# Frontend Ontwikkeling:

HTML: Voor het structureren van de inhoud van je website.

CSS: Voor het stylen van de HTML-elementen en het visueel aantrekkelijk maken van de website.

JavaScript: Voor het toevoegen van interactiviteit aan je website, zoals dynamisch toevoegen van vragen en opties.

# Backend Ontwikkeling:

Database: Om door gebruikers gegenereerde quizzen, vragen, opties en antwoorden op te slaan. Voor de database gaan we Docker gebruiken

ORM (Object-Relational Mapping): Als je een relationele database gebruikt, kan een ORM de database-interacties vereenvoudigen. Voorbeelden zijn Sequelize voor Node.js of SQLAlchemy voor Python.

# Gebruikersauthenticatie en Autorisatie:

Authenticatie: Om gebruikers toe te staan zich aan te melden en in te loggen op je website. Je kunt bibliotheken gebruiken zoals Passport.js (Node.js), Django's ingebouwde authenticatiesysteem, of Auth0.

Autorisatie: Om toegang tot bepaalde functies te regelen, zoals het bewerken of verwijderen van quizzen. Mogelijk moet je op rollen gebaseerde toegangscontrole (RBAC) of machtigingen implementeren.

# Quiz Creatie Interface:

Formulier: Een gebruikersvriendelijk formulier waar gebruikers hun quizvragen, opties en juiste antwoorden kunnen invoeren.

Validatie: Zorg ervoor dat gebruikers alle benodigde informatie verstrekken en valideer invoer om fouten en misbruik te voorkomen.

# Quiz Afname Interface:

Weergave: Een schoon en intuïtief interface voor gebruikers om quizvragen te lezen en te beantwoorden.

Scoreberekening: Automatisch scores berekenen en weergeven bij voltooiing van de quiz.

# Zoek- en Bladerfunctionaliteit:

Zoeken: Gebruikers in staat stellen om te zoeken naar quizzen op basis van verschillende criteria zoals onderwerp, moeilijkheidsniveau of maker.

Filteren: Gebruikers kunnen quizzen filteren op specifieke parameters om gemakkelijker te vinden wat ze zoeken.

Paginering: Als je een groot aantal quizzen hebt, implementeer dan paginering om de prestaties en gebruikerservaring te verbeteren.

# Community Functies:

Reacties en Beoordelingen: Gebruikers kunnen reacties en beoordelingen achterlaten op quizzen om feedback te geven en in contact te komen met andere gebruikers.

Sociaal Delen: Gebruikers kunnen quizzen delen op sociale mediaplatforms om de zichtbaarheid en het bereik te vergroten.

# Schaalbaarheid en Prestaties:

Caching: Implementeer caching mechanismen om de prestaties van vaak benaderde gegevens te verbeteren.

Database Optimalisatie: Optimaliseer database queries en indexes voor efficiënte gegevensopvraging en opslag.

# Testen:

Unit Testing: Schrijf tests om de functionaliteit van individuele componenten van je applicatie te verifiëren.

Integratietesten: Test de interactie tussen verschillende componenten van je applicatie.

Team Analyse:

Bas:

Goed in Html Css Javascript

Slecht in php

Tim:

Goed in Java / Css / Structueel

Slecht in php/Database

Erolcan:

Goed in Mysql / Html Css

Slecht in php

Goed in php / javascript

Quinn:

Slecht in oop