Leanboard

Oriëntatiefase

**Wat gaan we deze challenge doen?**Bij deze challenge gaan we een applicatie zoals de vorige challenge. We gaan hier werken met Front end en Back end. Deze verbind je. De applicatie gaat een transactiesysteem voorstellen, zoals PayPal. Hierbij zijn meerdere eisen gesteld die in de front- backend moeten voorkomen.   
  
**Waar halen we ontbrekende informatie vandaan?**Zodra we iets niet weten of ergens vastlopen kunnen we de ontbrekende informatie halen van google. We melden het probleem in de zoekbalk en hiermee kan je de oplossing gemakkelijk vinden.   
  
**Welke code moeten we gaan gebruiken?**Bij de front end moeten we ons voornamelijk gaan focussen op HTML, CSS en Javascript. Bij de backend kan je je het beste gaan focussen op, PHP en Java.   
  
Bronnen:  
  
<https://padlet.com/ictlyceum/sd-lrj-1-kom-een-end-met-front-en-backend-paypalooza-ewx7tuqcv8kwpmlb>   
<https://www.google.nl/>

Leerdoelen  
  
Quintis:  
  
**Technisch leerdoel:** Ik wil aan het eind van de challenge mijn deel van de backend code kunnen schrijven zonder hulp te gebruiken van internet. Ik ga dit doen door mezelf in te lichten over PHP door videos te kijken en de dingen die ik niet begrijp uit deze videos te vragen aan een van de docenten.

**Niet technisch doel:** Ik wil deze challenge zorgen dat ik goed contact kan houden met mijn groep en dat al het werk goed verdeeld gaat zijn zodat we een goed eindresultaat kunnen krijgen en we het beste er van kunnen maken.  
  
Samuel:  
  
**Technisch leerdoel:** Ik wil mij deze challenge gaan focussen als ik ergens tegenaan loop bij het technische deel dat ik vragen stel als ik er niet uitkom. Hierdoor toon ik meer initiatief en leer ik meer. Dit kan ik in samenwerking doen met de docenten.

**Niet technisch doel:** Ik wil net zoals vorige challenge weer geconcentreerd aan het werk gaan. Dit wil ik daarintegen verlengen naar 45 min werken. 10 minuten pauze. Ik merkte dat het de vorige keer goed ging dus ik heb het verlengd. Dit documenteer ik in een bestand.  
  
Jesse:  
  
**Technisch leerdoel:** Ik kan na deze challenge een mockup maken in affinity designer met behulp van kopieren van fotos en gebruik maken van hokjes voor tekst. Binnen 3 weken.  
  
**Niet technisch leerdoel:** Concentratie verbeteren bijhouden hoevajak ik word afgeleid. En daaruit bepalen of het beter gaat dan eerst of tot er nog verbetering aan toe is.

Hedwig:

**Technisch leerdoel:**

ik wil graag op het einde van de challenge weten hoe backend in elkaar zit, en zelf een database kunnen maken. Ik heb nog nooit een database gemaakt en ik ben eigenlijk best enthousiast om het uit te proberen, een database te ontwerpen en te testen.

**Niet Technisch leerdoel:**

ik wil deze challenge graag iets kunnen presenteren, vorige challenge hing ik zo vast dat ik niet het parallel programma had om een windows applicatie te kunnen maken. Dus wil ik deze challege iets kunnen presenteren. En maakt het niet uit als hij niet volledig af is of iets niet helemaal werkt.

**Resultaten**- **Front-end vereisten:**

* **Aanmeld- / Registratiescherm:** hier kunnen nieuwe gebruikers zich registreren of bestaande gebruikers zich aanmelden met hun gebruikersnaam en wachtwoord.
* **Dashboard:** hier krijgt de gebruiker een overzicht van zijn/haar financiële situatie, inclusief grafieken en rapporten van de uitgaven en inkomsten. Er moet ook een lijst met de meest recente transacties van de gebruiker beschikbaar zijn.
* **Transactiescherm:** hier kan de gebruiker nieuwe transacties invoeren of bestaande transacties bewerken. De gebruiker moet het bedrag, de datum, de categorie en eventuele opmerkingen van de transactie kunnen invoeren.
* **Budgetscherm:** hier kan de gebruiker zijn/haar budgetten voor elke categorie instellen en bewerken. Er moet een overzicht worden gegeven van de huidige uitgaven ten opzichte van het ingestelde budget.
* **Categorieënbeheerscherm:** hier kan de gebruiker nieuwe categorieën toevoegen, bestaande categorieën bewerken of verwijderen.

**-Back-end vereisten:**

* Tabel "Users": hier worden alle geregistreerde gebruikers opgeslagen, inclusief hun gebruikersnaam, e-mailadres en wachtwoord (dat veilig gehasht en gezouten moet worden).
* Tabel "Transactions": deze tabel bevat alle financiële transacties die door gebruikers zijn ingevoerd, inclusief de datum, het bedrag, de categorie en eventuele opmerkingen. Elke transactie moet worden gekoppeld aan een specifieke gebruiker, dus er zou een foreign key relatie moeten zijn met de "Users" tabel.
* Tabel "Categories": hier worden alle categorieën opgeslagen die gebruikers kunnen toewijzen aan hun transacties, zoals "Eten", "Huur", "Vervoer", enz. Elke categorie heeft een uniek ID en een naam.
* Tabel "Budgets": deze tabel bevat de budgetten die gebruikers hebben ingesteld voor elke categorie. Een budget heeft een uniek ID, een gebruikers-ID en een categorie-ID, samen met een limietbedrag en een periode (bijvoorbeeld maandelijks of wekelijks).

Taken

* Front end taken:  
    
  **Aanmeld- / Registratiescherm:** HTML en CSS gebruiken. Stylen en juiste kleuren erbij maken
* **Dashboard:** Het dashboard met een overzicht. Ook HTML en CSS gebruiken. Overzichtelijk maken
* **Transactiescherm:** HTML en CSS gebruiken. Ook overzichtelijk
* **Budgetscherm:** HTML en CSS gebruiken. Overzichtelijk budgetscherm dat simpel bruikbaar is
* **Categorieënbeheerscherm:** Buttons maken, waardoor de klant een nieuwe categorie kan maken. HTML en CSS.

Back end taken

* Tabel "Users": Via PHP een tabel maken die alle informatie over users bevat. Uitgebreid maken
* Tabel "Transactions": Database tabel maken waardoor alle transacties bewaart worden bij een bepaalde user
* Tabel "Categories": alle categorieën goed bewaren, waardoor ze niet weggaan nadat de user uitlogt
* Tabel "Budgets": Ook bij deze het budget goed bewaren in een tabel. Hierdoor word dit opgeslagen voor de bepaalde users.