**TITTEL & SAMMENDRAG**

**Tittel:** ReviewHub: En webapplikasjon for anmeldelser av filmer.

**Sammendrag:** ReviewHub er en webapplikasjon designet for å tillate brukere å se, legge til, og interagere med filmanmeldelser på en brukervennlig plattform. Prosjektet kombinerer moderne webteknologier for å skape en tilgjengelig og interaktiv opplevelse. Vårt mål var å tilby en plattform der filmentusiaster kan dele sine opplevelser og meninger om filmer gjennom anmeldelser og kommentarer. Ved bruk av HTML, CSS, JavaScript på frontend og ASP.NET Core med en MySQL database på backend, har vi sikret en sømløs brukeropplevelse med sikkerhetsmekanismer gjennom bruk av JSON web token-autentisering. Prosjektet har oppnådd målene om brukervennlighet, samt tilgjengelighet og sikkerhet, noe som reflekteres i systemets evne til å håndtere brukerinteraksjoner og data på en effektiv måte. Vi har også lagt inn mulighet for å utvide til å legge inn anmelder for bøker og TV-serier

.

**INNLEDNING**

**Prosjektbeskrivelse:** ReviewHub er en nettbasert plattform hvor filminteresserte kan lese og dele anmeldelser av filmer. Plattformen tilbyr et samfunn hvor brukere kan bidra og engasjere seg i innhold gjennom funksjoner som å legge til nye filmer med relevant informajson inkludert bildeopplastning(MoviePoster), skrive anmeldelser av filmer inkludert mulighet for bildeopplastning(ReviewPoster), og kommentere på andres anmeldelser.

Alle brukere, uavhengig av om de er innlogget eller ikke, kan se alle tilgjengelige filmer med deres rating og annen relevant informasjon, samt lese anmeldelser og kommentarer. Det er kun funksjoner som involverer interaksjon med systemet som krever innlogging og autentisering med JWT-token, basert på brukernavn og passord.

**Relevans og mål:** I dagens digitale tidsalder er interaksjon og engasjement blant brukere viktige aspekter ved enhver online plattform. Det er mange nettsteder og plattformer som tilbyr muligheten til å vurdere og gi tilbakemelding på filmer og TV-serier, men det er få som tillater brukerne å uttrykke sine meninger gjennom kommentarer på anmeldelser.

Denne funksjonen er på ingen måte original, men fraværende hos de store og etablerte nettstedene som «IMDb» og «Rotten Tomatoes».

Dette er en viktig utfordring som ReviewHub ønsker å takle på en innovativ, brukervennlig og effektiv måte. Ved å tilby et rom for å diskutere og kommentere anmeldelser, går ReviewHub utover det å bare å gi brukerne muligheten til å gi en tommel opp eller tommel ned på en vurdering.

Det var også et mål at brukerne selv skulle ha kontroll over sin digitale identitet. Registreringsprosessen ble derfor implementert på en slik måte at fornavn og etternavn er valgfrie felter. Dette valget reflekterer ønsket om å gi brukerne autonomi og trygghet når de deltar i fellesskapet på ReviewHub.

Ved å tilby kommentarfunksjon, sammen med muligheten for anonymitet, håper ReviewHub ikke bare å tilby en plattform for å vurdere og dele meninger om filmer, men også å dyrke et dynamisk fellesskap hvor lidenskapen for film kan blomstre gjennom åpne diskusjoner og interaksjoner.

**Målgruppe:** ReviewHub er for filmelskere av alle slag. Plattformen er designet for å fange interessen til alt fra kultfilm-entusiaster, til de som like gjerne setter på den nyeste blockbuster-filmen.

**Prosjektmål:** Prosjektets mål inkluderte å skape en enkel og sikker registreringsprosess, muligheten for brukerne å engasjere seg aktivt gjennom anmeldelser og diskusjoner, og å sikre at plattformen var tilgjengelig på tvers av forskjellige enheter og plattformer.

**METODE**

**Utviklingsprosess:** Utviklingen av ReviewHub fulgte en iterativ tilnærming, hvor vi startet med en grunnleggende versjon og gradvis la til funksjonalitet basert på brukertesting (oss selv) og identifiserte behov.

**Frontend:** Utviklingen av frontend brukergrensesnitt og design ble gjennomført ved bruk av HTML, CSS, og JavaScript for å skape en interaktiv og brukervennlig brukeropplevelse. Vi brukte moderne webteknologier for et responsiv design som tilpasser seg ulike skjermstørrelser. JavaScript ble intensivt brukt for å håndtere brukerinteraksjoner og dynamisk innhold. En vesentlig del av frontend-utviklingen var integrasjonen med backend via fetch-metoden, for å sende og motta data asynkront. Dette gjorde det mulig for brukerne å utføre handlinger som å legge til og vurdere filmer uten å måtte laste siden på nytt, noe som forbedrer brukeropplevelsen betydelig.

**Backend-arkitektur:**

Backend-delen av ReviewHub ble utviklet som en kontrollerbasert API ved å bruke ASP.NET Core, etter en lagdelt arkitektur for å sikre modulær kode:

* **Kontrollerlag:** Dette laget håndterte direkte brukerinteraksjoner gjennom HTTP-forespørsler. Det fungerte som en bro mellom brukergrensesnittet og forretningslogikken ved å delegere oppgaver til tjenestelaget og sende svar tilbake til klienten.
* **Servicelag:** Her lå forretningslogikken for applikasjonen, inkludert brukerautentisering og håndtering av filmer, anmeldelser og kommentarer. Tjenestelaget utførte funksjonene som kreves for å administrere datastrømmer og sikre at forretningsregler og prosesser ble korrekt implementert.
* **Repositorylag:** Ansvarlig for all direkte interaksjon med databasen gjennom Entity Framework. Dette ga en abstraksjon av data-laget og forenklet tilgang til databasen uten behov for å håndtere databaseforbindelser eller SQL-spørringer direkte.

**Databaseintegrasjon og migrering:**

ReviewHub bruker MySQL-database gjennom Entity Framework Core for å lagre og administrere data:

Vi definerte nødvendige enheter i API’et, som representerer de ulike dataobjektene i systemet, inkludert brukere, filmer, anmeldelser og kommentarer. Deretter utførte vi migreringer ved hjelp av Entity Framework CLI-verktøy, som genererte de tilsvarende databasetabellene basert på enhetene og deres relasjoner.

**Feilhåndtering og validering:**

For å sikre robust feilhåndtering på globalt nivå, implementerte vi en GlobalExceptionMiddleware som fanget opp unntak som ikke ble håndtert andre steder i applikasjonen og ga en brukervennlig feilmelding tilbake til klienten. Vi brukte også FluentValidation for å validere inndata både under registrering og pålogging.

**Autentiseringssystem:**

Vi utviklet en autentiseringsflyt ved å bruke JSON Web Tokens (JWT) for å administrere økter og sikre API-forespørsler:

Under pålogging blir brukerens legitimasjon validert mot databasen, og passord sikres gjennom hashing ved hjelp av Bcrypt.

Ved vellykket autentisering genererer backend-serveren et JWT-token, som sendes tilbake til klienten og må inkluderes i alle påfølgende API-kall som krever autentisering.

**Samarbeid og arbeidsdeling:**

Vi hadde ukentlige møter der vi gikk gjennom hva vi hadde gjort og planla arbeidet for den kommende uken. Hvis det oppsto noe akutt, sendte vi hverandre meldinger eller ringte. Vi støttet hverandre og hoppet inn der det trengtes, enten det var med frontend-design eller Backend. Vi fordelte oppgavene slik at Robin tok seg av backend-utvikling, databaseintegrasjoner og autentisering med JWT-token, mens Sakaria fokuserte på frontend-design, implementere JavaScript og CSS i tillegg til å bidra til backend-utviklingen når det var nødvendig. Selv om vi hadde våre hovedansvarsområder, jobbet vi begge på både frontend og backend. For å holde styr på koden vår og integrere endringene vi gjorde brukte vi GitHub flittig.

**RESULTATER**

I løpet av prosjektet har vi oppnådd flere resultater som møter de opprinnelige målene vi satte. Disse resultatene inkluderer:

**Autentiseringssystem:** Vårt autentiseringssystem, som anvender JSON Web Tokens (JWT), har spilt en kritisk rolle i håndteringen av brukersesjoner og sikringen av API-forespørsler. Systemet har effektivt sikret brukerdata ved hjelp av hashede passord og JWT-er for å vedlikeholde brukerens sesjon på klienten gjennom localStorage. Dette har tillatt sikker innlogging og kontinuerlig håndtering av brukersesjoner. Det må imidlertid påpekes at selv om JWT-er fungerer sømløst fra frontend til backend, representerer lagringen av disse tokenene i localStorage en potensiell sikkerhetsrisiko. Dette skyldes at tokens lagret i localStorage er tilgjengelige for skript kjørende i nettleseren, noe som kan utgjøre en risiko for cross-site scripting (XSS) angrep. Vi planlegger å forbedre dette ved å migrere til en sikrere lagringsløsning, som bruk av HttpOnly og Secure cookies, for å beskytte mot slike angrep og ytterligere styrke sikkerheten i systemet.

**CRUD-operasjoner:** Vi satte oss som mål å implementere fullstendige CRUD-operasjoner (Create, Read, Update, Delete) for anmeldelser og filmer og kommentarer, slik at brukerne enkelt kunne opprette, lese, oppdatere og slette disse på nettstedet vårt. Til tross for våre anstrengelser, har vi dessverre ikke lyktes med å oppnå dette målet.

En av hovedutfordringene vi støtte på var kompleksiteten ved å jobbe med JavaScript, HTML og CSS, som har begrenset vår evne til å implementere alle delene av CRUD-operasjonene. Dette har ført til mangler i systemet, spesielt når det gjelder muligheten til å oppdatere anmeldelse og oppdatere og slette kommentarer. Oppdater og slett endepunktene er klargjort i backend og er klare til å bli implementert med frontend, men vi hadde dessverre ikke tilstrekkelig tid til å fullføre sammenkoblingen. Å få nettstedet til å se korrekt ut har vært en betydelig faktor i denne begrensningen.

**Responsiv og sikker brukerinteraksjon:** Frontend-utviklingen sikret at applikasjonen var tilgjengelig og optimalisert for ulike enheter, og backend-arkitekturen sørget for sikker og - effektiv datahåndtering. Integreringen av frontend med backend via fetch-metoden forbedret brukeropplevelsen ved å muliggjøre dynamiske oppdateringer uten å laste siden på nytt.

**Feilhåndtering og datavalidering:** Vårt system for feilhåndtering og datavalidering på både frontend og backend har sikret at brukerdata behandles korrekt og sikkert. Dette har redusert forekomsten av feil og sikret at dataene som brukes og vises er nøyaktige og oppdateres korrekt over plattformen.

**Brukertest og evaluering:** Gjennom kontinuerlige brukertester (selvtestet) og iterasjoner, har vi forbedret funksjonaliteten basert på identifiserte behov og tilbakemeldinger. Dette har ført til en stadig forbedring av både brukergrensesnitt og funksjonalitet.

**DISKUSJON**

Gjennom utviklingen av ReviewHub har vi støtt på en rekke utfordringer og lært verdifulle læringsmuligheter som vil forme vår fremtidige tilnærming til lignende prosjekter. En sentral utfordring har vært håndteringen av frontend-teknologiene HTML, CSS og JavaScript. Denne delen av prosjektet viste seg å være mer kompleks enn forventet, spesielt når det gjaldt å sikre konsistent og responsiv design på tvers av ulike enheter og nettlesere

Selv om vi har oppnådd betydelige milepæler i utviklingen av ReviewHub, er det flere funksjoner vi gjerne skulle ha implementert hvis vi hadde hatt mer tid til rådighet. Blant disse funksjonene inkluderer muligheten for å like og dele kommentarer, som ville ha bidratt til å øke brukerengasjementet og interaktiviteten på plattformen.

I tillegg til funksjonene for å like og dele kommentarer, har vi også hatt ambisjoner om å implementere flere andre spennende elementer som ville ha forsterket brukeropplevelsen på ReviewHub. Blant disse er ønsket om å inkludere muligheten til å følge andre brukere, slik at brukerne kan holde seg oppdatert på deres aktiviteter og bidrag på plattformen.

Vi har også vurdert å legge til funksjonalitet for å vise andre brukeres profiler og deres individuelle bidrag, noe som ville ha skapt et mer personlig og inkluderende fellesskap på nettstedet. Videre har vi ønsket å styrke sikkerheten rundt registreringsprosessen ved å implementere en bekreftelse via e-post. Dette ville ha bidratt til å beskytte brukernes kontoer mot uautorisert tilgang og sikret at hver brukerkonto er knyttet til en gyldig e-postadresse.

En annen utfordring vi støtte på var implementeringen av fullstendige CRUD-operasjoner for anmeldelser og kommentarer. Selv om vi ikke lyktes med å fullføre dette målet innenfor tidsrammen, har vi identifisert viktigheten av grundig planlegging og prioritering av oppgaver for å oppnå suksess i lignende prosjekter i fremtiden.

**KONKLUSJON**

På tross av utfordringene har prosjektet vært en lærerik opplevelse. Vi har utvidet vår kunnskap om moderne webteknologier og rammeverk som HTML, CSS og JavaScript, samt forbedret våre ferdigheter innen versjonskontroll og samarbeid gjennom Git og GitHub. Gjennom å delta i et prosjekt av denne størrelsesordenen har vi fått verdifull innsikt i kompleksiteten ved å jobbe med et dynamisk og stadig skiftende kodebase, samt betydningen av å kommunisere og samarbeide effektivt med teammedlemmer for å oppnå felles mål.