

Bardita Larisa  
Bucnaru Raluca  
Tulbure Lucia

## Task-Descriere arhitectură

### 1.MVC

“Majoritatea aplicațiilor Web sunt dezvoltate conform MVC (Model-View-Controller).” (Curs Tehnologii WEB)

**Model:** se referă la datele ce vor fi stocate și regulile, restrângerile asupra acestora. Oferă reprezentarea datelor existente și se ocupă de validarea acestora. Pentru stocarea datelor ne vom folosi de MySQL, un sistem de gestiune a bazelor de date relaționale. Am ales MySQL deoarece acesta este foarte des folosit împreună cu limbajul PHP.

**Modelarea datelor și proveniența lor:** Există date ce provin din mai multe locuri. Vor exista date introduse de client, de exemplu pentru login, register, testele făcute de client spre a fi rezolvate de alți utilizatori. În plus, aplicația noastră se va folosi de date provenite de pe site-uri precum GitHub și site-uri de conferințe, evenimente, acestea având ca obiectiv principal discuția despre diferite limbaje de programare spre a îmbunătăți înțelegerea participanților într-un anumit domeniu. Preluarea datelor din repo-urile de pe GitHub se va face cu ajutorul API-ului de la GitHub. Preluarea datelor de pe site-urile de evenimente și integrarea în site-ul nostru se va face cu ajutorul unor http requests. Putem atașa unor site-uri tag-uri care să ne ajute să îi arătăm fiecărui utilizator evenimente cât mai relevante. Totuși cel mai ușor ar fi să folosim un recommendation engine pentru a oferi recomandări. În plus, pentru a localiza utilizatorul, vom folosi Geolocation API pentru a-i oferi recomandări în zone cât mai apropiate de acesta, iar pentru a genera statistici vom folosi Google API.

Pentru datele introduse de utilizator, vom folosi sistemul de baze de date relaționale pentru a le stoca. Totuși parola va fi criptată folosind o funcție hash sau funcții special create în MySQL pentru criptarea datelor (<https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html>).

**View:** furnizează metode de reprezentare a datelor furnizare de model. Aceasta este interfața vizibilă utilizatorilor. Pentru view am folosit HTML și CSS pentru a crea diferite pagini destinate utilizatorilor, pagini prezentate la componenta anterioară a proiectului.

**Controller:** se ocupă de cererile provenite de la client. Vom avea un server PHP Apache. Aplicația PHP va integra elemente și din alte limbaje, precum JavaScript, pentru a genera, de exemplu, unele dintre statisticile existente în partea de frontend a aplicației.

UserController se folosește de token pentru a menține un user conectat și se ocupă cu cererile provenite din partea userului, precum login, register, verificare token, etc.

TestController are ca model teste și se ocupă de funcționarea corectă a interacțiunii dintre client și testele, fie scrise de acesta ce urmează a fi salvate în baza de date, fie preluate din baza de date și afișate utilizatorului.

ActivitiesController se ocupă cu generarea de statistici.

KnowledgeController se ocupă cu adăugarea sau preluarea de evenimente pentru a fi afișate utilizatorului.

EventsController este folosit pentru a căuta evenimente pe baza unui keyword.

## 2. Tehnologiile folosite

Pentru stocarea datelor vom folosi sistemul de gestiune a bazelor de date relaționale MySQL. Ca limbaj de programare ne vom axa pe PHP și vom integra elemente din alte limbaje precum JavaScript pentru a genera Google Charts.

**API-uri preluate:** Pentru a ne folosi de datele de pe GitHub vom prelua API-ul de la GitHub pentru a accesa informațiile despre fiecare repository. Pentru a afla localizarea utilizatorului vom folosi Geolocation API, API utilizat pentru a oferi recomandări în zone cât mai apropiate de fiecare utilizator. Pentru generarea unor statistici ne vom folosi de API-ul de la Google, API pe care îl integrăm cu ajutorul unui script.

**Tehnologii ce pot fi folosite:** o altă metodă de afișare a recomandărilor poate fi prin folosirea unui recommendation engine și un search engine (când utilizatorul dorește să caute anumite evenimente).

**Alte API-uri:** vom implementa un REST API. Rest Api utilizează limbajul javascript pentru a crea, citi, modifica și șterge informații dintre diferite aplicații. De fiecare dată când o aplicație se poate conecta la internet și are nevoie de alte aplicații pentru a modifica datele, un Rest Api este de cele mai multe ori folosit

**Alte tehnologii ce pot fi folosite:** putem folosi diverse librării pentru a ne facilita munca, librării precum php Markdown sau Buzz.

## 3. Funcționalități-câteva dintre ele

**Login:** utilizatorii se vor putea autentifica imediat din pagina de pornire. După ce introduce username-ul și parola, se va verifica dacă utilizatorul există în baza de date și dacă există, i se va afișa pagina principală a utilizatorilor autentificați.

**Register:** utilizatorii ce nu se află încă în baza de date, se pot înregistra. După ce introduc datele și se verifică să nu mai existe aceeași adresă de e-mail sau același username, parola va fi criptată și datele vor fi stocate în baza de date. Din acest moment, utilizatorul se poate autentifica.

**Stabilirea nivelului la un anumit limbaj de programare:** nivelul unui utilizator se va stabili pe baza informațiilor de pe GitHub și unor teste create pentru diferite limbaje de programare, fiecare limbaj având trei dificultăți. Se va folosi o funcție ce va stabili punctele la un anumit limbaj. De exemplu testele contează 40% și GitHub 60%. Răspunsurile la teste reprezintă un anumit punctaj,etc.

**Generare statistici:** există două moduri de generare a statisticilor. În primul rând sunt statisticile pe baza a ceea ce a făcut pe GitHub, apoi statisticile pe baza evoluției sale de când s-a înregistrat, până în momentul actual. Ultimele statistici vor fi generate pe baza a Google Charts.

**Token:** menține utilizatorul conectat.

**Requesturi http:** folosite pentru a genera recomandări pentru utilizator, se vor folosi taguri pentru a face asocierea dintre limbajul cunoscut și nivelul pentru acel limbaj. De obicei ar trebui să fie afișate conferințe de același nivel sau un nivel puțin mai avansat, dacă utilizatorul este aproape să avanseze la următorul nivel.

**Scrierea de teste:** Un utilizator poate scrie teste pentru limbajele știute, la un nivel maxim egal cu al său. Testul va fi șters la un anumit număr de dislike-uri din partea altor utilizatori. Se va urmări ca unui utilizator să nu i se afișeze întrebări introduse de acesta.

**Adăugare linkuri evenimente la care a participat+feedback:** utilizatorul poate adăuga evenimente la care a participat și să vizioneze paginile evenimentelor ce s-au sfârșit și la care a luat parte.