Sovelluksen turvallisuus

LanguageApp-sovellus on pyritty kehittämään mahdollisimman turvalliseksi. Tämä on otettu huomioon kehityksessä kautta linjan alkaen siitä, että rekisteröinnissä ei kysytä asiakkaalta ylimääräisiä tietoja. Teknisellä tasolla turvallisuuten liittyen sovelluksessa on seuraavia ominaisuuksia:

MongoDB ja Prisma: Sovellus käyttää MongoDB-tietokantaa, ja Prisma ORM auttaa hallinnoimaan tietokantakyselyjä. Prisma auttaa estämään SQL-injektiot, koska se käyttää parametrisoituja kyselyitä.

Tietokantayhteyden hallinta: Koodi sulkee tietokantayhteyden asianmukaisesti. Tietokantayhteyksien hallinta on tärkeää sekä turvallisuuden että suorituskyvyn kannalta.

Tietokantatunnukset ja -osoite: Nämä on erotettu .env-tiedostoon. Tämä käytäntö estää arkaluontoisten tietojen vuotamisen versionhallintaan ja mahdollistaa niiden helpon hallinnan ympäristömuuttujina.

Bcrypt eli salasanojen kryptaus: Koodi käyttää bcrypt-kirjastoa kryptaamaan käyttäjän salasanan ennen sen tallentamista tietokantaan. Tämä on perustavanlaatuinen turvallisuuskäytäntö, koska se varmistaa, ettei selkokielisiä salasanoja tallenneta tietokantaan, mikä vaikeuttaa hyökkääjien mahdollisuuksia saada käyttäjien salasanoja, vaikka he pääsisivätkin tietokantaan käsiksi.

Salasanan vertailu: Kirjautumisprosessissa koodi käyttää bcryptin compare-funktiota vertaamaan annettua salasanaa turvallisesti tietokantaan tallennettuun tiivisteeseen. Tämä varmistaa, että käyttäjä todennetaan vain, jos salasana vastaa tallennettua tiivistettä.

JWT-tunniste: Onnistuneen todennuksen jälkeen koodi generoi JSON Web Tokenin (JWT), joka sisältää käyttäjätiedot ja allekirjoittaa sen. JWT:t ovat yleinen tapa toteuttaa turvallinen tunnistautuminen, koska ne ovat digitaalisesti allekirjoitettuja ja niitä voidaan validoida palvelimella niiden eheyden varmistamiseksi.

HttpOnly-eväste: Generoitu JWT asetetaan HttpOnly-evästeeksi. Tämä on suositeltava käytäntö, koska se auttaa suojaamaan tunnistetta JavaScriptiltä asiakaspuolella ja estää tiettyjä ristiinsivustoinen skriptaus (XSS) -hyökkäyksiä. Lisäksi evästeellä on rajoitettu voimassaoloaika (1 viikko), mikä parantaa turvallisuutta.

Käyttäjätiedon puhdistaminen: Ennen käyttäjätietojen lähettämistä vastauksessa koodi poistaa salasanakentän, jotta salasanan tiiviste ei paljastu asiakkaalle. Tämä käytäntö on tärkeä arkaluontoisen käyttäjätiedon suojaamiseksi.