Prosjekt Oppgave
QuizMaster 2023

Jøran Lillegård

28. mai 2023

Innhold

1 Introduksjon **1**

2 Teknisk Bakgrunn **1**

2.1 Database . . . 1

2.1.1 Structured Query Language (SQL) . . . 1

2.1.2 Database Management System (DBMS) . . . 1

2.2 ER(Entity-Relationship) Modell . . . 1

2.2.1 Entitet . . . 1

2.2.2 Attributter . . . 1

2.3 Normalisering . . . 1

2.4 Flask . . . 1

2.5 Sikkerhet . . . 2

2.5.1 Cross-site Request Forgery (CSRF) . . . 2

2.5.2 SQL Injeksjon . . . 2

2.5.3 XSS . . . 2

2.6 Kontoinformasjon . . . 2

3 Design **2**

4 Implementasjon **2**

5 Diskusjon **4**

6 Konklusjon **4**

1 Introduksjon

I denne oppgaven går prosjektet ut på å lage et Quiz-nettsted hvor utvikleren skal få innsikt i hvordan en fullstack-utvikler jobber. For dette quiz nettstedet er det en liste over funksjoner som klienten har bedt om. Det må være forskjellige typer spørsmål som enkeltvalg, flervalg og tekst svar. Administratoren må kunne lage quizer og spørsmålene som hører til de forskjellige quizene. Alle spørsmålene må godkjennes før quizen offentliggjøres for de andre brukerne. Klienten ønsket ikke en automatisk rette funksjon, men ønsket i stedet et system der administratoren kan gå inn og legge igjen en kommentar for hvert spørsmål og for hele quizen.

2 Teknisk Bakgrunn

2.1 Database

[**1**] ≪En database er en organisert samling av strukturert informasjon, eller data, vanligvis lagret elektronisk i et datasystem. En database styres vanligvis av et databasestyringssystem (DBMS). Sammen blir dataene og DBMS, sammen med applikasjonene som er knyttet til dem, referert til som et databasesystem, ofte forkortet til bare database.

Data innenfor de vanligste databasetypene som er i drift i dag, er typisk modellert i rader og kolonner i en serie tabeller for å gjøre behandling og dataspørring effektiv. Dataene kan deretter enkelt fås tilgang til, administreres, endres, oppdateres, kontrolleres og organiseres. De fleste databaser bruker strukturert spørringsspråk (SQL) for å skrive og spørre data.(Oversatt)≫

2.1.1 Structured Query Language (SQL)

[**1**] ≪SQL er et programmeringsspråk som brukes av nesten alle relasjonsdatabaser for å spørre, manipulere og definere data, og for å gi tilgangskontroll(Oversatt).≫

2.1.2 Database Management System (DBMS)

[**1**] ≪En database krever vanligvis et omfattende databaseprogram kjent som et databasestyringssystem (DBMS). En DBMS fungerer som et grensesnitt mellom databasen og dens sluttbrukere eller programmer, slik at brukere kan hente, oppdatere og administrere hvordan informasjonen er organisert og optimalisert.(Oversatt)≫

2.2 ER(Entity-Relationship) Modell

[**2**] ≪En ER-modell (Entity-Relationship) er en konseptuell datamodell som brukes til å representere strukturen og relasjonene til data i et databasesystem. Den gir en grafisk representasjon av enhetene (objektene), attributtene (egenskapene) og relasjonene mellom enhetene.(Oversatt)≫

2.2.1 Entitet

[**3**] ≪I et databasestyringssystem (DBMS) er en enhet et stykke data som er lagret i databasen. En enhet kan være en person, sted, ting eller til og med en hendelse.(Oversatt)≫

2.2.2 Attributter

[**4**] ≪I et databasestyringssystem (DBMS) er et attributt et stykke data som beskriver en enhet. For eksempel, i en kundedatabase, kan attributtene være navn, adresse og telefonnummer.(Oversatt)≫

2.3 Normalisering

[**5**] ≪Normalisering er en databasedesignteknikk som reduserer dataredundans og eliminerer uønskede egenskaper som innsetting, oppdatering og slettingsanomalier. Normaliseringsregler deler større tabeller inn i mindre tabeller og kobler dem ved hjelp av relasjoner. Formålet med normalisering i SQL er å eliminere redundante (repeterende) data og sikre at data lagres logisk.(Oversatt)≫

2.4 Flask

[**6**] ≪Flask er et lett WSGI-nettapplikasjonsrammeverk. Den er designet for å gjøre det raskt og enkelt å komme i gang, med muligheten til å skalere opp til komplekse applikasjoner.(Oversatt)≫

2.5 Sikkerhet

2.5.1 Cross-site Request Forgery (CSRF)

[**7**] ≪En metode for å forhindre forfalskning av forespørsler på tvers av nettsteder er å bruke et utfordringstoken som er knyttet til en bestemt bruker og som sendes som en skjult verdi i hvert tilstandsendringsskjema i nettappen.(Oversatt)≫

2.5.2 SQL Injeksjon

[**8**] ≪Er et sikkerhetssårbarhet for nett som gjør at en angriper kan forstyrre spørringene som en applikasjon gjør til databasen. Det lar vanligvis en angriper se data som de normalt ikke er i stand til å hente.(Oversatt)≫

2.5.3 XSS

[**9**] ≪Skripting på tvers av nettsteder (også kjent som XSS) er et sikkerhetsproblem på nett som lar en angriper kompromittere interaksjonene brukere har med en sårbar applikasjon.(Oversatt)≫

2.6 Kontoinformasjon

Administrator
Username: Admin
Password: ImAdmin123
Test User
Username: User
Password: ImUser123

3 Design

Designet er basert på å ha to sider av nettsiden, den ene er brukeren og den andre er administratorsiden. Når en konto logger på systemet vil systemet sjekke dataene mot databasen på serveren. Den vil sjekke brukerkontoen som prøver å logge på om den har administrator rettigheter. Hvis brukeren ikke har rettighetene, vises den vanlige brukersiden der brukeren kan starte en quiz, hvis det foreligger noen offentligjort quizer for brukerne. Når quizen er ferdig og administratoren har rettet quizen, kan brukeren sjekke spørsmålene og svarene som ble gitt eller laste ned en fil som inneholder denne informasjonen. Men brukeren vil ikke få fasiten til quizen.
På administratorsiden er det noen flere menyer, hvor admin kan rette en quiz, lage en ny quiz / spørsmål, oppdatere et spørsmål eller bare slette det. Det er også en side for administratoren for å vurdere quizen og gi administratorrettigheter til en annen bruker.
Når en administrator logger inn og hvis det er en quiz der, må administratoren godkjenne alle spørsmålene i quizen før quizen kan offentliggjøres for de andre brukerne.
Før retting av en quiz kan administratoren stenge quizen, med å gjøre dette vil ingen flere kunne gjøre quizen, og de som holder på med quizen vil kun kunne svare på det spørrsmålet som de holder på med.

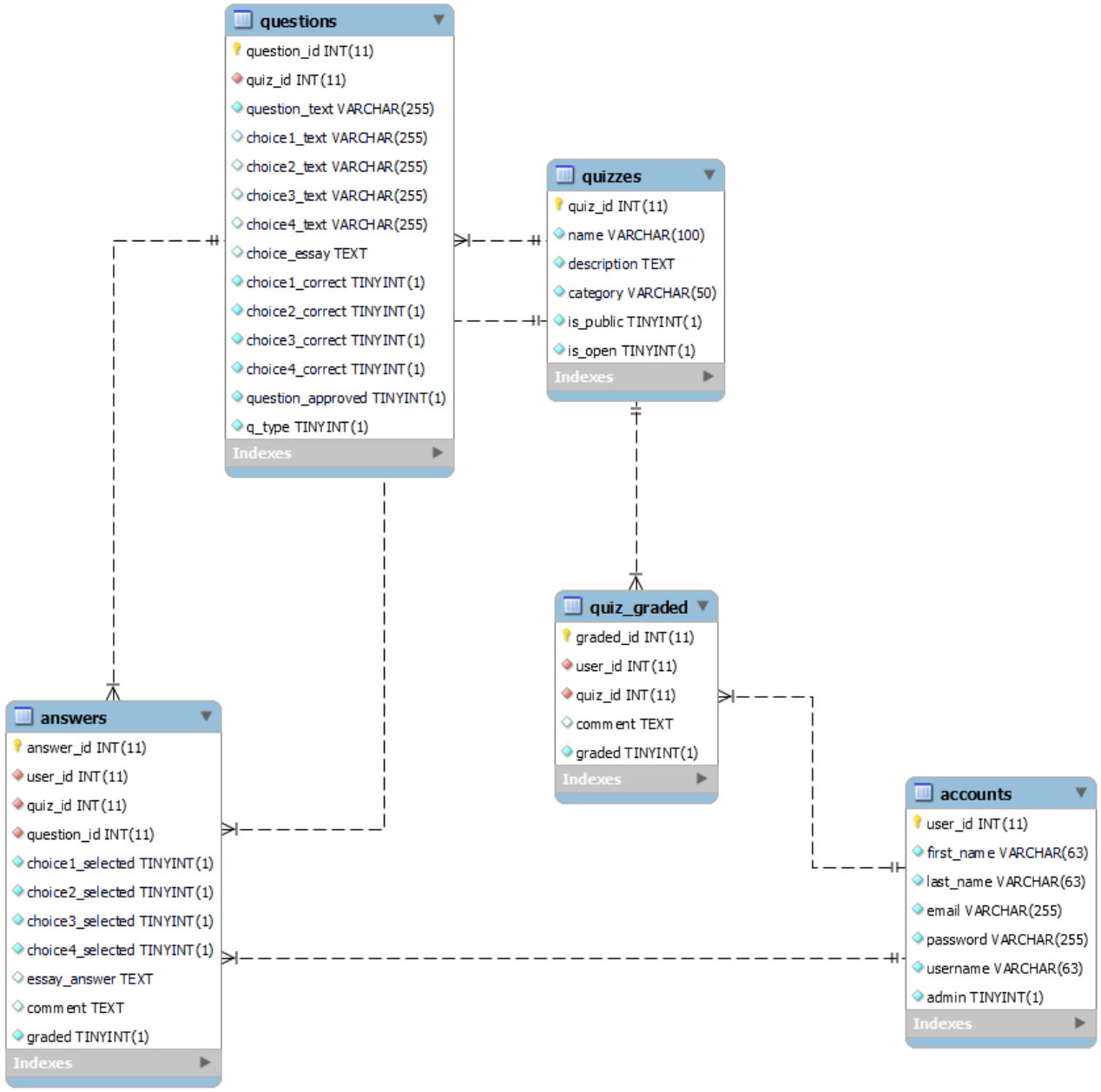
4 Implementasjon

Til backend er modulen for bruk av Flask valgt, Flask er en enkel å bruke backend for å lage webtjenester.
Når man planlegger byggingen av webaplikasjonen, må det være to forskjellige kontroller på serversiden som hindrer vanlige påloggede brukere til å kunne få tilgang til administratorpanelet. For å kunne beskytte mot noen angrep bruker applikasjonen skjemaer med et csrf-token, bruker tuples i SQL-spørringen for å beskytte mot SQL-injeksjoner og bruker forespørselskommandoer i backend for å beskytte mot xss.
Ved planlegging av databasen og tabellene som trengs for prosjektet, har databasen utvidet seg et par ganger. Det er fem tabeller hvor dataene er lagret:

- **Accounts:** inneholder bruker id, brukernavn, navn, epost, passord og om kontoen har Administrator rettigheter.
- **Quizzes:** inneholder quiz id, navn, beskrivelse, kategori, om quizen er offentlig og om den er åpen.
- **Questions:** inneholder spørrsmåls id, hvilken quiz spørrsmålet hører til, hva er spørrsmålet, alternativene, korrekt svar, og hvilken type spørrsmål det er (enkeltvalg, flervalg, tekst oppgave).
- **Answers:** inneholder alle id'ene, hvilket svar som er gitt, kommentar til oppgaven og om oppgaven er rettet.

- **Quiz_Graded:** inneholder id'ene til om quizen er rettet, brukeren og quizen, samt kommentar til hele quizen og om alle spørrsmål er rettet.

Databasen er utformet i henhold til den tredje normalformen (3NF) basert på ER-modellen (se figur 1). Dette betyr at tabellene i databasen er organisert på en måte som eliminerer gjentatte grupper av attributter, og attributtene i hver tabell er avhengige av primærnøkkelen. Ved å følge 3NF sikrer man en effektiv og godt strukturert database. Dette reduserer redundans og unødvendig dataoppbevaring, samtidig som man opprettholder dataintegritet og forhindrer anomalier under oppdatering, innsetting eller sletting av data.



Figur 1: ER-model

For å sikre dataintegritet og beskytte mot angrep som CSRF og XSS, brukes det sikkerhetstiltak. To av disse tiltakene er bruken av CSRF-token og XSS-beskyttelse gjennom skjemainput og argumenter.
Når du oppretter en ny konto vil det bli gjort noen kontroller opp mot databasen, den vil sjekke om e-posten eller brukernavnet eksisterer i systemet, hvis det er tilfelle vil det bli avvist. Passordet vil hashes før det lagres, dette fungerer som en ekstra sikkerhet hvis det skuller evt ha vært et datainnbrudd, slik at angriperen ikke får alle passordene i klartekst.
Når en vanlig bruker logger på, vil brukeren få en side som viser de tilgjengelige quizene hvis det er noen. Når en quiz er tilgjengelig for brukeren kan brukeren svare på alle spørsmålene, og når quizen avsluttes og quizen har blitt rettet kan brukeren laste ned/se spørsmålene og svarene som er gitt samt kommentarene som er lagt til på oppgavene og hele quizen.
Administrator: Det første en administrator vil se er alle quizene, fra listen kan administratoren godkjenne alle spørsmålene med en knapp eller gå gjennom quizen som en vanlig bruker for å godkjenne spørsmålene ett etter ett. Quizen kan ikke settes til offentlig før alle spørsmålene er godkjent. Administratoren kan slette eller åpne/lukke quizen, når quizen er stengt kan ingen nye brukere ta quizen og de som gjør quizen får ikke fulføre flere spørsmål enn det spørmålet brukeren holder på med. Denne funksjonen er laget slik at når administratoren trenger å sette karakter på en quiz eller tiden er ute.
Ved vurdering av en quiz hvis brukeren ikke har svart på alle spørsmålene og hvis admin ønsker å kommentere de ubesvarte spørsmålene, vil den automatisk generere et svar i databasen slik at kommentaren har et sted å koble den til.
Når man lager en ny quiz vil administratoren bli møtt med et skjema for å lage quizen, etter å ha laget quizen må administratoren gå over til legg til/oppdater spørsmålene. Her er alle funksjonene for å kunne lage spørsmålene til quizen. Når du lager et spørsmål, kan administratoren legge inn svaret for det riktige alternativet, men på grunn av at klienten ikke vil ha en automatisk rette funksjon, er denne funksjonen med å merke rett alternativ der kun for at administratoren lettere skal kunne rette spørsmålet senere. Den siste funksjonen er at admin kan gi admin tilgang til andre brukere, ved å gjøre det vil navnet i listen bli tagget med en (ADMIN) tag.

5 Diskusjon

Det første problemet for min del var UI/UX, på grunn av at jeg ikke er så flink med frontend-tingene. Så valget om å bruke myslq.connector i stedet for å bruke sqlalchemy, siden dette kurset er for database, gikk valget til å bruke mysql i stedet for sqlalchemy på grunn av behovet for å skrive SQL. Ellers ville nok jeg ha brukt sqlalchemy for å kunne gjøre denne oppgaven.
Ved å lage quizen slik at admin kan markere riktig alternativ for spørsmålet, dette er litt utenfor spesifikasjonen som klienten ba om, men det vil gjøre rettingen lettere for klienten. Hvis klienten ønsker å få auto rette funksjonen er det ikke så mye som skal implementeres for at det skal fungere. Ved å legge til gi admin-rettighetene, forenkler det bare prosessen, slik at adminen ikke trenger å måtte endre noe på databasen for å gi flere brukere admin-rettigheten.
Implementering av e-postvalideringen er ikke gjort på grunn av at jeg ikke hadde en e-postserver jeg kan bruke, men det er noe som må implementeres før siden publiseres. Dette vil bidra til å forhindre opprettelse av falske kontoer.
Når man legger til en ny bruker er ikke kravene til å opprette et sterkt passord tilstede. Hvis dette skulle brukes for offentligheten, er det også et nøkkelpunkt å implementere en måte å sikre at brukeren oppretter et sterkt passord.
På en sidenotat er det en Funksjon som skjer når du prøver å godkjenne/oppheve godkjenningen av alle spørsmålene. Knappen vil ikke oppdateres slik at siden må oppdateres manuelt for at endringen skal skje på knappen, men oppdateringene skjer i databasen selv om ikke knappen oppdateres. For designvalget som ble gjort på hvordan siden settes opp, kan ui/ux forbedres fordi det er noen ting som kan gjøres mer brukervennlig.

6 Konklusjon

QuizMaster var litt mye arbeid for én person, og det er noen funksjoner"i koden som må strykes ut. Men alt i alt var det et morsomt prosjekt som ga noen virkelig gode leksjoner.

Referanser

[1] Oracle. "What is a database?Besøkt Mai 18, 2023. https://www.oracle.com/database/what-is-database/

[2] Biscobing, Jacqueline. "What is a database? Besøkt Mai 18, 2023. https://www.oracle.com/database/what-is-database/

[3] Sugandhi, Abhresh. Types of Attributes in DBMSBesøkt Mai 18, 2023. https://www.knowledgehut.com/blog/database/entity-in-dbms

[4] Sugandhi, Abhresh. "What is an Entity in DBMS?Besøkt Mai 18, 2023. https://www.knowledgehut.com/blog/database/attributes-in-dbms

[5] Peterson, Richard. "What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples.Besøkt Mai 18, 2023. https://www.guru99.com/database-normalization.html

[6] Flask — Read the Docs.Besøkt on Mai 18, 2023. https://readthedocs.org/projects/flask/

[7] "CSRF Attacks: Anatomy, Prevention, and XSRF Tokens.Besøkt on Mai 18, 2023. https://www.acunetix.com/websitesecurity/csrf-attacks/#:~:text=This%20CSRF%20protection%20method%20is

[8] "What is SQL Injection? Tutorial & Examples — Web Security Academy.Besøkt on Mai 18, 2023. https://portswigger.net/web-security/sql-injection#:~:text=SQL%20injection%20(SQLi)%20is%20a

[9] "What is cross-site scripting (XSS) and how to prevent it?Besøkt on Mai 18, 2023. https://portswigger.net/web-security/cross-site-scripting