

Sentje

Snel, safe en simpel

Poespasbank

Onze opdrachtgever Poespasbank wil betaalverzoeken kunnen verwerken. Ze zijn geïnspireerd geraakt door apps als bunq.me en tikkie en willen zo snel mogelijk ook die markt op. Aan jullie de taak deze app te ontwikkelen. Verderop vind je de wensen van de klant en de randvoorwaarden van QA. Deze omvatten de essentiële functionaliteit. Voor de rest verwachten we eigen creativiteit om dit aan te vullen tot een volwaardige tikkie-sentje-app.



Assessment

- Je moet tijdens het assessment bruikbare en normale data hebben in je applicatie zodat de onderstaande user stories uitgevoerd kunnen worden. Geen 'blabla' of 'hatsieflats' data.
- Neem bij het assessment ook een zelf ingevuld beoordelingsformulier mee! Zo kan je goed testen of je voldaan hebt aan alle knockouts/eisen.
- Alleen producten waarvan de broncode is ingeleverd via Blackboard worden beoordeeld

Leerdoelen

- De student ontwikkelt, door middel van PHP en hedendaagse (basis) beveiligingstechnieken, een server-side webapplicatie zodanig dat dynamische data on-line getoond kan worden.
- De student maakt gebruik van internationalisering (i18n) en localisatie (l10n) technieken zodanig dat je applicatie in verschillende talen en land specifieke opmaak getoond kan worden.
- De student maakt gebruik van PSR1, PSR2 & PSR4 standaarden zodanig dat de code voldoet aan de voorgestelde afspraken van <https://www.php-fig.org>.
- De student implementeert een API call naar een webservice / Database verbinding zodanig dat de applicatie externe data kan consumeren.
- De student configureert een webserver, met standaardinstellingen, in combinatie met PHP zodanig dat een webapplicatie via HTTP beschikbaar is.

Quality Assurance

- De app moet gemaakt zijn in PHP. Er mogen frameworks ingezet worden voor de frontend. denk aan Bootstrap of Zurb Foundation. Voor PHP wordt voor nu alleen Laravel geaccepteerd maar je kan het ook met puur php maken.
- Bij fouten die normaal gebruik belemmeren wordt het product afgekeurd
- De app moet internationaal in te zetten zijn. Met andere woorden laat zien dat er een vertaling kan plaatsvinden op basis van een instelling (minimaal twee talen/landen, een taal als fallback wanneer browsertaal niet ondersteund)
- De app moet meerdere connecties hebben naar (web)services. Wat is dat dan? Bijvoorbeeld een database maar ook **de Mollie API** (deze heb je nodig voor betalingen). Je kan ook een connectie maken met een weer-API om te kijken of het sneeuwt.. maar dat zou gek zijn. Locatiebepaling is dan handiger.
- Maak je gebruik van Laravel? Dan verwachten we ook correct gebruik hiervan! Dus geen gekke 'raw sql' statements terwijl je de ORM van Laravel kan gebruiken. Lees de documentatie!
- De app moet draaien op je eigen webserver (lokaal of ergens anders) onder een zelfverzonnen url (Apache VirtualHost configuratie)
- De code moet zich aan de PSR-1/2/4 standaarden houden
TIP PSR1 en 2 kun je checken met een statische analyse tool als php codesniffer: `phpcs --standard=PSR1,PSR2 /path/to/eindopdracht`
https://github.com/squizlabs/PHP_CodeSniffer
- De applicatie moet een MVC architectuur bevatten
- De applicatie is volgens OO principes geschreven

Functionele eisen

- Als gebruiker wil ik meerdere rekeningen kunnen beheren (aanmaken, inzien, verwijderen)
- Als gebruiker wil ik een betaalverzoek kunnen aanmaken en verwijderen als er nog geen betalingen hebben plaatsgevonden op het verzoek
- Als gebruiker wil ik een betaalverzoek kunnen delen.
suggestie: unieke link etc.
- Als gebruiker wil ik een overzicht van bekende personen welke in (tijdelijke) groepen ingedeeld kunnen worden (handig tijdens een etentje)
- Als gebruiker wil ik anderen geld laten doneren (en/of zelf laten bepalen welk bedrag diegene overmaakt)
tip: bunq.me
- Als gebruiker wil ik mijn rekening overzicht exporteren en extern opslaan (google drive/dropbox of vergelijkbaar)
Opslag in platte tekst (JSON, CSV)
- Als gebruiker wil ik met intervallen betalingen uitvoeren/inplannen. Dit betekent dat je een 'kalender' functie in je app krijgt waarin je de verschillende betalingen kan inplannen. Afhankelijk van je instellingen (internationaal) kan de kalender er anders uit zien.
- Een betaling moet minimaal de volgende informatie bevatten: Tijdstip, persoon, beschrijving, bedrag, valuta (optioneel locatie).
- Een betaling kan worden verrijkt met extra informatie zoals een notitie of een afbeelding (rekening)
- Als gebruiker wil ik betalingen maken / ontvangen in andere valuta. De valuta is in stellen bij het betaalscherm en wordt na betaling omgerekend naar Euro's. Bonuspunten voor een koppeling met een valuta converter service.
- Als gebruiker wil ik dat mijn wachtwoord, rekeningen en gegevens beveiligd zijn