**Documentație Tehnologii proiect Cloud Computing- Bookaholic**

Aplicatia consta intr-un magazin online de carti ce va oferi diferite functionalitati pentru utilizatori normali cat si pentri utilizatori premium.

**I.Frontend**

Pe partea de interfata, tehnologiile folosite vor fi HTML, CSS, Javascript, Bootstrap, Jquery.

Aplicatia va fi multipage app.

Utilizatorul isi va putea crea un cont nou, sau se va putea loga cu un cont de microsoft. Dupa logare, acesta va fi redirectat catre o pagina de bun venit, de unde poate selecta ce vrea sa vada in continuare: pagina cu carti, chatbot pentru recomandare de carti, sau isi poate upgrada contul la un cont premium.

**II.Backend**

Backendul va fi implementat in NodeJS, fiind utilizat Express ca framework de dezvolate a aplicatiei web. Frontendul va fi o aplicatie headless, in asa mod incat backendul va fi in fapt un REST Api ce indeplineste diferite sarcini necesare in Front End. Cu toate acestea, va fi util si un server dedicat acestei aplicatii in mod exclusiv, deoarece avem nevoie de o stare a conexiunii ce nu poate fi simplu modelata intr-un REST Api (autentificare, securizare etc.). Acest server, pe langa servirea paginii web, va fi folosit in a indeplini sarcini ce necesita autentificarea userului. Pentru acest aspect vom folosi Passport impreuna cu metodele de logare de la Microsoft, cat si metoda locala pentru a valida o autentificare.

Am aceste tehnologii deoarece:

* exista o usurinta in a folosi a NodeJS, o documentatie consistenta si un suport excesiv oferit de cloud pentru aceasta tehnologie.
* NodeJS reprezinta limbajul deja cunoscut de multi dezvolatori ai acestei aplicatii
* conform modelului aplicatiei, vor exista multe requesturi de dimensiune mica: de manipulare a datelor, de autentificare sau de payment, model potrivit pentru NodeJS.
* Express este o alegere naturala in cadrul dezvoltarii unei aplicatii web in NodeJS
* Passport este de asemenea o tehnologie adecvata in cadrul ecosistemului NodeJS cand vine vorba de autentificare

Evident, vor fi folosite servicii cloud pentru a indeplini unele sarcini ale API-ului. Majoritatea dintre ele sunt dedicate suportului REST API-ului, cum ar fi API Management service ce va fi folosit pentru a documenta API-ul. Acesta este necesar incat va exista si o parte de monetizare in ideea utilizarii acestui serviciu ca un CAAS; o metoda de vinde informatii acumulate de catre aplicatie principala. Pe de alta parte REST Api-ul va fi cu siguranta publicat in cardul unui App Service deoarece:

* exista suport nativ pentru API Management Service cand vine vorba de API-uri expuse prin App Service
* hostingul este necesar evident pentru astfel de API, iar ecosistemul aplicatiei este reprezentat de platforma Azure

**III.Machine Learning**

Pentru a putea face recomandari de carti cat mai precise si mai aproape de dorintele utilizatorilor folosim un serviciu de ML. Fiecare utilizator, in functie de tipul de utilizator, are un numar de recomandari pe saptamana. Utilizatorul isi poate alege un set de criterii de cautare, un anumit autor, a fost publicata intr-un anumit an, apartine unui anumit gen, si asa mai departe. Toate aceste criterii si id-ul utilizatorului sunt date ca argument serviciului nostru pentru predictie.

Serviciul pentru predictie are doua componente principale:

1. Algoritmul de ML care face predictiile
2. Partea de selectie a predictiilor in functie de criteriile utilizatorului

Algoritmul de ML face 2 tipuri de recomandari:

1. recomanda carti similare cu cele care au fost evaluate pozitiv de catre utilizator
2. recomanda carti care au fost evaluate pozitiv de catre alti utilizatori similari cu utilizatorul nostru

Serviciul nostru pentru recomandari este o aplicatie scrisa in Python ce are expus un endpoint API care poate fi apelat printr-o metoda POST.

Algoritmul de ML este implementat folosind platforma Microsoft Azure machine Learning Studio. Aceasta platforma face posibila usor lansarea algoritmului ca un serviciu web, care poate fi apelat printr-o metoda POST. Toate partile algoritmului exceptand conexiunea si interogarea bazei de date sunt scrise in Python. Bibliotecile principale folosite sunt: pandas si numpy pentru a putea lucra cu date cat mai multe si variate sub forma de dataframe, sklearn pentru a putea invata modelul.

**IV.Payment**

Pentru a accesa features precum Adviser, Blog, user ul va trebui sa si faca upgrade la cont.

Tehnologiile utilizate pentru plata vor fi Node JS-frameworkul Express, Request si Paystack API.

Modulul Request va fi utilizat pentru a face request uri catre payment gateways.

Paystack API endpoints:

1.Initializare tranzactie

<https://developers.paystack.co/v2.0/reference#initialize-a-transaction>

2. Verificare tranzactie

<https://developers.paystack.co/v2.0/reference#verify-transaction>