Wiki Procesboek

**31 november:**

Git is compleet nieuw voor mij, en dus begon ik met het schrijven van dit procesboek in word, zonder dat ik nog precies wist waar ik dit bestand uiteindelijk moest gaan uploaden.

Waar ik vandaag achter ben gekomen is dat ik dus dit procesboek gewoon via word bijhoud, en tussendoor moet alles geupload worden in mijn github repository.

**8 December:**

Adventure van programmeren 2 heb ik nu bijna af, en ik ben me ervan bewust dat ik vanaf nu elke dag volle bak aan wiki en commerce moet werken wil ik aan alle eisen voldaan hebben. Vandaag heb ik het wiki project ingeladen op mijn computer en aan de slag gegaan met Design Document. Ik vind het lastig om in te beelden hoe de pagina’s eruit moeten zien en wat ik daarvoor precies nodig heb zonder dat ik zelf geëxperimenteerd heb met hoe alles met Django werkt. Daarom ben ik tijdens het maken van het design document ook al gaan proberen de specificaties te implementeren. Terwijl ik hiermee bezig was moest ik erg vaak de lecture notes van de django lecture erbij pakken. De lecture bevat erg veel nieuwe informatie maar met de notes erbij krijg je hier snel een beter begrip van.

Ik ben gaan proberen om een nieuwe pagina css aan te maken, in dezelfde layout als index.html en die het css.md bestand gebruikt om de info van deze pagina op te halen.

Eerst voegde ik een path toe in de urls.py van encyclopedia: path("CSS", views.css, name="CSS"), waarna ik in view.py van encyclopedia een functie toevoegde die de info van css.md doorstuurt naar css.html. Vervolgens heb ik een css.html file gemaakt met dezelfde layout als index.html. Nu heb ik succesvol een nieuwe pagina aangemaakt wiki/css. Maar op dit moment is deze pagina, los van de gewenste layout, nog zonder informatie.

Waar ik nu tegenaan liep was: Hoe kan ik de css.md file gebruiken voor mijn css.html file.

Na even opnieuw het een en ander gelezen te hebben, kwam ik erachter dat ik door de get\_entry functie te gebruiken in mijn views.py, de md file kan gebruiken.

Waar ik nu tegenaan liep was: Hoe kan ik de css.md file gebruiken voor mijn css.html file. Nu staat alleen de titel nog met hashtag en al in mijn pagina.

Op deze manier zal ik alleen voor elke entry een andere html file moeten maken dus ik ben nu aan het kijken hoe ik dit zo goed mogelijk kan “parametrizen” voor elke mogelijke entry.

Na even oefenen en zoeken is het me gelukt om de entry pages te parametrizen zodat elke mogelijke entry makkelijk bezocht kan worden.

Er is nu een begin van het design document en de readme wil ik proberen morgenochtend geupload te hebben op github. Mijn doel is om dit weekend zover mogelijk te komen met wiki en hopelijk maandagavond het project af te hebben.

Aan het eind van deze dag wou ik alles uploaden op github, maar om git push te kunnen gebruiken moest ik een personal access token aanmaken wat nog niet eens zo makkelijk bleek….

**9 December:**

Vandaag heb ik toegevoegd dat wanneer er een entry ingevuld wordt die niet bestaat, dat er dan een errorpage tevoorschijn komt. Dit heb ik gedaan door een if statement in views.py toe te voegen. Het search gedeelte van de opdracht kost me wel veel moeite. Dit omdat wanneer je een normale form maakt, zoals we bij de opdracht search hebben gedaan, wordt de url: <http://127.0.0.1:8000/127.0.0.1:8000?q=CSS> en niet http://127.0.0.1:8000/CSS . Nadat dit niet lukte heb ik geprobeerd django forms te gebruiken zoals ze in de lecture deden bij add task. Hier ben ik erg lang mee bezig geweest maar helaas werkt dit nog steeds niet.

**12 December:**

Achteraf had ik misschien een branch moeten maken voor het werken aan search, en het proberen van hoe ik dit moest implementeren, maar omdat ik mijn github nog niet aan de praat heb gekregen lukte dit nog niet. Vandaag heb ik met Martijn gekeken naar Github, en blijkbaar liep ik hiermee tegen een probleem aan wat andere studenten een tijdje geleden al hadden, met de SSH-key. Ik ging hierna verder werken aan het search gedeelte van wiki, ik keek onder andere het volgende filmpje, <https://www.youtube.com/watch?v=AGtae4L5BbI> maar ook dit leek me niet verder te helpen. Inmiddels was het 3 uur geweest en was er dus geen TA meer. Toch is het me na ontzettend lang verschillende maniertjes proberen, toch gelukt om uit te vinden waarom het telkens niet werkte. In mijn aangemaakte class voor de forms, had ik de form een naam gegeven: entry\_searched. Ik was me er niet van bewust dat ik daarna in de functie entry.cleaned\_data()[] vervolgens deze entry\_searched moest invullen. Hierdoor had ik een andere variabele telkens ingevuld en returnde de functie niet wat ik wilde. Toch blij dat het uiteindelijk gelukt is, en omdat ik hier lang mee bezig was wist ik inmiddels wel goed hoe ik met django forms moest werken en dus ging de opdracht createpage een stuk sneller dan de vorige. Ik maakte een nieuwe form aan met twee entries: titel en content, waar content van het type textarea was. Hieronder een input die een save button maakt. Vervolgens maakte ik een nieuwe functie aan waarnaar deze informatie gestuurd werd: createdpage. Hier maakte ik met de ingevulde titel en content een nieuwe pagina, waarna de user gelijk naar de aangemaakte pagina doorverwezen werd.

Misschien ga ik de opdracht nog wat mooier maken en fine-tunen, maar op dit moment heb ik aan alle vereisten van de opdracht voldaan en dus ga ik nu verder met Commerce.

**16 December**

Ik ben net klaar met Commerce dus dacht ik laat ik nog even wiki controleren. Ik kwam erachter dat ik voor de entry pagina’s niet wiki/entry had maar alleen de entry naam achter de url, dit heb ik nog aangepast. Ook als ik een pagina probeerde aan te maken die al bestond, schreef hij hem over. In de opdracht stond dat als de gebruiker een pagina probeerde aan te maken die al bestond, degene naar een error page gestuurd zou moeten worden. Dit heb ik ook nog toegevoegd. Nu zijn alle functionaliteiten zoals de opdracht voorschrijft.