

GLITCH

GAMIFIED LEARNING INTERFACE THROUGH CHALLENGES AND HEURISTICS

Platformconcept

Het GLITCH-platform (Gamified Learning Interface Through Challenges and Heuristics) is een innovatief, thema-onafhankelijk leerplatform ontworpen om onderwijs op een dynamische en interactieve manier te benaderen. Het platform gebruikt een modulaire structuur om een breed scala aan vakken en cursussen te huisvesten, elk georganiseerd binnen specifieke domeinen om de navigatie en specialisatie voor gebruikers te vergemakkelijken.

Projectoverzicht

Achtergrond

Het ontwikkelteam krijgt als opdracht het creëren van GLITCH, een gamified, cross-platform leeromgeving ontworpen om studenten via uitdagingen en heuristieken een diepgaand en interactief educatief traject te bieden.

Concreet betekent dit het ontwikkelen van een React Native applicatie die compatibel is met zowel iOS als Android, met een achterliggende infrastructuur ondersteund door een (SQL) database volgens het MVC-architectuurpatroon. De applicatie moet gebruikers authenticatie via een username/password, of OAuth van de HR, bieden, een mogelijkheid tot het uploaden van profielfoto's direct vanaf de smartphone, en een persoonlijke profielsectie met essentiële gebruikersinformatie. Daarnaast wordt er een dynamisch dashboard verwacht waarop gebruikers hun voortgang binnen het platform kunnen volgen, en een admin dashboard voor het beheren van gebruikersgegevens en spelcontent. Het team moet ook push-notificaties implementeren voor relevante updates en

herinneringen. De documentatie van het project, inclusief API-specificaties en gebruikershandleidingen, dient in Markdown opgesteld te worden en beschikbaar te zijn op een GitHub-repository.

Om een **Module** binnen het GLITCH-platform te implementeren, moet het ontwikkelteam beginnen met het definiëren van de **Kernopdracht** die de centrale leerdoelstelling of het project van de module vormt. Deze Kernopdracht dient als de basis waaromheen **Activiteiten** worden georganiseerd; elke Activiteit is een stap of taak die studenten moeten voltooien om dichterbij het afronden van de Kernopdracht te komen. Elke Activiteit moet worden ontworpen met vier progressieve **Niveaus** van moeilijkheid of diepgang, om een geleidelijke leercurve te faciliteren. Voor de implementatie in het dashboard moet het team een intuïtieve gebruikersinterface ontwikkelen waar studenten gemakkelijk hun huidige progressie binnen elke Activiteit en de algehele voortgang van de Kernopdracht kunnen zien. Bovendien moeten er twee soorten **Challenges** worden geïntegreerd: de **Point Challenge**, waar studenten punten verzamelen door Activiteiten te voltooien, wat leidt tot het ontgrendelen van aanvullende modules; en de **Concept Challenge**, die studenten uitdaagt om kritisch na te denken en de geleerde concepten op creatieve wijzen toe te passen voordat ze aan de Kernopdracht beginnen. Deze elementen moeten naadloos samenwerken binnen het dashboard om een motiverende, educatieve ervaring te bieden die zowel kennis als vaardigheden opbouwt.

Doelstelling

Voor dit project gaan wij de basis van het spelplatform ontwikkelen.

Dit platform zal bestaan uit de volgende onderdelen:

- Een website voor de speler/student waar hij zijn voortgang kan zien en de volgende opdrachten kan vinden.
- Een mobile app voor de speler/student waar hij zijn voortgang kan zien en de volgende opdrachten kan vinden.
- Een centrale database waarin de voortgang van de studenten wordt opgeslagen.

- Een webapplicatie voor de docent waar hij de voortgang van de studenten kan zien en de opdrachten kan beoordelen.

Deze onderdelen zullen worden ontwikkeld met behulp van React voor de frontend en React native voor de mobile app.

Voor de backend en database is het aan het team om te bepalen welke technologieën zij willen gebruiken.

Ontwikkel een veelzijdig, thematisch onafhankelijk platform dat educatieve opdrachten faciliteert doormiddel van een spelomgeving.

Het prototype spel zal sterrenstelsels, sterren, en planeten gebruiken als metaforen voor de verschillende onderdelen van de cursus en de bijbehorende opdrachten.

Er mag gekozen worden voor een ander thema.

User Journey

Stap 1: Registratie en Login

- De gebruiker opent de GLITCH-app en ziet een welkomstschermb.
- De gebruiker selecteert 'Registreren' en voltooit de OAuth-authenticatieprocedure met zijn/haar HR-login of gebruikt een username en wachtwoord.

Stap 2: Dashboard Overzicht

- Na succesvolle login, landt de gebruiker op het hoofddashboard, waar een overzicht van beschikbare Cursusdomeinen te zien is.
- De gebruiker kiest een Domein dat bij de gebruiker hoort, bijvoorbeeld 'Software Development (SWD)'.

Stap 3: Cursus en Instantie Selectie

- Binnen het Domein 'SWD', verkent de gebruiker de lijst met beschikbare Cursussen en kiest er een, bijvoorbeeld 'User Experience'.
- De gebruiker kiest vervolgens een Instantie (2023-2024/OP3) van de cursus om te starten, gebaseerd op het kwartaal of semester.

Stap 4: Verkennen van Modules

- In de gekozen Instantie ziet de gebruiker een reeks Modules die voltooid moeten worden.
- Elke Module wordt gepresenteerd met een korte beschrijving en een voortgangsindicator. De gebruiker kiest een Module om te starten.

Stap 5: Kernopdracht en Activiteiten

- Binnen de Module, ziet de gebruiker de Kernopdracht en begint met de eerste Activiteit.
- Elke Activiteit heeft vier Niveaus; de gebruiker start bij Niveau 1 en werkt zich omhoog door de niveaus door de taken te voltooien.

Stap 6: Controleren en Afvinken door Docent

- Na het voltooien van bepaalde Activiteiten of de gehele Kernopdracht, dient de gebruiker een verzoek in voor controle bij de docent.
- De docent controleert de ingeleverde taken en, indien goedgekeurd, vinkt deze af in het systeem waardoor de voortgang van de gebruiker wordt bijgewerkt in het spel.

Stap 7: Voortgang en Challenges

- Naast de reguliere Activiteiten, neemt de gebruiker deel aan de Point Challenge en Concept Challenge voor extra uitdagingen en beloningen. Dit gaat automatisch.
- Het succesvol voltooien van deze Challenges draagt bij aan het ontgrendelen van aanvullende modules en het verbeteren van de spelervaring.

Stap 8: Profiel en Community Interactie

- De gebruiker bezoekt regelmatig zijn/haar profielpagina om foto's, bio en voortgang te updaten.
- De gebruiker deelt prestaties met vrienden op het platform en reageert op updates van anderen met emoticons.

Doelgroep

De primaire doelgroep bestaat uit studenten die vakken volgen gerelateerd aan softwareontwikkeling.

Structuur en Benamingen

Domain

Het hoogste organisatieniveau binnen GLITCH, waarbinnen diverse educatieve gebieden worden gecategoriseerd, zoals Software Development, IOT, Service Management, enzovoorts. Elk Domein omvat een reeks van Cursussen.

Course

Courses (vakken) vormen de specifieke studiegebieden of onderwerpen binnen een Domein. Een Cursus is een brede educatieve categorie die één of meer Instanties bevat, waarbij elke Instantie een aparte uitvoering van de Cursus vertegenwoordigt.

Instance

Een Instantie refereert aan een specifieke aanbieding van een Cursus binnen een bepaald tijdframe, zoals een kwartaal, semester of schooljaar. Instanties bieden de mogelijkheid om de Cursusinhoud te herhalen of te actualiseren voor verschillende groepen studenten. Binnen elke Instantie zijn er verschillende Modules.

Module

Binnen een Instantie van een Cursus bestaan er verschillende Modules. Elke Module focust op een subonderwerp of specifiek leerdoel binnen de bredere context van de Cursus. Een Module omvat een Core Project, diverse Activities en de Point Challenge en de Context Challenge.

Core Assignment

Het Core Assignment vormt de kernopdracht binnen een Module en is ontworpen om de integratie en toepassing van kennis en vaardigheden te toetsen die zijn opgedaan door het voltooien van de Activities. Te vergelijken met een opdracht.

Activities

De specifieke taken of opdrachten binnen een Module die studenten moeten voltooien om zich voor te bereiden op het Core Assignment. Elke Activity omvat vier niveaus van moeilijkheid, ontworpen om progressieve leerervaringen te bieden.

Challenges

Point Challenge

Een spelelement dat studenten motiveert door het verzamelen van punten, die worden verdiend door het voltooien van Activities op verschillende niveaus binnen Modules. Het behalen van de Point Challenge kan leiden tot het vrijspelen van extra Modules of Courses binnen het Domein.

Concept Challenge

Een voorwaarde voor toegang tot het Core Assignment van een Module. De Concept Challenge vereist dat studenten een diepgaand begrip tonen van de concepten geleerd in de Activities. Het succesvol voltooien ervan zorgt ervoor dat studenten klaar zijn voor de complexiteit en uitdagingen van het Core Assignment.

Door deze benadering worden studenten niet alleen gemotiveerd door het vrijspelen van nieuwe leerinhouden en het bereiken van mijlpalen, maar worden ze ook aangemoedigd om een diepgaand begrip en meesterschap over de leerstof te ontwikkelen. GLITCH bevordert zo een leeromgeving waar progressie hand in hand gaat met persoonlijke en academische groei.

Technische Specificaties

Platform

Type: Native app (iOS en Android) en webapplicatie m.b.v. React native.

Technologieën: React (voor de frontend), met een MVC-modelarchitectuur bestaande uit een modellenlaag, controllerlaag en viewlaag.

Database: SQL of NoSQL oplossing.

Onafhankelijkheid van Thema: Het platform moet zo worden ontworpen dat thema's gemakkelijk kunnen worden aangepast of gewijzigd zonder de onderliggende structuur te beïnvloeden. Dit geldt voor de naamgeving, de opmaak en de grafische elementen.

Interfaces: Frontend voor de gebruiker als speler. Backend voor de docent voor administratie en een samenvatting in een dashboard.

Operations: Docker compose configuratie file waarmee de gehele applicatie opgezet kan worden.

Ontwikkelingseisen

MVC Model Implementatie

Modellenlaag: Verantwoordelijk voor het aanspreken van de database.

Controllerlaag: Bevat de logica van het spel, authenticatie, en filtering van content.

Viewlaag: De frontend native App, ontwikkeld React. Backend voor administratie en dashboard overzicht (React, HTML, CSS).

Interface

Speler

- De interface is twee ledig. Voor de speler is er de interface als persoonlijk dashboard of profiel waarbij de speler zijn eigen voortgang kan terugzien.
- Ook kan de speler hier vinden welke volgende speelelementen er beschikbaar zijn om uit te voeren.
- De speler moet hier het gevoel krijgen wat zijn of haar voortgang is en duidelijk weten wat de volgende stappen zijn binnen het spel.

Beheerder

- Voor de docent is er een backend nodig waarbij de docent, per speler, de gegevens kan updaten. Hier moet de activiteiten, per niveau, kunnen worden afgevinkt.
- Ook moet er een beoordeling met feedback gegeven kunnen worden per core assignment.
- Daarnaast moet er een samenvatting en overzicht gegeven worden per speler en van alle spelers bij elkaar.

MOSCOW

Must Haves

- **Gebruikersregistratie en authenticatie:** Veilige OAuth-authenticatiemechanismen of username/password moeten worden geïmplementeerd om nieuwe gebruikers te registreren en bestaande gebruikers toe te staan in te loggen met hun HR-gegevens of eigen wachtwoord.
- **Dynamisch gebruikersdashboard:** Een intuïtief dashboard dat gebruikers een overzicht biedt van beschikbare cursusdomeinen, hun voortgang binnen geselecteerde cursussen en toegang tot modules.

- **Module- en activiteitenselectie:** Een systeem waarmee gebruikers gemakkelijk modules binnen een cursusinstantie kunnen verkennen en selecteren, inclusief de mogelijkheid om de voortgang van activiteiten en de kernopdracht te bekijken.
- **Voortgangvalidatie door docent:** Functionaliteit waarmee gebruikers voltooide activiteiten en kernopdrachten ter controle en goedkeuring kunnen indienen bij docenten, die op hun beurt de voortgang binnen het systeem kunnen bijwerken.
- **Point Challenge en Concept Challenge:** Implementatie van gamification-elementen die gebruikers motiveren door het verzamelen van punten en het aangaan van conceptuele uitdagingen, wat bijdraagt aan het ontgrendelen van nieuwe modules.
- **Profielbeheer:** Mogelijkheid voor gebruikers om hun profiel te personaliseren, inclusief het uploaden van een profielfoto en het bijwerken van persoonlijke en academische informatie.
- **Notificatiesysteem:** Een systeem voor push-notificaties om gebruikers te informeren over belangrijke updates, deadlines en feedback van docenten.
- **Admin- en docentendashboard:** Een webgebaseerd dashboard voor docenten en beheerders om gebruikersgegevens, cursusvoortgang en andere administratieve taken te beheren.
- **MVC Architectuurpatroon:** Implementeer het Model-View-Controller (MVC) patroon voor duidelijke scheiding van de applicatielogica, gebruikersinterface en inputlogica.
- **(SQL) Database:** Ontwikkel een (relationele) database voor opslag van gebruikersdata, cursusinformatie, voortgang, etc.
- **React Native App:** Ontwikkel een cross-platform app met React Native voor zowel iOS als Android, inclusief geoptimaliseerde prestaties en responsiviteit.

Should Haves

- **Admin dashboard:**
 - Aanpassen van spelgegevens.
 - Exporteren van gegevens naar Excel of CSV.
 - Thema selector voor de interface van instanties.
- **Analytics en rapportage:** Tools voor zowel gebruikers als beheerders om inzicht te krijgen in gebruiksstatistieken, voortgangsanalyse, en engagement metrics.

- **Licht-/donkermodus:** Een instelling die gebruikers toestaat om tussen een lichte en donkere modus van de app-interface te wisselen, afhankelijk van hun voorkeur of omgevingslicht.
- **Deelfunctie voor voortgang:** Functionaliteit waarmee gebruikers hun voortgang binnen het platform kunnen delen met andere gebruikers of via externe sociale media.
- **Interactie binnen de community:** Features zoals een vriendenlijst, de mogelijkheid om prestaties van vrienden te bekijken en te reageren op hun voortgang, versterken de sociale interactie en het gemeenschapsgevoel op het platform.
- **Integratie met Discord:** Een koppeling met Discord voor updates en community-engagement, wat een extra kanaal biedt voor communicatie en samenwerking tussen gebruikers.
- **Emoticons voor reacties:** Sta gebruikers toe om te reageren op gedeelde voortgang met emoticons.
- **Veiligheid en privacy:** Sterke beveiligingsmaatregelen voor de bescherming van gebruikersgegevens en privacy, inclusief naleving van relevante wetgeving zoals GDPR

Could Haves

- **Delen op social media:** Sta gebruikers toe om hun voortgang op externe social media platforms te delen.
- **Meerdere thema's:** Ontwikkel meerdere thema's voor de app-interface.
- **Thema selector:** Bied een thema-selector aan voor gebruikers om de app interface te personaliseren.
- **Automatische updates:** Zorg voor een automatische updatefunctie.
- **Achtergrondmuziek:** Voeg achtergrondmuziek toe.
- **Visual animaties:** Implementeer animaties voor een aantrekkelijkere gebruikerservaring.
- **Gebruikersfeedback en beoordelingssysteem:** Mechanismen voor gebruikers om feedback te geven op cursusinhoud of -ervaring en content te beoordelen.
- **Toegankelijkheid en inclusiviteit:** Zorg ervoor dat het platform voldoet aan toegankelijkheidsnormen (zoals WCAG) om een inclusieve leeromgeving te garanderen.

Won't Haves

- **Chatfunctie:** Een directe chatfunctie tussen gebruikers zal niet worden geïmplementeerd.

- **Custom thema's:** De mogelijkheid voor gebruikers om eigen thema's te creëren en aan te passen is niet voorzien.

De must haves zijn hierin dus de **Minimale functionele eisen**. Hieraan moet je applicatie dus voldoen voor een voldoende.

Documentatie

Een gestructureerde en toegankelijke documentatie is belangrijk voor zowel het ontwikkelteam als de eindgebruikers. Voor GLITCH moet de documentatie bestaan uit:

- **Technische documentatie:** Bevat details over de architectuur van het systeem, codeconventies, API-referenties, en setupgidsen. Deze documentatie is primair bedoeld voor ontwikkelaars en technische beheerders.
- **Bronnenvermelding:** Een sectie met gebruikte bronnen. Denk hierbij aan gevolgde tutorials, Gelezen boeken & code snippets

De documentatie moet worden geschreven in Markdown voor eenvoudige leesbaarheid en onderhoud. Markdown maakt het ook eenvoudig om de documentatie op GitHub te plaatsen en te integreren met Git-versiebeheer voor updates en revisies.

Planning en Resources

Een voorlopige planning en een overzicht van benodigde middelen moeten worden opgesteld om de ontwikkeling te structureren en te begeleiden.

Graphics en andere visuele elementen kunnen worden aangeleverd.

Bijlagen

Repository Structuur

De volgende structuur wordt aanbevolen voor de GitHub-repository van het project.

```
/GLITCH-platform  
  /docs  
    architecture.md  
    api-reference.md  
    setup-guide.md  
  /src  
  /tests  
  README.md
```