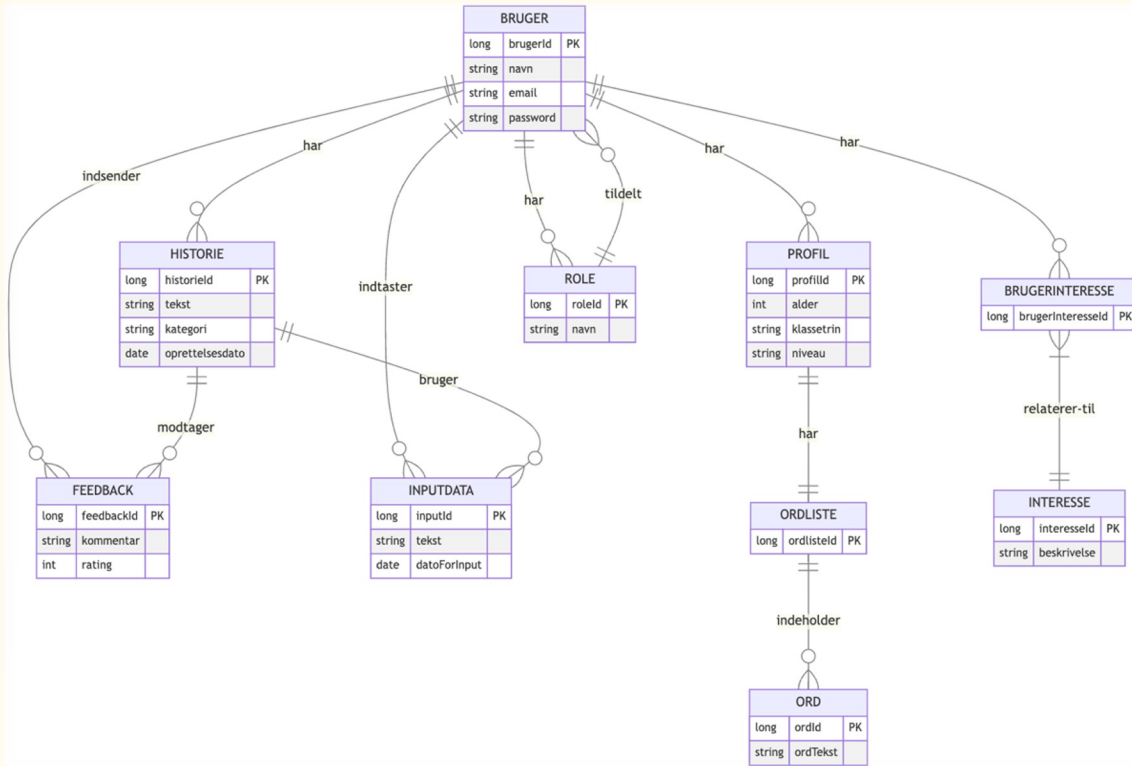
Glad!

Ord Til Ord

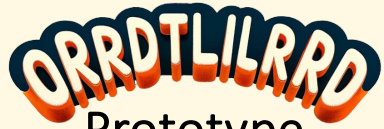
Titel	Historie	Dato	Feedback
Stig og Uheldet med Stikbolden	Stig elsker stikbold. Han spiller hver dag. Han kaster bolden. Han rammer en ven. Vennen siger "Av!" og falder. En ambulance kommer. Vennen kører væk. Stig er ked af det. Han venter. Tiden går. Så ser han vennen igen. Vennen smiler. Han har et plaster på hovedet. "Er du ok?" spørger Stig. "Ja," siger vennen. "Vi spiller snart igen." De griner. De er venner. Stig er glad. Vennen er glad. De spiller igen. Men nu passer de på.	10-Nov-23	God
Farverige Følelser	En dreng blev sur. Han tog maling. Han malede sit hoved. Rød, gul, og blå. "Hvad laver du?" spurgte hun. "Jeg maler," sagde han. "Det gør mig glad." Hun lo. "Nu er du sjov," sagde hun. Han så i spejlet. Han grinede. Han var ikke sur længere. Han havde mange farver på hovedet. "Se mig!" råbte han. Alle lo. Han var som en clown. "Du er god til at male!" sagde hun. Han fælte sig glad. Han malede ikke mere.	8-Nov-23	Medium



Prototype MySQL



- ☐ **BRUGER** har præcis én **PROFIL**.
Dette er en **en-til-en** relation, hvor hver bruger har en unik profil.
- ☐ **BRUGER** kan have mange **HISTORIER**.
Dette er en **en-til-mange** relation, hvor en bruger kan skabe flere historier.
- ☐ **BRUGER** kan indsende mange **FEEDBACKS**.
Dette er en **en-til-mange** relation, hvor en bruger kan give feedback på flere historier.
- ☐ **BRUGER** kan indtaste mange **INPUTDATA**.
Dette er en **en-til-mange** relation, hvor brugerens input kan bruges til at generere flere historier.
- ☐ **BRUGER** har mange **BRUGERINTERESSER**.
Dette er en **en-til-mange** relation, hvor en bruger kan have flere interesser.
- ☐ **BRUGER** kan have mange **ROLES**.
Dette er en **mange-til-mange** relation, hvor en bruger kan have flere roller, og en rolle kan være tildelt til flere brugere.
- ☐ **ROLE** er tildelt til mange **BRUGERE**.
Dette er en **mange-til-mange** relation, hvor en rolle kan være tildelt til flere brugere, og en bruger kan have flere roller.
- ☐ **PROFIL** har præcis én **ORDLISTE**.
Dette er en **en-til-en** relation, hvor hver profil har en unik ordliste.
- ☐ **ORDLISTE** indeholder mange **ORD**.
Dette er en **en-til-mange** relation, hvor en ordliste kan indeholde flere ord.
- ☐ **HISTORIE** bruger mange **INPUTDATA**.
Dette er en **en-til-mange** relation, hvor en historie kan bruge input fra flere dataindtastninger.
- ☐ **HISTORIE** modtager mange **FEEDBACKS**.
Dette er en **en-til-mange** relation, hvor en historie kan modtage feedback fra flere brugere.
- ☐ **BRUGERINTERESSE** relaterer til præcis én **INTERESSE**.
Dette er en **mange-til-en** relation, hvor flere brugerinteresser kan relateres til en enkelt interesse.



Prototype Config

Rollerne; ADMIN og USER bliver oprettet i databasen, når applikationen starter, hvis de ikke allerede findes.

Dette kan være nyttigt for at sikre, at der er nogle grundlæggende brugerroller tilgængelige for applikationens autorisationslogik.

```
package com.ordtilord.mvc.config;

import ...

@Configuration
public class DataInitializer {

    @Bean
    CommandLineRunner initDatabase(RoleRepository roleRepository) {
        return args -> {
            opretRolleHvisIkkeEksisterer(roleRepository, rolleNavn: "ADMIN");
            opretRolleHvisIkkeEksisterer(roleRepository, rolleNavn: "USER");
        };
    }

    2 usages
    @Transactional
    private void opretRolleHvisIkkeEksisterer(RoleRepository roleRepository, String rolleNavn) {
        if (roleRepository.findByNavn(rolleNavn).isEmpty()) {
            RoleEntity nyRolle = new RoleEntity();
            nyRolle.setNavn(rolleNavn);
            roleRepository.save(nyRolle);
        }
    }
}
```

Run: MvcApplication

Console: 2023-11-09 13:59:37.116 INFO 37485 --- [restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect



Prototype Properties

- Opdater database-URL, brugernavn og adgangskode til dine lokale indstillinger.
- Erstat OpenAI API-nøglen med din egen nøgle.
- Juster server.port til en ledig port, hvis nødvendigt.

```
# Database konfiguration
spring.datasource.url=jdbc:mysql://localhost:3306/ordtilord
spring.datasource.username=
spring.datasource.password=
spring.jpa.hibernate.ddl-auto=update

# Spring Security Credentials (hvis du bruger Spring Security)
spring.security.user.name=test
spring.security.user.password=test

# OpenAI API konfiguration
# text-davinci-003
OPEN_AI_URL=https://api.openai.com/v1/completions
OPEN_AI_KEY=

# Server port
server.port=8081
```



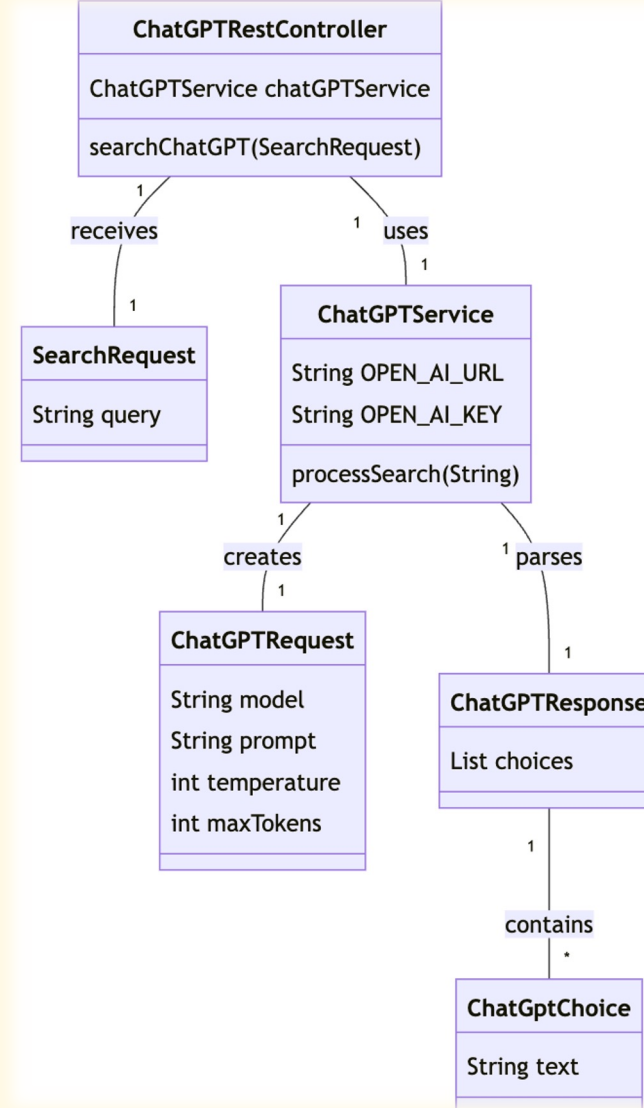
Prototype OpenAI API

Kommunikationen med OpenAI er ikke fuldt integreret i applikationens øvrige struktur, da det er en prototype.

ChatGPTRestController - Dette er en REST controller, der håndterer HTTP POST-anmodninger til /api/v1/searchChatGPT endpointet. Den modtager en SearchRequest, logger forespørgslen og bruger ChatGPTService til at behandle søgningen.

SearchRequest - Denne klasse bruges til at modtage en søgeforespørgsel fra en bruger af applikationen. Den indeholder en enkelt streng, query, som er det brugeren ønsker at spørge eller søge efter.

ChatGPTRequest - Denne klasse bruges til at oprette en anmodning til OpenAI's API. Den indeholder oplysninger som den anvendte model (model), selve forespørgselsteksten (prompt), en temperaturparameter, der styrer determinismen af svaret, og et maksimalt antal tokens (maxTokens), som svaret må indeholde.



ChatGPTService - Dette er servicen, der faktisk kommunikerer med OpenAI's API. Den opretter en ChatGPTRequest baseret på brugerens query, sender anmodningen til OpenAI, og omdanner svaret til et ChatGPTResponse objekt.

ChatGPTResponse - Når OpenAI's API har behandlet anmodningen, returneres svaret som et objekt af denne klasse. Det indeholder en liste af ChatGptChoice objekter, som repræsenterer de mulige svar.

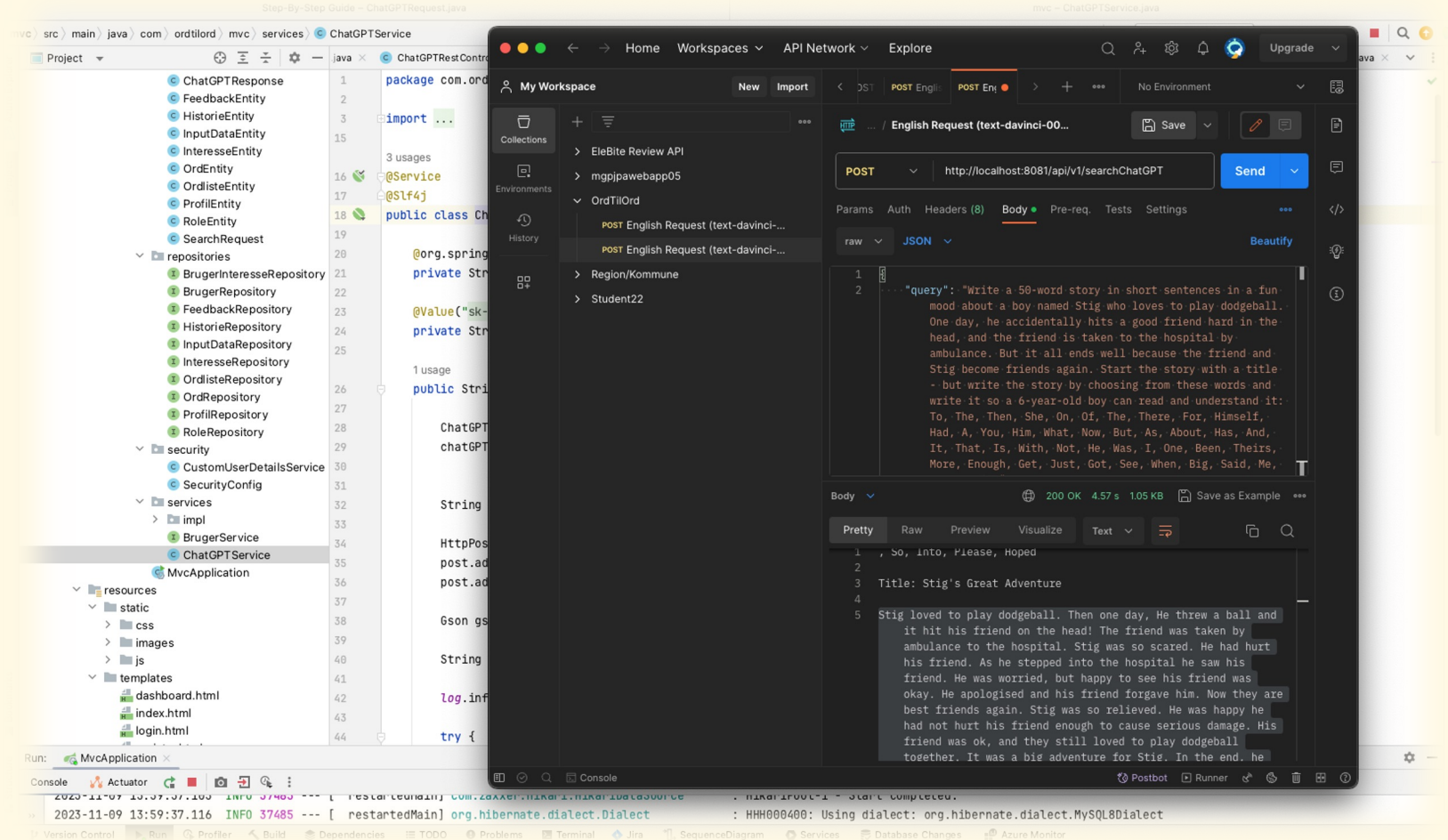
ChatGptChoice - Denne klasse repræsenterer et valg eller et resultat fra OpenAI's API. Det indeholder en tekststreng, som er selve svaret fra API'et.

ORDTLILRD

Prototype Postman

I stedet for at være indlejret i applikationens kode, er kommunikationen med OpenAI's API foreløbigt kun blevet valideret eksternt via Postman.

Den faktiske kommunikation og validering af svarene fra API'et blevet håndteret manuelt gennem Postman, snarere end automatisk gennem applikationen selv.





Tak for opmærksomheden!

DAT22A / 3. semester

Martin Grosen Petersen

<https://github.com/LutherMGP/OrdTilOrd>