### UNIVERSITATEA DIN BUCUREȘTI

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

# DEZVOLTAREA UNEI APLICAȚII WEB PENTRU ÎNVĂȚAREA DE LIMBI STRĂINE

### Absolvent Scobiola Radu

**Coordonator științific**

**Lect. Univ. Dr. Mihăilescu Marius Iulian**

**București, iunie 2023**

#### Rezumat

La nivel global, aproximativ 40% din populația planetei nu știe să vorbească decât într-o singură limbă. În comparație, 43% din oameni știu să vorbească fluent în două limbi diferite, însă doar 17% sunt vorbitori fluenți în cel puțin trei limbi.

În momentul de față, aproximativ 1,2 miliarde de oameni studiază o limba străină, deci există un număr foarte mare de persoane care sunt în căutare de cât mai multe metode si resurse diferite pentru a le ajuta cu procesul de învățare.

Pentru aceasta vom realiza o aplicație web care va încerca să facă procesul de învățare cât mai ușor, dar și interactiv pentru a oferi o experiență cât mai plăcută fără a sacrifica calitatea cunoștiințelor oferite. Aplicația va fi realizată în Angular pentru partea de front-end și va utiliza Firebase pentru partea de back-end.

#### Abstract

On a global scale, approximately 40% of the world’s population knows how to speak just one language. By comparison, 43% of people know how to fluently speak two languages, but only 17% are fluent in at least three different languages.

Right now, approximately 1.2 billion people are studying a foreign language, so there is a very large number of people in the search for as many different methods and resources as possible to help them with the learning process.

For this reason, we will create a web application which will attempt to make the learning process as easy as possible, while also interactive to offer an experience as pleasant as possible without sacrificing the quality of the information offered. The application will be made in Angular for the front-end and for the back-end we will use Firebase.

# Cuprins

1. [Introducere](#_bookmark0) 4
   1. [Motivați](#_bookmark1)e 4
   2. Obiectivele proiectului 4
   3. Descrierea aplicației și a funcționalităților principale 5
2. Metodologie 6
   1. Studierea altor aplicații 6
      1. Identificarea elementelor pozitive 6
      2. Identificarea nevoilor utilizatorilor 8
   2. Definirea specificațiilor aplicației 9
      1. Tehnice 10
      2. User experience 11
3. Aplicația web 12
   1. Tehnologiile utilizate 12
      1. HTML 12
      2. CSS 13
      3. TypeScript 14
      4. Angular 8
      5. Firebase 8
   2. Procesul de dezvoltare a aplicației 9
      1. Front-end 10
      2. Back-end 11
      3. Legătura dintre front și back 11
4. Prezentarea aplicației 7
5. Concluzii 8

[Bibliografie](#_bookmark6) 9

# Capitolul 1

# Introducere

## Motivație

Învățarea de limbi străine a fost dintotdeauna un subiect care m-a pasionat foarte mult și mereu am căutat metode cât mai eficiente, dar și interesante totodată pentru a învăța cât mai ușor. Scopul principal al acestei lucrări este de a crea o aplicație care oferă lecții cât mai interactive și ușor de urmat, fără a renunța însă la calitatea acestora. De asemenea, în plus față de lecții, aplicația va oferi multe alte resurse folositoare pentru a învăța o limbă străina mai ușor.

## Obiectivele proiectului

Obiectivul principal al acestei lucrări este de a crea o aplicație web care în primul rând are o interfață simplă și estetică, pentru a oferi o experiență cât mai placută posibil utilizatorului. De asemenea, aplicația va încerca să reprezinte o alternativă pentru oamenii care au încercat să învețe limbi străine folosind alte astfel de aplicații. Pentru aceasta, vor fi implementate metode sau resurse pentru învățare ce nu se regăsesc în alte aplicații de învățare a limbilor străine.

## Descrierea aplicației și a funcționalităților principale

Aplicația va fi realizată în Angular și va avea următoarele pagini principale:

* **Pagina de register/login,** care va fi o pagină simplă pentru crearea unui cont/logarea într-un cont deja existent.
* **Pagina principală (dashboard-ul),** unde utilizatorul își va putea edita contul și alege limba maternă și limba pe care își dorește să o învețe.
* **Pagina cu resursele de învățare.** Aici utilizatorul va putea alege din diferitele metode de învățare.
* **Pagina cu exercițiile,** unde utilizatorul va avea o listă cu diferite categorii de exerciții din care va putea alege, fiecare având un anumit nivel de dificultate și testând cunoștiințe diferite.

# Capitolul 2

# Metodologie

## Studierea altor aplicații

Pentru a putea realiza o aplicație care să folosească metode moderne de învățare am ales să studiez cele mai populare două aplicații pentru învățarea limbilor străine: Duolingo și Babbel. Apoi am strâns mai multe recenzii ale utilizatorilor pentru a vedea care erau lucrurile principale pe care le apreciau la aceste aplicații și care erau lucrurile pe care le displăceau cel mai mult la ele. Astfel, cu ajutorul acestor informații a fost mult mai ușor de identificat ce anume ar trebui să implementeze aplicația și ce lucruri ar trebui să evite, sau să încerce să îmbunătățească. Site-urile de review-uri folosite au fost Trustpilot și GetApp, oferind mii de recenzii atât pozitive cât și negative pentru ambele aplicații. Dat fiind numărul mare de recenzii, am putut extrage principalele avantaje și dezavantaje ale fiecărei aplicații.

## Identificarea elementelor pozitive

În primă fază am dorit să identific care sunt lucrurile pe care aceste două aplicații le fac cel mai bine și care sunt motivele principale pentru care au atât de mulți utilizatori. Știind aceste lucruri, va fi mult mai ușor să ne asigurăm că aplicația are cele mai importante funcții. Aspectele principale pozitive pentru aplicații au fost următoarele:

Duolingo:

* Lecțiile sunt scurte, interesante și captivante.
* Modul de structurare al lecțiilor este asemănător unui joc, ceea ce creeaza o experiență distractivă.
* Competiția săptămânală și structurarea acesteia într-un sistem de ligi diferite oferă o motivație în plus pentru a învăța.
* Versiunea gratuită a aplicației oferă multe, cea platită nu este absolut necesară pentru învățarea unei noi limbi.

Babbel:

* Explicații foarte amănunțite, informațiile sunt apoi fixate cu ajutorul exercițiilor.
* Multe noțiuni teoretice, mai asemănător cu un curs clasic.
* Funcție de revizuire pentru utilizator, permițându-i acestuia să parcurgă din nou un curs anterior și să dea iarăși testele în cazul în care își dorește să repete anumite noțiuni.
* Explică gramatica unei limbi.
* Oferă exemple de dialog întâlnite în viața de zi cu zi.

Astfel, aspectele principale ale aplicațiilor care duc la o experiență bună a utilizatorului au fost identificate ca fiind următoarele:

* Lecții scurte, dar care să includă noțiuni teoretice.
* Un echilibru între o lecție interactivă și distractivă și una informativă.
* Lecții ce învață mai ales cuvinte sau fraze care ar fi de cel mai mult folos într-un dialog. Oferirea de exemple de limbaj colocvial.
* Abilitatea unui utilizator de a revizui lecțiile și exercițiile trecute.

## Identificarea nevoilor utilizatorilor

## 

După identificarea aspectelor pozitive ale celor două aplicații, următorul pas a fost de a identifica principalele funcții care le displac utilizatorilor pentru a le evita în implementarea aplicației. De asemenea, au fost identificate și anumite funcții pe care utilizatorii și le-ar dori să existe. Principalele critici aduse aplicațiilor sunt următoarele:

Duolingo:

* Structura gramaticii nu este explicată din punct de vedere teoretic, ca și utilizator trebuie să o memorezi pe de rost.
* Aplicația nu este foarte bună pentru a învăța cum să porți un dialog într-o limbă străină.
* Multe cuvinte noi care apar direct în exerciții, fără a fi explicate înainte într-un modul teoretic.
* Lipsa de flexibilitate în privința ritmului de învățare. Utilizatorii nu pot alege să meargă direct la alte lecții dacă deja consideră că stăpânesc suficient de bine anumite concepte.

Babbel:

* Unele lecții sunt plictisitoare, ceea ce tinde să ducă la pierderea concentrării și a interesului utilizatorului.
* Lecțiile sunt scurte și apar între exerciții. Nu există o listă în care utilizatorul să poată vedea toate lecțiile la un loc pentru a învăța. Singurul mod în care utilizatorii pot învăța noțiuni teoretice este prin a alege să facă exerciții.
* Conținutul cursurilor diferă în funcție de limba nativă a vorbitorului, ceea ce duce la o lipsă de echilibru. Unele cursuri au mult mai multe informații pentru unele limbi native decât pentru altele.
* Nu are și o versiune gratuită a aplicației, doar un free trial pentru o săptămână.

În urma acestor recenzii, am stabilit care ar fi cele mai importante lucruri de evitat în crearea aplicației. De asemenea, am observat și care sunt funcțiile principale pe care utilizatorii și le-ar dori de la o astfel de aplicație, însa lipsesc. Acestea sunt:

* Prezența explicațiilor teoretice, pentru ca și procesul de învățare să aibă loc tot pe aplicație, nu doar cel de exersare.
* Punerea la dispoziție a tuturor resurselor de învățare pentru a fi descărcate sau accesate de către utilizator oricând are acesta nevoie de ele.
* Libertatea utilizatorului de a învăța în orice ordine dorește acesta și în propriul său ritm.
* Menținerea unei uniformități privind calitatea cursurilor indiferent de limba nativă a utilizatorului.

## Definirea specificațiilor aplicației

În urma identificării funcționalităților care sunt cele mai apreciate și dorite de către utilizatori, și cele care ar trebui evitate, a urmat să stabilim care vor fi specificațiile aplicației.

Acestea sunt împărțite în două mari categorii. Prima dată, vom stabili care sunt specificațiile tehnice ale aplicației, care sunt funcționalitățile principale ale acesteia. Apoi vor fi stabilite si aspectele aplicației ce țin de user experience, de interfața pe care o are utilizatorul și de funcționalitățile implementate pe care le va avea acesta la dispoziție.

## Tehnice

În primă fază, va fi nevoie de un sistem pentru logare și autentificare a utilizatorilor atunci când aceștia doresc să acceseze conținutul aplicației. Pentru aceasta vom realiza o bază de date în Firebase care va memora toate conturile create, fiecare cu propriul său ID, adresă de email, și parolă. De asemenea, vor fi reținute și alte date care ar putea fi de folos administratorilor, cum ar fi data la care fiecare cont a fost creat și când a fost ultima dată accesat.

După aceasta, va trebui implementat un dashboard, unde utilizatorul își va putea gestiona contul. Acesta va avea opțiunea de a-și schimba parola și poza de profil. De asemenea, aici el își va putea alege limba nativă, fapt ce va schimba limba peste tot în aplicație.

Pe lângă acestea, vor fi implementate sisteme de a randomiza exercițiile ce apar în cadrul unui test precum și opțiunea de a descărca materialele puse la dispoziție pentru învățare.

Progresul fiecărui utilizator va fi salvat și stocat de asemenea în baza de date pentru ca acesta să nu fie pierdut.

Pentru a gestiona interacțiunea cu baza de date Firebase vom folosi Firebase SDK și API-uri. Acestea ne vor permite să efectuăm operații de scriere și citire în baza de date, precum și să gestionăm autentificarea utilizatorilor.

De asemenea, vor fi implementate măsuri de criptare a datelor sensibile ale utilizatorilor pentru a asigura securitatea acestora. Vom utiliza practici recomandate de securitate, cum ar fi verificarea și validarea datelor de intrare pentru prevenirea atacurilor de tip SQL injection.

Nu în ultimul rând, performanța bună a aplicației va fi unul dintre obiectivele principale. Pentru a asigura acest lucru vom implementa tehnici de optimizare precum încărcarea tardivă a resurselor și minimizarea fișierelor. Aceste lucruri vor reduce timpul de încărcare al aplicației și îi vor îmbunătăți viteza de răspuns.

## User experience

Pentru a oferi o experiență cât mai placută utilizatorilor, vom acorda o atenție deosebită tuturor aspectelor ce țin de user experience (UX) în cadrul aplicației noastre. Iată aspectele ce le vom lua în considerare atunci când realizăm design-ul aplicației.

În primul rând, design-ul aplicației trebuie să fie intuitiv și coerent. Interfața aplicației va fi ușor de înțeles și de utilizat, astfel încât utilizatorii să poată naviga prin aplicație fără niciun fel de dificultăți. Vom folosi elemente de navigare familiare și des întâlnite pentru a asigura acest lucru și consistența aplicației.

Fluxul de utilizare al aplicației va fi unul simplu, ușor de urmărit de către utilizatori. Vom evita supraîncărcarea cu informații sau opțiunile inutile pentru a oferi a o experiență cât mai fluidă și ușor de urmărit.

Utilizatorii își vor putea urmări progresul în timp real pentru a le oferi o imagine de ansamblu asupra tuturor lucrurilor pe care le-au învățat. Acest lucru îi va motiva să continue să progreseze.

Vom implementa notificări și mesaje clare, atât pentru eventuale erori, cât și pentru performanța utilizatorului în cadrul unui test. Oferind feedback în timp real, utilizatorii vor înțelege mai bine interacțiunile lor cu aplicația.

Ne vom asigura că aplicația este complet responsivă și adaptată la toate tipurile diferite de dispozitive. Unul dintre cele mai importante aspecte este ca aplicația să arate bine și să funcționeze corect indiferent de dispozitivul de pe care este accesată, sau de dimensiunea ecranului acestuia.

Vom folosi o paletă de culori consistentă în cadrul întregii aplicații împreună cu fonturile și elementele grafice specifice. Astfel, nu doar că dăm o identitate a aplicației, dar vom forma o atmosferă plăcută și prietenoasă pentru un mediu de învățare, luând în considerare psihologia culorilor. De asemenea, menținerea unui design consistent va duce la familiarizarea în timp a utilizatorilor cu aplicația. Unele dintre cele mai mari nemulțumiri ale utilizatorilor apar atunci când o aplicație își schimbă complet design-ul cu care erau aceștia obișnuiți.

# Capitolul 3

# Aplicația web

## Tehnologiile folosite

În realizarea aplicației au fost folosite mai multe tehnologii pentru a realiza diferite aspecte ale acesteia. Am implementat diferite limbaje de programare, framework-uri și platforme în comuniune unul cu celelalte pentru a realiza o aplicație completă, ce oferă o experiență placută. În continuare vom prezenta fiecare tehnologie în parte ce a fost folosită în cadrul lucrării.

## HTML

HTML (HyperText Markup Language) este un limbaj de marcare utilizat pentru crearea și structurarea conținutului paginilor web. Este un element fundamental și esențial al dezvoltării web și reprezintă fundația pe care este construită o pagină web. Am folosit cea mai recentă versiune a limbajului și anume HTML5.

Principala funcție a HTML este de a defini structura paginii și de a-i organiza conținutul. În esență, HTML se folosește de etichete (sau tag-uri) pentru a delimita diferite secțiuni și elemente ale unei aplicații web. Acestea sunt mai apoi interpretate de către broswer și afișate în mod corespunzător.

Un document HTML este realizat dintr-o de serie de elemente, fiecare având o etichetă de început și una de încheiere. Există o multitudine de etichete diferite, fiecare având roluri diferite, pentru a defini tipuri diferite de conținut. Pe lângă etichetele clasice, cum ar fi <**title**>, <**div**>, <**p**>, există multe etichete diferite pentru funcții mai specifice. Dintre acestea amintim etichetele pentru imagini (<**img**>), link-uri (<**a**>), liste (<**ul**>, <**ol**>, <**li**>), tabele (<**table**>, <**tr**>, <**td**>) și formulare și butoane (<**form**>, <**input**>, <**button**>).

Pe lângă etichetele predefinite, HTML oferă și opțiunea de a crea etichete personalizate prin intermediul atributelor **class** și **id**. Acestea permit dezvoltatorilor să atribuie stiluri personalizate prin intermediul CSS, sau funcționalități cu ajutorul JavaScript.

Prin utilizarea tuturor uneltelor puse la dispoziție de către HTML, dezvoltatorii web pot crea pagini web moderne și care funcționează pe o multitudine de motoare de căutare diferite. HTML reprezintă o fundație solidă pentru dezvoltarea de aplicații web, stând la baza construirii paginilor web.

## CSS

CSS (Cascading Style Sheets) este un limbaj de dezvoltare web folosit pentru a stiliza aspectul conținutului HTML. Acesta funcționează în strânsă legătură cu HTML, permițând dezvoltatorilor să controleze modul în care elementele HTML sunt afișate în pagină.

Funcția sa principală este de a aplica stiluri și de a schimba design-ul elementelor HTML. CSS utilizează reguli și selecții pentru a aplica stiluri la o întreagă colecție de elemente similare în loc de a defini aspectul fiecărui element individual.

CSS folosește un model cascading, ceea ce înseamnă că se pot defini regulile în mai multe locuri. Acestea pot mai apoi să fie aplicate în funcție de specificitatea selectorului și de ordinea în care sunt definite. Astfel, utilizatorii pot controla aspectul elementelor într-un mod detaliat și să creeze diferite ierarhii de stiluri.

Un stil CSS este definit într-un bloc de reguli, care conține un selector și un set de declarații. Selectorul specifica elementul, sau elementele asupra cărora se aplică stilul. Declarațiile definesc diferite proprietăți, precum culoarea, dimensiunea, fontul, marginile și multe altele.

Există o gamă largă de valori și proprietăți ce pot fi editate de către dezvoltatori pentru a adapta aspectul elementelor HTML la nevoile lor. De asemenea, CSS are și suport concepte avansate, cum ar fi selectori de clasă, identificatori, pseudo-clase și pseudo-elemente. Toate aceste lucruri permit stilizarea selectivă a anumitor stări sau elemente.

Un alt aspect foarte des folosit și important al CSS este faptul că permite dezvoltatorilor să aplice stiluri responsive, care sunt adaptate la diferite dimensiuni de ecrane și dispozitive (desktop, laptop, mobile etc.). Acest lucru se realizează prin utilizarea de media queries pentru definirea stilurilor diferite în funcție de mărimea ecranului și prin intermediul tehnicilor de layout flexibil sau grid.

CSS poate fi inclus într-un document HTML prin intermediul unui bloc <**style>** în antetul paginii, sau prin includerea unui fișier extern CSS prin utilizarea unui element <**link>**. Această separare a conținutului de către stiluri face gestionarea și întreținerea paginilor web mult mai facilă.HTMHT

Asemeni lui HTML, CSS este un limbaj esențial pentru dezvoltarea de aplicații web deoarece cu ajutorul său este creat aspectul unei pagini. De la stilizări simple, până la design-uri complexe, interactive și responsive, CSS oferă o gamă largă de opțiuni pentru a transforma conținutul unei pagini web într-un aspect vizual plăcut și atractiv utilizatorilor.

## TypeScript

TypeScript este un limbaj de programare open-source și gratuit care a fost dezvoltat de către Microsoft care extinde și îmbunătățește JavaScript adăugând diferitre funcționalități. Acesta a fost lansat la 1 octombrie 2012 și a devenit foarte repede unul dintre cele mai populare limbaje pentru dezvoltarea de aplicații web datorită avantajelor sale față de JavaScript.

Principalul său obiectiv este de a adăuga tipizare statică la JavaScript, ceea ce înseamnă că dezvoltatorii pot specifica tipurile de date pentru variabile, funcții și alte elemente din codul lor. Acest lucru aduce mai multe avantaje, cum ar fi:

* **Detectarea erorilor la compilare.** TypeScript verifică semantica codului înainte de a rula aplicația. Astfel, multe erori des întâlnite sunt detectate și semnalate în timpul procesului de dezvoltare, evitându-se apariția lor în timpul rulării programului.
* **Instrumente de dezvoltare și funcție de autcomplete îmbunătățite.** Editorii și mediile de dezvoltare care susțin TypeScript oferă funcționalități avansate de autocompletare și sugestii de cod. Acest lucru crește nivelul de productivitate al programatorilor și reduce erorile de cod ce pot apărea pe parcursul dezvoltării.
* **Înțelegere mai bună a codului.** Tipurile de date și semnăturile funcțiilor ajută la o mai bună înțelegere a codului. Acest lucru duce la o eficiență mai mare a programatorilor deoarece mai puțin timp este pierdut descifrând codul.
* **Refactorizare și reutilizare ușoară.** TypeScript facilitează refactoring-ul și reutilizarea codului prin furnizarea de informații precise despre structura și tipurile de date ale acestuia.

La momentul compilării, codul de TypeScript este transformat în cod de JavaScript înainte de a fi rulat într-un mediu de dezvoltare sau într-un browser. Acest lucru înseamnă că dezvoltatorii pot utiliza sintaxa și funcțiile avansate ale lui TypeScript, iar rezultatul final va fi cod perfect funcțional de JavaScript. De asemenea, cum TypeScript este un super-set al lui JavaScript, acest lucru înseamnă că orice cod valid din punct de vedere sintactic de JavaScript va fi și cod valid de TypeScript.

Pe lângă tipizare, TypeScript mai adaugă în plus câteva funcționalități avansate, cum ar fi:

* **Clase și obiecte:** TypeScript permite dezvoltatorilor să folosească concepte de programare orientată pe obiecte, cum ar fi definerea de clase și crearea de obiecte. Aceste lucruri duc la o organizare mai bună a codului și de asemenea permite utilizarea de concepte precum încapsularea și moștenirea.
* **Module:** TypeScript oferă suport nativ pentru modulare, permițând împărțirea codului în module pentru a fi importat sau exportat între diferite fișiere sau proiecte. Aceasta ajută la orgaizarea și reutilizarea codului în aplicații mai mari.
* **Generice:** TypeScript oferă suport pentru generice, care permit dezvoltatorilor să scrie cod generic pentru a-l adapta la diferite tipuri de date. Acest lucru sporește flexibilitatea și reutilizabilitatea codului.
* **Interfețe:** TypeScript permite implementarea de interfețe. Acestea definesc sintaxa pe care ar trebui s-o respecte o entitate. Interfețele conțin declarațiile pentru proprietăți, metode și event-uri, care mai apoi trebuie definite de către clasa derivată.