Kats DD, Daan

378969|

Functioneel ontwerp

eindopdracht Webtechnologie II

Inhoudsopgave

Inleiding

Dit functioneel ontwerp beschrijft de context en doelstellingen van een tentamenvolgsysteem. Het systeem zal worden ontwikkeld als een webapplicatie met een moderne PHP-code en een door de student geschreven PHP-framework dat aan de randvoorwaarden voldoet die zijn opgesteld in de opdrachtbeschrijving.

Context en doelstelling

# Context:

Het tentamenvolgsysteem is een applicatie die studenten, docenten en beheerders in staat stelt om de voortgang van tentamens bij te houden. Studenten moeten zich kunnen inschrijven voor tentamens en hun cijfers kunnen inzien. Docenten moeten tentamenresultaten kunnen invoeren en resultaten van tentamens die zij beheren kunnen zien. Beheerders moeten alle onderdelen van het systeem kunnen beheren.

# Doel:

Het doel van het tentamenvolgsysteem is om een gebruiksvriendelijke en veilige omgeving te bieden voor studenten, docenten en beheerders om de voortgang van tentamens bij te houden. Specifieke doelstellingen zijn:

* Het bieden van een gebruikersvriendelijke interface voor het inschrijven voor tentamens, het bekijken van tentamenresultaten en het beheren van het systeem.
* Het bieden van veilige en betrouwbare toegang tot de applicatie en de opgeslagen gegevens.
* Het bieden van een gestructureerde opslag van tentamengegevens voor eenvoudige toegang en bijwerken.

Requirements

Aan de hand van de context en doelstellingen zijn de volgende requirements opgesteld:

* Studenten moeten kunnen inloggen met e-mailadres en wachtwoord.
* Docenten moeten kunnen inloggen met e-mailadres en wachtwoord.
* Beheerders moeten gebruikers kunnen registreren en moeten kunnen inloggen met e-mailadres en wachtwoord
* Studenten moeten zich kunnen inschrijven voor tentamens.
* Docenten moeten tentamenresultaten kunnen invoeren en resultaten van tentamens die zij beheren kunnen zien.
* Beheerders/Docenten moeten nieuwe tentamens kunnen aanmaken, bestaande tentamens kunnen wijzigen of verwijderen en gebruikersrollen kunnen beheren.
* Het systeem moet veilige opslag van wachtwoorden bieden.
* Het systeem moet toegangscontrole bieden op basis van gebruikersrollen.
* Het systeem moet een gestructureerde opslag van tentamengegevens bieden voor eenvoudige toegang en bijwerken.

Use cases

1. Beheerder maakt docenten en student accounts aan

* De student voert zijn/haar gegevens in (naam, e-mailadres, wachtwoord, enz.).
* Het systeem valideert de gegevens van de student en slaat deze op in de database.

1. Gebruiker logt in op het systeem

* De student voert zijn/haar inloggegevens in (e-mailadres en wachtwoord).
* Het systeem valideert de inloggegevens en geeft de student toegang tot zijn/haar account.

1. Student schrijf zich in voor een tentamen

* De student navigeert naar de pagina voor het inschrijven van tentamens.
* De student selecteert het tentamen waarvoor hij/zij zich wil inschrijven.
* Het systeem controleert of de student voldoet aan de vereisten voor deelname aan het tentamen (bijvoorbeeld: ingeschreven voor de bijbehorende cursus).
* Als aan de vereisten is voldaan, slaat het systeem de inschrijving van de student op in de database.

1. Docent voert cijfers in voor een tentamen

* De docent navigeert naar de pagina voor het invoeren van cijfers.
* De docent selecteert het tentamen waarvoor hij/zij cijfers wil.
* Het systeem haalt de lijst op van studenten die zich hebben ingeschreven voor het tentamen.
* De docent voert de cijfers in voor de studenten en slaat deze op in de database.

1. Student bekijkt zijn/haar cijfers

* De student navigeert naar de pagina waarop zijn/haar tentamencijfers worden weergegeven.
* Het systeem haalt de tentamencijfers op uit de database en toont deze aan de student.

1. Docent bekijkt de resultaten van zijn/haar tentamen

* De docent navigeert naar de pagina waarop de resultaten van een tentamen worden weergegeven.
* Het systeem haalt de tentamencijfers op uit de database en toont deze aan de docent.

1. Beheerder voegt nieuwe gebruiker toe

* De beheerder navigeert naar de pagina voor het toevoegen van gebruikers.
* De beheerder voert de gegevens van de nieuwe gebruiker in.
* Het systeem valideert de gegevens en slaat deze op in de database.

1. Beheerder wijzigt gegevens en rollen van gebruikers.

* De beheerder navigeert naar de pagina voor het wijzigen van gebruikersrollen.
* De beheerder selecteert de gebruiker wiens gegevens hij/zij wil wijzigen.
* Het systeem haalt de huidige gegevens van de gebruiker op uit de database.
* De beheerder wijzigt de gegevens van de gebruiker en slaat deze op in de database.

1. Beheerder of Docent voegt een nieuw tentamen toe

* De beheerder/Docent navigeert naar de pagina voor het toevoegen van tentamens.
* De beheerder/Docent voegt de gegevens in voor het tentamen.
* Het systeem haalt de courses op om een tentamen aan te koppelen.