pbw_2sem_p2

Dkpbw22a1

Marc Kastholm

Projekt 2. Semester

15.03.2023

GITHUB LINK: https://github.com/Kastholm/quizApp

Indholdsfortegnelse

1. Indledning

I denne opgave blev der udviklet en webapplikation, der tilbyder forskellige quizzer til brugerne. Applikationen er udviklet ved hjælp af Vue.js som frontend-framework, Express.js til at håndtere backend-funktionalitet og MongoDB som databaseløsning. Brugere kan oprette en konto, logge ind, tage quizzer baseret på forskellige kategorier og se deres statistikker. Der er også mulighed for udvalgte brugere at oprette nye quizzer og redigere deres brugeroplysninger. Applikationen sikrer, at kun autoriserede brugere har adgang til bestemte sider og funktioner. Derudover er der udviklet et simpelt API, der giver tredjeparter mulighed for at forespørge quizzer og sortere dem efter forskellige parametre som navn, id, dato og kategori.

2. Problemformulering

Hvordan kan man udvikle en webapplikation ved hjælp af Vue.js, Express.js og MongoDB, der giver brugerne mulighed for at oprette en konto, logge ind, tage quizzer baseret på forskellige kategorier, se deres statistikker, oprette nye quizzer og redigere deres brugeroplysninger, mens der sikres, at kun autoriserede brugere har adgang til bestemte sider og funktioner? Derudover, hvordan kan man implementere et simpelt API, der gør det muligt for tredjeparter at forespørge quizzer og sortere dem efter forskellige parametre som navn, id, dato og kategori?

3. Metodeovervejelser

For at udvikle en effektiv og moderne webapplikation, blev der valgt følgende teknologier og frameworks:

a. Vue.js: Et populært JavaScript-framework, der er kendt for sin fleksibilitet og brugervenlighed. Det blev valgt for at lette opbygningen af den interaktive brugergrænseflade og håndtere klientens logik.
-

b. Express.js: Et minimalt og fleksibelt Node.js webapplikationsframework, der giver et robust sæt af funktioner til web- og mobilapplikationer. Det blev valgt for at håndtere server-siden logik og interaktionen med databasen.

Databasevalg - MongoDB i stedet for MySQL:
Et af de vigtigste valg i forbindelse med udviklingen af webapplikationen var valget af databaseteknologi. Her blev MongoDB valgt i stedet for MySQL af følgende grunde:
-
a. Fleksibel og skemaløs struktur: MongoDB er en NoSQL-database, der lagrer data i fleksible, JSON-lignende dokumenter, hvilket gør det nemmere at arbejde med dataene og tilpasse dem efter behov. Dette kontrasterer med MySQL, der er en relationel database og kræver et fast skema.
-
b. Skalérbarhed: MongoDB er kendt for sin høje skalérbarhed, da det er nemt at tilføje flere servere og opdele data mellem dem. Dette gør det til et godt valg for fremtidig vækst og udvikling af applikationen.
-
c. Bedre integration med JavaScript: Da både Vue.js og Express.js er JavaScript-baserede teknologier, giver MongoDB en mere naturlig integration og en ensartet udvikleroplevelse, da det også arbejder med JSON-lignende dokumenter.

Sikkerhed:
For at sikre en sikker webapplikation blev der truffet foranstaltninger for at beskytte følsomme brugeroplysninger og forhindre uautoriseret adgang til visse dele af applikationen. Dette inkluderer autentificering og autorisering ved hjælp af JSON Web Tokens (JWT), hashing af adgangskoder og korrekt validering af inputdata.

API-design:
For at muliggøre integration med tredjeparter blev der designet et simpelt RESTful API. Dette API giver mulighed for at forespørge quizzer og sortere dem efter forskellige parametre som navn, id, dato og kategori. API'et blev

designet med fokus på enkelthed og lettilgængelighed for eksterne klienter.

4. Research

Inden udviklingen af webapplikationen blev der gennemført research inden for følgende områder for at sikre en solid forståelse og informere designbeslutningerne:

- Teknologier og frameworks:
  Undersøgelse af forskellige alternativer som React, Angular, Vue.js, Express.js, Flask, Django, MySQL og MongoDB førte til valget af Vue.js, Express.js og MongoDB som de mest egnede teknologier for projektet.

- Autentificering og autorisering: JSON Web Tokens (JWT) blev valgt som den primære metode på grund af dens lette implementering, sikkerhed og kompatibilitet med de valgte teknologier.

- Quiz design og struktur: Research førte til valget af et enkelt design med multiple-choice spørgsmål, der er lette at forstå og interagere med for brugerne.

- Brugergrænseflade og brugeroplevelse: Undersøgelse af designprincipper og bedste praksis for at skabe en intuitiv og indbydende brugergrænseflade, der inkluderede farveskemaer, layout, typografi og brug af ikoner og billeder.

- API-design og integration: Research omkring RESTful API-designprincipper, datavalidering og fejlhåndtering hjalp med at oprette et enkelt og letforståeligt API, der er kompatibelt med forskellige klienter og teknologier.

Denne research bidrog til at sikre, at webapplikationen lever op til moderne standarder og brugerforventninger.

5. Analyse

Efter udviklingen af webapplikationen blev der foretaget en analyse for at vurdere applikationens funktionalitet, brugervenlighed, ydeevne og sikkerhed. Analyseprocessen omfattede flere aspekter og krævede, at alle elementer blev vurderet for at sikre, at applikationen opfylder brugernes forventninger og projektets mål.

Brugeroplevelsen og brugervenligheden blev nøje undersøgt ved at vurdere applikationens design, layout, farveskema og typografi samt navigationssystemet.

Dette hjalp med at vurdere, hvor nemt det er for brugerne at interagere med applikationen, finde og tage quizzer og administrere deres kontooplysninger.
Funktionaliteten og fejlfri drift blev også analyseret ved at gennemgå og teste alle funktioner i applikationen, herunder oprettelse af konto, login, quiz-navigation, quiz-oprettelse og administration af brugeroplysninger. Dette sikrede, at applikationen fungerer korrekt og opfylder de fastlagte mål og krav.
Ydeevne og responsivitet blev vurderet ved at teste applikationens evne til at skalere til forskellige skærmstørrelser og enheder, herunder computere, tablets og smartphones. Dette omfattede også at undersøge applikationens hastighed og evne til at håndtere et stort antal samtidige brugere.
Sikkerhed og databeskyttelse blev nøje undersøgt ved at analysere applikationens sikkerhedsforanstaltninger, herunder autentificering, autorisering, validering af inputdata og beskyttelse af brugeroplysninger. Identifikation af potentielle sårbarheder og sikring af, at applikationen beskytter brugernes data og privatliv, var en central del af denne analyse.
Endelig blev API-integration og funktionalitet vurderet ved at teste API'ets evne til at forespørge quizzer og sortere dem efter forskellige parametre samt integration.

Gennem analysen blev der identificeret muligheder for forbedringer og optimering af webapplikationens brugeroplevelse, funktionalitet, ydeevne og sikkerhed.
Denne feedback vil blive anvendt til at implementere nødvendige ændringer og sikre, at applikationen lever op til moderne standarder og brugerforventninger.

6. Konstruktion

Konstruktionsfasen involverede udviklingen af applikationens komponenter og funktionaliteter identificeret under research- og analysefasen.
Følgende trin blev udført:

- Arkitektur og struktur: Opbygning af applikationens overordnede arkitektur ved hjælp af Vue.js og Express.js.

- Brugergrænseflade og navigation: Design af en intuitiv og brugervenlig grænseflade og navigationssystem.

- Autentificering og autorisering: Implementering af JSON Web Tokens (JWT) for sikker autentificering og adgangskontrol.

- Quiz-funktionalitet: Udvikling af mekanismer til oprettelse og gennemførelse af quizzer, inklusive spørgsmålsformulering og svarvurdering.

- Brugeradministration: Implementering af funktioner til opdatering af profiloplysninger og sletning af profiler.

- Statistik og resultater: Opbygning af en statistikside til visning af brugernes quiz-resultater og præstationer.

- API-udvikling: Design og implementering af et RESTful API ved hjælp af Express.js, der håndterer forespørgsler, svar og datavalidering.

- Under konstruktionsfasen blev de forskellige funktioner og krav omsat til en fungerende webapplikation, der lever op til moderne standarder og Brugerforventninger.

7. Evaluering

I starten af projektet var jeg bekymret for, at processen ville blive alt for krævende samt tidskrævende, især da vi alle i min gruppe valgte at arbejde alene.
Dette medførte et ekstra pres, fordi min viden om opbygning af databaser og implementering af datasikkerhed var begrænset.
Men jeg vidste, at jeg arbejder bedst alene og har evnen til at forblive motiveret og fokuseret i lange perioder, hvis ikke hele dagen. Som projektet skred frem, opdagede jeg, at dette ekstra pres faktisk tjente som motivation for mig, og jeg lærte en utrolig mængde ny viden gennem hele processen.
Gennem dette projekt forbedrede jeg mine færdigheder inden for databasedesign, datasikkerhed og webudvikling generelt. Jeg blev konstant udfordret og nød at løse problemer og opdage nye teknikker og værktøjer.
Selvom processen var krævende, føler jeg, at jeg har vokset fagligt som et resultat.

Afslutningsvis er jeg meget positiv omkring hele forløbet.
Selvom det var udfordrende og krævede hårdt arbejde, har jeg lært en masse og udviklet mine evner inden for webudvikling og databaser.
Jeg er glad for det arbejde, jeg har udført, og ser frem til at anvende den viden, jeg har fået, i fremtidige projekter og arbejde.

8. Konklussion

Gennem projektet var der flere aspekter, der gik godt, såsom at arbejde alene og opnå stor personlig udvikling og læring inden for webudvikling og databaser.
Den kontinuerlige udfordring og opdagelse af nye teknikker og værktøjer bidrog til en værdifuld erfaring.

Samtidigt kan jeg nu konkludere at det var et dårligt valg først at kigge ind i Datasikkerhed til sidst, og implementerer JSON webtokens til mit allerede eksisterende login system. Dette forvalte mig mange problemer samt mange forspildte timer.
I sidste ende måtte jeg give op på at få indført json webtokens.

Grundet de mange timer brugt på dette, havde jeg heller ikke mere tid til give yderligere sikkerhed til min applikation.
Jeg ønskede som det mindste at brugeren ville blive slettet i localStorage() efter noget tid , få indført passport.js til login siden samt bruge helmet for at beskytte mine filer. Men jeg må desværre som ene mand se mig slået af tiden.

Sammenfattende har projektet været en lærerig og udfordrende oplevelse, der har bidraget til personlig og faglig vækst. Selvom nogle aspekter, såsom datasikkerhed, kræver yderligere forbedringer, har projektet som helhed været en succes. Den opnåede viden og erfaring kan anvendes i fremtidige projekter for at sikre endnu bedre resultater.

9. Bilag og referenceliste

Vue.js (n.d.). Vue.js Guide. Hentet fra https://vuejs.org/v2/guide/

Express.js (n.d.). Express.js Documentation.
Hentet fra https://expressjs.com/en/starter/installing.html

MongoDB (n.d.). MongoDB Manual. Hentet fra https://docs.mongodb.com/manual/

JSON Web Tokens (n.d.). JSON Web Tokens Introduction.
Hentet fra https://jwt.io/introduction/

OWASP (n.d.). OWASP Top Ten Project.
Hentet fra https://owasp.org/www-project-top-ten/

MDN Web Docs (n.d.). MDN Web Docs.
Hentet fra https://developer.mozilla.org/