

BABEŞ–BOLYAI TUDOMÁNYEGYETEM KOLOZSVÁR
MATEMATIKA ÉS INFORMATIKA KAR
VÁLLALATI SZOFTVERTERVEZÉS ÉS FEJLESZTÉS SZAK

Magiszteri dolgozat

**TingBuDong: egy valós idejű,
közösségalapú kínai nyelvtanuló alkalmazás**



TÉMAVEZETŐ:

**DR. ING. VARGA LEVENTE
EGYETEMI ADJUNKTUS**

SZERZŐ:

NAGY BORÓKA

2025

BABEŞ-BOLYAI UNIVERSITY OF CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: ENTERPRISE SOFTWARE DESIGN AND
DEVELOPMENT

Master's Thesis

**TingBuDong: A Real-Time,
Community-Based Chinese Language
Learning Application**



ADVISOR:

LECTURER DR. ING. LEVENTE VARGA

AUTHOR:

BORÓKA NAGY

2025

UNIVERSITATEA BABEŞ–BOLYAI, CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA PROIECTAREA ȘI DEZVOLTAREA APLI-
CAȚIILOR ENTERPRISE

Lucrare de disertație

**TingBuDong: O aplicație de învățare a limbii
chineze în timp real, bazată pe comunitate**



CONDUCĂTOR ȘTIINȚIFIC:
LECT. UNIV. DR. ING. VARGA LEVENTE

ABSOLVENT:
NAGY BORÓKA

2025

BABEŞ–BOLYAI UNIVERSITY OF CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: ENTERPRISE SOFTWARE DESIGN AND
DEVELOPMENT

Master's Thesis

TingBuDong: A Real-Time, Community-Based Chinese Language Learning Application

Abstract

Over the past years, many people around the world have shown an interest in learning Chinese as a foreign language. There are many learning opportunities available, both online and offline. For complete beginners, it is hard to decide which road to take: language courses are costly and not available in all regions, language apps and online materials don't always follow the requirements of the standard language exam, and studying online might feel isolating.

The TingBuDong application offers a combined solution: it is a platform where beginners can start their journeys as part of a community. They can engage with other students in built-in chatrooms, practice dialogues through WebRTC voice calls and play competitive games together. Besides these, they can practice with an AI-powered chatbot, learn the correct pronunciation of words with the incorporated speech-to-text functionality, and access learning materials that align with the syllabus of the standard language exam.

By integrating social engagement and exam-aligned study materials, TingBuDong provides a supportive space for effective Chinese learning.

This work is the result of my own activity. I have never provided or received unauthorised assistance in this work.

2025

BORÓKA NAGY

ADVISOR:
LECTURER DR. ING. LEVENTE VARGA

Tartalomjegyzék

1. Bevezető	1
2. A TingBuDong alkalmazás bemutatása	4
2.1. A projekt célja	4
2.2. Főbb komponensek, funkcióik és funkcionalitásaiak	4
2.3. UI/UX tervezés	7
3. Technológiák és fejlesztési környezet	8
3.1. Szerver oldal: Node.js és Express	8
3.2. Kliens oldal: React	9
3.3. Verziókezelés	9
3.3.1. Git	9
3.3.2. GitLab	9
3.4. Fejlesztői környezet	10
3.4.1. IDE: Visual Studio Code	10
3.4.2. Csomagkezelés: Node Package Manager	10
3.5. Adatbázis	11
3.5.1. MongoDB	11
3.5.2. Cache: Redis	12
3.6. Valós idejű kommunikáció	12
3.6.1. Socket.IO	12
3.6.2. RabbitMQ	12
3.7. Peer-to-peer kommunikáció: WebRTC	13
3.8. Konténerizálás és deployment	13
3.8.1. Docker és docker-compose	13
3.8.2. Kubernetes	14
3.8.3. Helm	14
3.8.4. Minikube	14
4. A rendszer megvalósítása	15
4.1. Frontend és backend kapcsolata	16
4.2. Adatbázis-kezelés és adatmodell	19
4.3. Felhasználókezelés, hitelesítés, biztonság	20
4.4. Chat funkciók: chatszobák és AI API implementáció	21

TARTALOMJEGYZÉK

4.5. WebRTC integráció	23
4.6. Nyelvspecifikus funkciók implementálása (pinyin, hangfelismerés)	25
4.7. Konténerizálás	26
5. A TingBuDong működése	28
6. Következtetések és továbbfejlesztési lehetőségek	47

1. fejezet

Bevezető

Az utóbbi években a kínai nyelv oktatása és tanulása világszerte nagy mértékben elterjedt. 2021-es adatok alapján több mint 180 országban vannak elérhető nyelvkurzusok. Ezekre lehetséget nyújt az 1500 darab globálisan elhelyezkedő Confucius intézet. Az intézetekkel együttműködve megközelítőleg 75000 kis- és középiskolának, továbbá 4000 külföldi egyetemnek van már kínai nyelvre fókusztáló szakiránya vagy kurzusa. 2021-ben világszerte nagyjából 25 millió külföldi ember tanult aktívan kínaiul, viszont, ha az összes olyan külföldi létszámát szeretnénk meghatározni, akik az életük során tanultak valamikor kínaiul, akkor ezek száma 200 millió köré tehető [1].

Az emberek több célból kezdik el tanulni a kínai nyelvet, motivációjukból adódóan a tanulási stílusuk is jelentősen eltérő. Vannak az örökségük fenntartása céljából tanulók (heritage learners), akik Kínán kívül élő, de kínai gyökerekkel rendelkező személyek. Ők általában nagyobb tudással rendelkeznek a kínai hagyományokról és szokásokról, motivációjuk az identitássukkal és a kultúrájukkal való kapcsolódás. A másik nagyobb csoport a nem Kínából származó személyek (non-heritage learners). Ezekenél a személyeknél ösztönző faktor lehet a kínai nyelv folyékony ismeretének gyakorlati előnye, többek közt a globális versenyképesség növelése érdekében. Vannak, akiket a kínai gazdag kultúra érdekel, ezt szeretnék jobban megismerni a nyelven keresztül, vagy saját képességeiket akarják próbára tenni azáltal, hogy a világ egyik legnehezebb nyelvét próbálják elsajátítani [2].

A mai felgyorsult világban nem meglepő, hogy a nyelvtanulás is, mint sok más egyéb tevékenység, digitalizálódott. Számos platform, fórum, telefonos- és asztali alkalmazás létezik, amely az idegen nyelvet tanulni vágyókat célozza meg. Egy 2020-as amerikai kutatás eredményei alátámasztják azt, hogy az applikáció alapú nyelvtanulás nagyon hatékony a kezdő nyelvtanulóknál. Akár napi 10 percnyi tanulással látható fejlődést lehet elérni a használatukkal úgy

1. FEJEZET: BEVEZETŐ

szókincs, mint nyelvtan és szóbeli kommunikáció terén is [3]. Erre talán a legjobb példa a Duolingo.

A Duolingo jelenleg a legnépszerűbb és leggyakrabban választott nyelvtanuló applikáció, amellyel összesen 40 különböző nyelvet lehet tanulni. Ezek között 2024-ben a kínai nyelv a 8. legnépszerűbb volt [4]. A Duolingo legjellemzőbb részei közé tartozik az alkalmazás zöld bagoly kabala, a napi leckékkel meghosszabbítható tűz (streak) és a heti versenyre ösztönző leaderboard. Ezek a funkciók segítettek az applikáció brand építésében, továbbá elősegítik a felhasználók motivációtövelését is. Viszont egy Mogavi és mások által 2022-ben végzett kvalitatív kutatás rávilágított a túlzottan nagymértékű gamifikáció potenciális hátrányaira is. A folyamatos heti verseny a többi felhasználóval és a tűz fenntartása is gyakran obszessziót vált ki a tanulókból, ezáltal annyira a játéakra és a versenyre koncentrálnak, hogy teljesen elvonódik a figyelmük a kezdeti célról, ami a nyelvtanulás [5]. Ezek mellett a Duolingo, habár lehetőséget nyújt a többi felhasználóval való kapcsolódásra, mégis inkább a versenyvágy növelése köré épül. Láthatjuk a barátaink elért eredményét, gyűjthetünk közösen napi tüzet, gratulálhatunk és kommentelhetünk az új mérföldköveikhez. Viszont nem léphetünk direkt kapcsolatba velük, nincs lehetőségünk személyes üzeneteket váltani, vagy közösen megoldani feladatokat, ezért az applikáció nem nevezhető közösségalapúnak. Mind ezek mellett talán a legfontosabb megemlítendő dolog a Duolingoval kapcsolatban a feladatokban lévő fura mondatok. Valószínűleg a legtöbb felhasználó találkozott már ilyen példával.

Nem vonatkoztathattunk el attól, hogy a nyelvtanulás mögötti egyik legnagyobb motiváló faktor a nyelv használatával elért gyakorlati haszon. Viszont a *Kenyeret eszek és a padlón sírok* ("I am eating bread and crying on the floor") [4] és hasonló mondatok, nem járulnak hozzá a gyakorlatban gyakran használt mondatok gyarapításához. Ez könnyebben tanulható nyelvek esetében nem biztos, hogy akkora problémát okoz, viszont a kínai nyelvnél demotiváló lehet sok befektetett energiával olyan szavakat és kifejezéseket tanulni, amelyek egyáltalán nincsenek összhangban a való életben használt, vagy a nyelvvizsgához szükséges tananyaggal.

A kínai nyelvvizsgák közül a HSK (Hànyǔ Shuǐpíng Kǎoshì) nevezhető a globális standardnak. Itt 2021-es módosítások óta 9 különböző szintű vizsga van, az 1-3. szintek (elementary) felelnek meg a kezdő vagy az általános szinteknek. A szintek elvárásainál konkrétan meghatározott nyelvtani struktúrák és szavak is meg vannak adva, amelyek logikusan egymás után kerültek összeállításra, azért, hogy minél hatékonyabban lehessen elsajátítani a nyelvet. Például az HSK 3-as szintnél 900 konkrét szót és 210 nyelvtani pontot kell ismerni a vizsgához [6].

1. FEJEZET: BEVEZETŐ

Mivel ennyire specifikus elvárások vannak a vizsgáknál, fontos, hogy az online platformokon elérhető tananyagok összhangban legyenek ezekkel.

A dolgozat alapjául szolgáló alkalmazás elkészítése mögötti motiváció a kínai nyelvtanulás alatt szerzett személyes pozitív (és időnkénti negatív) tapasztalat volt. Az applikáció fejlesztése során a cél egy olyan megoldás létrehozása volt, ahol lehetőség van másokkal együtt haladni, a saját tempónkban készülni a nyelvvizsgákra, saját időbeosztásunk szerint, ingyen.

A TingBuDong az eddig említett dolgokat ötvözi. Egy valós idejű webalkalmazás, ami egy közösségalapú nyelvtanuló platformot valósít meg. A felhasználók az HSK 1-3. szintfelmérő vizsgák elvárásainak megfelelő anyagokból tanulhatnak, de a nyelvtudás fejlesztése mellett a hasonló céllal rendelkező személyekkel való kapcsolódásnak is teret ad. Lehet barátkozni, barátainkkal beszélgetni, közösen kínai párbeszédet gyakorolni, de akár kompetitív játékokat is játszani. Ezen funkciók a közösséghoz való tartozás érzését hivatottak megteremteni, anélkül, hogy a túlzott gamifikáció elvenné a hangsúlyt a tanulásról.

A dolgozat tartalmilag 6 külön részből áll össze. A második fejezetben a TingBuDong alkalmazás főbb részei és funkcionalitásai kerülnek bemutatásra, ezután a harmadik rész a projektben használt technológiák ismertetésére fókuszál. A negyedik fejezetben az olvasó a konkrét implementációs részletekbe tekinthet be, kezdve a kliens- és szerveroldal kommunikációjával, adatbázis-műveleteken át, egészen a különböző valós idejű funkciók megvalósításáig. Az ötödik részben az applikáció működése kerül bemutatásra, végigkövethető, hogy mit lát a kliens és milyen lehetőségei vannak az applikáció használata közben. A hatodik részben a következetésekéről és a potenciális továbbfejlesztési lehetőségekről lehet olvasni.

2. fejezet

A TingBuDong alkalmazás bemutatása

Összefoglaló: Ebben a fejezetben az alkalmazás célja, célcsoportja, fő funkciói és a tervezési folyamat kerül bemutatásra.

2.1. A projekt célja

A TingBuDong alkalmazás mögötti motiváció egy díjmentes, bárki számára elérhető platform létrehozása volt, lehetőséget kínálva a felhasználóknak arra, hogy a nyelvvizsga kritériumoknak megfelelő szavakat és nyelvtani szabályokat tanuljanak. Emellett az is fontos szerepet játszott, hogy kapcsolatba tudjanak lépni más felhasználókkal, egy közösség részeként megtapasztalva az online nyelvtanulást. Így a fő célcsoportnak azok a személyek mondhatók, akik szeretnének kínaiul tanulni, de nem tudják, hogy hol és hogyan kezdjék el az útjukat.

Az applikáció neve, TingBuDong, a kínai *tīng bù dōng* - 听不懂 mondatból született, aminek jelentése tükörfordításban "hallom, de nem értem", vagy egyszerűbben "nem értem". Az alkalmazás mottója így lett a *The journey from tīng bù dōng to tīng dōng* (Az út a nem értemtől az értemig).

2.2. Főbb komponensek, funkcióik és funkcionalitásai

Kezdőlap

Az alkalmazás megnyitása után a Home oldal egy rövid üdvözlő szöveggel fogadja a felhasználót, ahol el kell döntenie, hogy következő lépésként regisztrálni vagy pedig bejelentkezni szeretne, annak függvényében, hogy rendelkezik-e már felhasználói fiókkal. A kezdőoldalnak van egy másik verziója is, amit az autentikált felhasználók láthatnak, ennek célja egy átfogó

2. FEJEZET: A TINGBUDONG ALKALMAZÁS BEMUTATÁSA

képet alkotni az alkalmazásban elérhető funkcionálisokról: barátkozás, csevegés, játékokhoz és tananyagokhoz való hozzáférés.

Regisztráció és bejelentkezés

A regisztrációs oldal célja a leendő felhasználók adatainak begyűjtése: felhasználónév, jelszó, név, születési dátum, származási ország, kínai nyelvtudás szintje és bemutatkozó szöveg. A bejelentkezés a felhasználó név - jelszó párossal történik.

Profil

A Profil oldal tartalmazza a felhasználó személyes adatait, amiket regisztrációkor megadott magáról. Itt lehetősége van ezeket az információkat módosítani, továbbá megjelenik a teljesített leckék száma és az esetlegesen bejövő baráti jelölések. Célja tehát, hogy a felhasználóval és tevékenységével kapcsolatos fontosabb információkat szemléltesse.

Közösségi oldal

A Community oldal arra szolgál, hogy a TingBuDong felhasználókat jobban meg lehessen ismerni a profiljaik alapján, továbbá lehessen baráti jelöléseket küldeni nekik. Lehetőség van a diákok közti szűrési műveletek alkalmazására is: származási ország, kínai nyelvtudási szint és kínai horoszkóp szerint. A filterek segítenek abban, hogy minél könnyebben lehessen rátalálni hasonló személyekre, növelte ezzel a potenciális barátságok kialakulásának valószínűségét.

Oktatóanyagok

A **Materials** oldal egy átvezető oldal: innen lehet navigálni a leckék térképes oldalához (Roadmap) vagy a szószedet (Vocabulary) oldalra.

A **Roadmap** oldalon lineárisan elhelyezve találhatók a felhasználó nyelvismereti szintjének megfelelő leckék. Az oldal célja, hogy tudassa a felhasználóval, hogy jelenleg hol tart a tanulási útján, amelyet egy színkódos jelölés mutat meg.

2. FEJEZET: A TINGBUDONG ALKALMAZÁS BEMUTATÁSA

Az leckék egységes struktúra alapján épülnek fel: a fejezet fő pontjainak bemutatása és egy rövid párbeszéd. A párbeszéd és a benne lévő szavak helyes kiejtését lehetőség van meghallgatni, illetve a szavak esetében a felhasználók a saját kiejtésüket is tesztelni tudják. Ezek a funkciók egy interaktív tanulási élményt hivatottak teremteni: nem csak írott szöveg van, hanem a hallás utáni értést és a szavak kiejtését is lehet általuk fejleszteni.

A **Vocabulary** oldalra navigálva 9 különböző tematika szavai közül lehet választani. Egy téma oldalán megtalálható a kategóriához tartozó összes szó, szólapokra elhelyezve. A leckénél található funkcionálással megegyező módon, meg lehet hallgatni a szavak helyes kiejtését, valamint tesztelni a saját kiejtés helyességét. Emellett több válaszlehetőséges kvízzel is próbára teheti a felhasználó a memóriáját és megszerzett tudását.

Csevegés oldal

A TingBuDong 3 különböző chatelési formára ad lehetőséget: privát csevegés egy baráttal, csoportos beszélgetés több baráttal vagy privát tanulás egy mesterséges intelligencia (MI) tanárral. Az utóbbinál 2 különböző nyelvi modell közül lehet választani, Gemma és ChatGPT. A MI tanárral folytatott beszélgetés egy interaktív kínai órát próbál lemodellezni. Itt lehetősége van a diáknak kérdezni is, ha plusz magyarázatra van szüksége.

Játékok

A Games oldalon két különböző oktató célú játék található. Az egyik kompetitív, míg a másik inkább kollaboratív jellegű. Mindkettő meghívás alapon működik, vagyis akkor kezdődik a játék, ha a meghívást kapó fél elfogadja a felkérést.

A **Watch and Tell** játék egy hagyományos képfelismerő játék. Amelyik fél hamarabb kattint a képen ábrázolt doleg kínai megfelelőjére, kap egy plusz pontot. Ezt a játékot a felhasználó játszhatja egy általa választott (szintén aktív) barátjával, vagy pedig egy randomizált módon kiválasztott aktív felhasználóval. A játék szórakoztató környezetben fókuszál a kínai szavak tanulására, de emellett a közösségi tagjaival való kapcsolódást is lehetővé teszi.

A kollaboratív **Dialogue** játék, a nevéből is adódóan párbeszéd gyakorlásra készült. A diákok közt ki lesz osztva a dialógusban résztvevő két szereplő, ennek megfelelően kell helyesen kiejteniük a saját szereplőjük mondatait. A játék teljes ideje alatt hallják egymást a játékosok,

2. FEJEZET: A TINGBU DONG ALKALMAZÁS BEMUTATÁSA

ezért a biztonságos légkör fenntartása érdekében nem lehet ezt a játékot random felhasználóval játszani, csak barátokkal. Ha a felhasználó olyankor is szeretne gyakorolni, amikor egy barátja sincs online, akkor megteheti ezt az egyszemélyes verzióval, amikor az egyik szerepkört a számítógép tölti be, automatikusan kiejtve a második szereplő mondatait. Ez a játék hozzájárul ahhoz, hogy a passzív tanulás mellett, valós élethelyzetekre jellemző párbeszédekkel is találkozzon a felhasználó.

Fórum és Dashboard

Mindkét oldal célja, hogy a felhasználó betekintést nyerjen a vele egy nyelvi szinten lévő diákok munkájába és teljesítményébe. A Fórum oldal egy afféle tevékenység-megosztóként működik, ahol az elérte eredményekről lehet tudomást szerezni: teljesített új leckét, játszott és megnyert játékok, újonnan szerzett barátok.

A Dashboard oldalon rangsorolások találhatók a kvízekkel gyűjthető pontok alapján. minden tematikának saját táblázata van. Az oldal célja, hogy a diákok minél gyakrabban ismételjék a megtanult szavakat és karaktereket, minden helyes válasz után pontot szerezve. Nem azon van a lényeg, hogy a felhasználók állandó versenyben álljanak egymással, hanem a saját tempóban való fejlődésen, a motiváció növelésén és az egymás inspirálásán.

2.3. UI/UX tervezés

A felhasználói felület tervezésénél fontos faktor volt az intuitív módon való használhatóság, ezért leginkább az egyszerű, de kohézív kinézet megalkotása volt a cél. Az alkalmazás színpallettaját a narancssárga és szürke színek alkotják, kiegészülve a fehér és fekete alapszínekkel. A játékoknál vagy helyesség ellenőrzésénél bejött még a zöld és a halvány piros, amelyek univerzálisan a helyes és helytelen szimbólumai.

Az alkalmazásban található profilképekhez a Flaticon.com oldalon, Roundicons Premium felhasználó által megosztott ingyenesen letölthető képek lettek használva [7]. Ezek a projektbe való integráció előtt át lettek színezve, hogy találjanak az alkalmazás kiemelőszínéhez. A párbeszédekhez használt karakter ikonok szintén erről az oldalról származnak.

3. fejezet

Technológiák és fejlesztési környezet

Összefoglaló: Ebben a fejezetben az applikáció során használt különböző technológiák kerülnek bemutatásra.

3.1. Szerver oldal: Node.js és Express

Node.js

A Node.js egy nyílt forráskódú, cross-platform szerveroldali JavaScript futtatókörnyezet. Nem a böngésző futtatja, hanem szerveroldali alkalmazásként fut. A Google Chrome V8-as motorjára épül, ami miatt magas teljesítményt tud nyújtani. A Node.js applikáció egyetlen folyamatban fut, nem hoz létre külön szálat minden egyes kéréshez, ezért akár több ezer egyidejű kapcsolatot is hatékonyan tud kezelni a szálkezeléssel járó bonyodalmak és nehézségek nélkül. Egyik fő előnye, hogy a fejlesztők kliens- és szerveroldali kódot is tudnak írni ugyanazzal a JavaScript nyelvvel [8].

Express

Az Express vagy Express.js egy keretrendszer, ami kiterjeszti a Node.js alapszintű képességeit. Az alkalmazás végpontjainak meghatározására és az ezek közötti routingra (útválasztásra) nyújt megoldást. Segíti a RESTful API-k megvalósítását, mivel a kliens által küldött kérés URL-t és a HTTP metódust (GET, POST, UPDATE stb.) szembeállítja az alkalmazás által definiált route-okkal, majd úgy irányítja a bejövő kérést, hogy a megfelelő kódrészlet hajtódjon végre. A middleware-k lehetővé teszik a kérés (req) és válasz (res) objektumok módosítását, ezáltal lehet hibakezelést végezni vagy valamilyen plusz feldolgozási logikát meghatározni [9].

3.2. Kliens oldal: React

React

A React egy nyílt forráskódú JavaScript könyvtár, amelynek jelenleg a Meta felel a karban-tartásáért. A React-et a felhasználói felületek renderelésére (UI elemek kirajzolására) használ-ják. Jellegzetessége a komponens alapúság, vagyis a felhasználói felület apró egységeit (mint a szövegeket, képeket és gombokat) külön újrahasználható és egymásba ágyazható komponen-sekre lehet szétszedni. minden komponens a saját állapotát kezeli. Az alkalmazás fejlesztése közben, ha egy vagy több komponens forráskódjában változtatás történik, a React a Fast Refresh segítségével futási időben újrarendereli csak a módosított komponenseket, anélkül, hogy a többi komponenst befolyásolná, ezáltal felgyorsítva a fejlesztési folyamatot és az esetleges hibák kijavítását [10].

3.3. Verziókezelés

3.3.1. Git

A Git egy ingyenes, elosztott és nyílt forráskódú verziókövető rendszer, manapság projekt-mérettől függetlenül majdnem mindenhol használják. A Git úgy nevezett "pillanatfelvételket" (snapshot-okat) készít a projekt aktuális állapotáról, a későbbiekbén pedig, ha egy fájl tartalma nem változott, akkor nem készít új másolatot, hanem eltárol egy hivatkozást a már meglévő snapshotra. A fájlok három különböző állapotban lehetnek: módosított (modified - nem képezi a következő commit részét), előkészített (staged - következő commit részét képezi) és kommitolt (committed - véglegesen elmentett) [11].

3.3.2. GitLab

A GitLab egy nyílt forráskódú, Git repository-k kezelésére és tárolására kifejlesztett web-alapú rendszer. Korlátlan mennyiségű privát és publikus repository-t lehet létrehozni költség-mentesen, ezáltal lehet rajta tárolni a projektverziókat a fejlesztés teljes időszaka alatt. A GitLab számos segédesközöt nyújt, többek között CI/CD-re, csomagkezelésre vagy a fejlesztői csapat-on belüli kollaboráció elősegítésére [12]. Egy hosszabb ideig tartó projekt esetén, mint amilyen

3. FEJEZET: TECHNOLÓGIÁK ÉS FEJLESZTÉSI KÖRNYEZET

a TingBuDong készítése volt, elengedhetetlen részét képezte a verziókövetés a fejlesztési folyamatnak.

3.4. Fejlesztői környezet

3.4.1. IDE: Visual Studio Code

A Visual Studio Code (VSCode) egy ingyenes és nyílt forráskódú kód szerkesztő, melyet a Microsoft fejleszt és tart karban. Egy személyre szabható platformot biztosít, amelyet a felhasználó igényei szerint bővíthet különböző bővítmények (extension-ök) telepítésével. Emellett beépített terminállal rendelkezik, ezért szinte bármilyen programozási nyelv esetén megfelelő környezetet tud nyújtani a fejlesztéshez. Platformfüggetlen, tehát Linux, macOS és Windows rendszereken is elérhető [13]. Egy 2024-ben a Stack Overflow által készített kvantitatív felmérés eredménye kimutatta, hogy a több, mint 44 ezer hivatásos programozó 74%-a használja a Visual Studio Code-ot, ezáltal ez lett a legnépszerűbb IDE [14].

3.4.2. Csomagkezelés: Node Package Manager

A Node Package Manager (npm) a Node.js alapértelmezett csomag- és függőségkezelője. Az egész hátterében egy óriási központi adatbázis áll (registry), amely a JavaScript szoftvert és az ehhez tartozó meta információkat tárolja. Több mint 2 millió csomag van eltárolva benne, ezek egy része bárki által hozzáférhető, publikus csomag, de vannak cégek, amelyek itt hosztolják a privát csomagjaikat. Az adatbázissal az npm CLI (parancssoros felhasználói felület) használatával lehet interaktálni, függőségeket installálni vagy eltávolítani, vagy akár saját csomagokat is publikálni. A projektben a függőségeket a használt verziószámukkal együtt a package.json fájl tartalmazza [15].

3.5. Adatbázis

3.5.1. MongoDB

MongoDB

A MongoDB 2024-es adatok szerint a legnépszerűbb NoSQL (nem relációs, "not only SQL") típusú adatbázis, az összes adatbázis-kezelő rendszer között (relációs és nem relációs) pedig az ötödik helyen áll [16]. Fő jellemzői közé tartozik, hogy hagyományos táblázatsorok helyett JSON-szerű BSON formátumban tárolja az adatokat, kulcs-érték párokba rendezve azokat. Az egyik nagy előnye, hogy félstruktúrált, így sokkal nagyobb rugalmasságot biztosít a fejlesztőknek, hogy az adatok a lehető legkönnyebben kezelhető formában legyenek tárolva. A sharding-gal megvalósított horizontális skálázhatóságnak és a beépített replikációs mechanizmusnak köszönhetően nagy forgalommal rendelkező alkalmazások esetén is megbízható a teljesítménye és hozzáférhetősége [17].

MongoDB Atlas

A MongoDB Atlas lehetővé teszi, hogy a felhasználó úgy hozzon létre MongoDB adatbázist, hogy ennek telepítésével, skálázásával és egyéb működési részleteivel nem kell foglalkoznia. A weboldal intuitív használhatóságának köszönhetően a személyre szabások és módosítások könnyedén végrehajthatók. A lehető legjobb teljesítmény elérése érdekében egyszerre több felhőszolgáltató infrastruktúrájára támaszkodik, mint például az AWS, Azure és Google Cloud. Ezáltal mindenkorán ott lesz futtatva az alkalmazás, ahol a legkisebb késési időt (latency) tudja nyújtani, így eredményül globális skálázás esetén is magas rendelkezésre állást tud biztosítani [18].

Mongoose

A Mongoose egy könyvtár, amelyet a MongoDB adatobjektumok modellezésére hoztak létre a Node.js-hez. Lehetőséget ad, hogy szerveroldalon végezzünk adatvalidációt vagy bizonyos sémának megfelelő formára hozzuk az adatobjektumokat, mielőtt az adatbázisba kerülnének. Az npm csomagkezelő segítségével installálható [19].

3. FEJEZET: TECHNOLÓGIÁK ÉS FEJLESZTÉSI KÖRNYEZET

3.5.2. Cache: Redis

A Redis (Remote Dictionary Server) egy eszköz, amellyel adatokat lehet tárolni a memóriában. Ezt a funkciót többféleképpen lehet használni egy alkalmazásnál: adatbázisként, message broker-ként (üzenetközvetítő) vagy cache-ként (gyorsítótár). Az adatstruktúrák a RAM-ban vannak tárolva, így az alkalmazás válaszidejének gyorsítása érdekében könnyen és gyorsan lehet őket kiolvasni vagy beilleszteni [20]. A TingBuDong esetében a Redis olyan adatbázis-lekérdezések eredményeinek cache-elérése szolgál, amelyek várhatóan nagyobb terjedelműek, ilyenek például a chat-előzmények vagy a kínai szavak listája.

3.6. Valós idejű kommunikáció

3.6.1. Socket.IO

A Socket.IO egy könyvtár, amely lehetővé teszi a kétféle kommunikációt szerver és kliens között, így a szerver nem csak a kliens kéréseire tud választ küldeni, hanem tud kommunikációt is kezdeményezni. A Socket.IO egységes szintaxisú végpontokat biztosít az üzenetek küldésére (`socket.emit()`) és fogadására (`socket.on()`) kliens- és szerveroldalon is. Az üzenetküldés eseményalapú: egyedi eseménynévvel kerül elküldésre és ugyanezzel a névvel lehet fogadni. A szerver- és kliensoldalnak a sikeresen szinkronizált kommunikáció érdekében ezt az eseménynevet kell ismernie, illetve az üzenet formátumát. Üzenetek sugárzására (broadcast) is van lehetőség, különböző szobák létrehozásával és az ezekre való feliratkozással. Így egyszerre lehet egy bizonyos szobára feliratkozott összes felhasználóhoz eljuttatni ugyanazt az üzenetet [21].

3.6.2. RabbitMQ

A RabbitMQ egy ingyenes és nyílt forráskódú message broker, amely az AMQP (Advanced Message Queuing Protocol) implementációján alapszik. Az alkalmazás különböző komponensei közti üzeneteket küldési sorrendben tárolja és továbbítja a fogyasztóknak. Egy plusz megbízhatóságot nyújt az applikációnak, mivel az üzenetsor ideiglenesen tárolva van, bármilyen hiba esetén garantált, hogy ezek az üzenetek és sorrendük nem veszik el [22]. Az alkalmazásban a Dashboard oldalon található Leaderboard frissüléseinek számítására van használva.

3.7. Peer-to-peer kommunikáció: WebRTC

WebRTC

A WebRTC (Web Real-Time Communication) egy nyílt forráskódú, Google által is támogatott projekt és szabvány. Lényege, hogy JavaScript API-kon keresztül, a legtöbb modern böngészőben valós idejű adat-, hang- és videóalapú kommunikációt tesz lehetővé. A kapcsolat peer-to-peer módon jön létre, az adatátvitel végpontok közötti DTLS-SRTP titkosítással védett [23].

3.8. Konténerizálás és deployment

3.8.1. Docker és docker-compose

Docker

A Docker egy eszköz, amely az alkalmazások konténerizálását valósítja meg, így szállíthatóvá téve azokat. A konténerek hasonlóan működnek, mint a virtuális gépek, saját fájlrendszerük és környezetük van, viszont a host gép kernelét használják, ezért nem teljesen izoláltak. Könnyedén lehet őket indítani és megállítani. A konténerben lehetőség van különböző parancsok végrehajtására, például meg lehet adni a kiindulási image-et (FROM), lehet fájlokat másolni (COPY) vagy környezeti változókat megadni (ENV). Ezeket a műveleteket egy Dockerfile-ban lehet lépésről lépésre felsorolni. A konténer csak futtatókörnyezetként szolgál, az alkalmazás kódja és függősei image-kben vannak tárolva, melyeket csak olvasni lehet. Az image-k előnye a hordozhatóság, el lehet őket tárolni többek közt a Docker Hub platformon, ezáltal bárholnan hozzá lehet félni, ami nagy előny a fejlesztő csapatokk esetében [24].

Docker Compose

A Docker Compose egy olyan eszköz, amivel az applikáció összes konténerét egy helyen lehet irányítani, így lehet kezelni és beállítani a konténerek környezeti változóit, portjait, kiindulási image-ét, megadni a konténerek között levő kapcsolatokat és függősségeket, mindezt egyetlen docker-compose.yaml fájlon keresztül. A Docker CLI-ben a docker compose paranccsal lehet utasításokat végrehajtani [25].

3. FEJEZET: TECHNOLÓGIÁK ÉS FEJLESZTÉSI KÖRNYEZET

3.8.2. Kubernetes

A Kubernetes egy konténerizált alkalmazások menedzselésére létrejött nyílt forráskódú platform. Produkciós környezetben elengedhetetlen tényező az applikáció állandó elérhetősége, függetlenül attól, hogy milyen mértékű az aktuális terhelés, ezért a Kubernetes a fel- és leskálázásért, valamint a hibák kiküszöböléséért felel. A bejövő kéréseket terheléselosztó (LoadBalancer) segítségével tudja irányítani. Ha egy konténer leáll, akkor automatikusan lecseréli vagy újraindítja, hogy ne legyen kiesett idő. Főbb Kubernetes erőforrás típusok: Pod (ez a legkisebb Kubernetes egység), Service, Deployment, ReplicaSet, StatefulSet, Ingress [26].

3.8.3. Helm

A Helm a Kubernetes csomagkezelője, a csomagokat chartoknak hívják. A chartok újrahasznosítható, előre konfigurált Kubernetes erőforrás sablonok (például servicek, deploymentek), amelyekben a konkrét értékek helyein paraméterek találhatók. Ezeknek a paraméterekeknek a values.yaml fájlban lehet értéket adni. A Helm chartok moduláris felépítése miatt ugyanazt a csomagot eltérő környezetekben is lehet használni, könnyen telepíthetők vagy frissíthetők helm install és update parancsokkal [27].

3.8.4. Minikube

A Minikube használatával a fejlesztés és tesztelés során egy egynode-os Kubernetes klasztert lehet lokálisan és díjmentesen futtatni. Linux, Windows és macOS operációs rendszereken is elérhető [28].

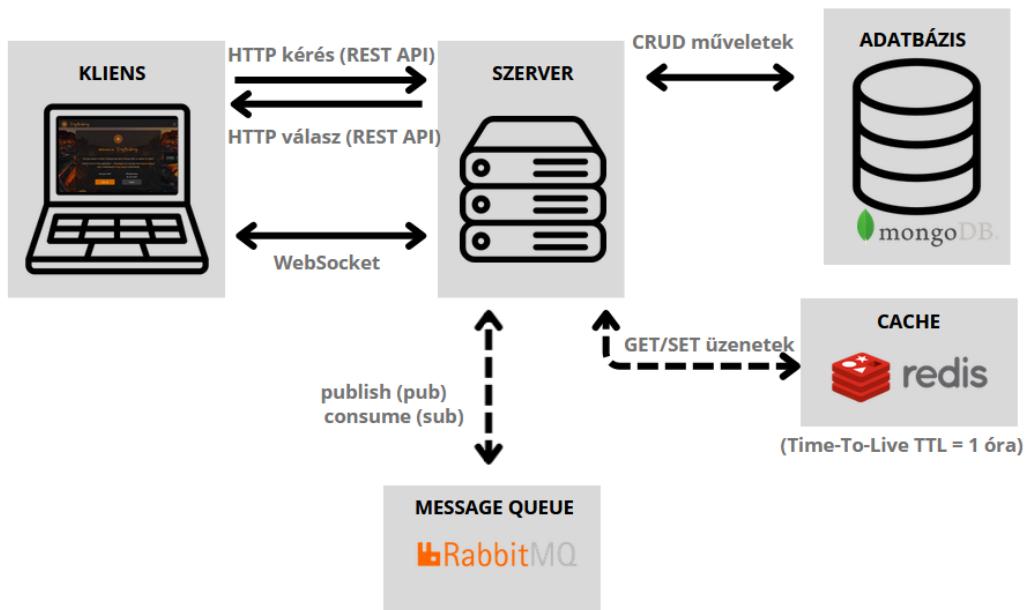
4. fejezet

A rendszer megvalósítása

Összefoglaló: A fejezetben az alkalmazás megvalósításának részletei kerülnek bemutatásra.

A TingBuDong alkalmazás kétrészes architektúrával rendelkezik: kliens és szerver. Monorepo, vagyis ugyanabban a repository-ban található a két rész forráskódja, könnyebbé téve például a docker-compose általi indítást és a közös konfigurációkat.

Architektúrailag 5 fő komponens képezi az applikációt: React-alapú kliens, Node.js/Express szerver, MongoDB adatbázis, Redis gyorsítótár és RabbitMQ üzenetközvetítő. A 4.1. ábra ezen komponensek egymással való kommunikációját modellezi, itt látható, hogy melyek azok a részek, amelyek között van kapcsolat, valamint az üzenetáramlás irányítottságát is szemlélteti. A következőkben ezen kapcsolatok gyakorlati megvalósításának részletei kerülnek bemutatásra.



4.1. ábra. TingBuDong: Kommunikációs diagram.

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

4.1. Frontend és backend kapcsolata

A kliens- és szerveroldal megfelelő kommunikációja elengedhetetlen egy működő alkalmazásnál. A TingBuDong esetében ez a kommunikáció két különböző módon történik, attól függően, hogy tartós adatkezelésről vagy valós idejű interakcióról van szó. A különböző adatbázis-műveleteknél (CRUD - Create, Read, Update, Delete) REST API végpontok kerülnek meghívásra, a valós idejű funkcionálitások pedig WebSocket-ek (Socket.IO) segítségével lettek implementálva.

REST API (HTTP)

Konkrét példaként, amikor a felhasználó meg akarja hívni valamelyik barátját egy játéakra, szükség van a barátainak listájára (adatbázis READ művelet). Ehhez a kliens oldal getFriends függvénye (lásd a 4.1. kód részlet) az authFetch hívással elküldi az aktuális felhasználó felhasználónévét, illetve az autentikációhoz szükséges JWT tokent a szerver /api/users/:username/friends végpontjához (lásd a 4.2. kód részlet). A szerver első lépésben a kéréssel érkező JWT token érvényességét hitelesíti a /middleware/auth.js szerint, majd a User.findOne({username}).select('friends') adatbázis-lekérdezést hajtja végre. Ha minden lépés sikeresen ment, akkor az így megkapott barátlista JSON formátumban kerül visszaküldésre a klienshez.

```
1 //client/src/services/profileService.js
2 export const getFriends = async (username) => {
3     try {
4         const response = await authFetch(`${API_URL}/${username}/friends`);
5         if (!response.ok) {
6             throw new Error(`Error fetching friends: ${response.statusText}`);
7         }
8         return response.json();
9     } catch (error) {
10        console.error('Error fetching friends:', error);
11        throw error;
12    }
13};
```

4.1. kód részlet. Kliensoldali getFriends függvény: REST API-hívás a /api/users/:username/friends végpontra.

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

```
1 //server/routes/users.js
2 router.get('/:username/friends', auth, async (req, res) => {
3   const { username } = req.params;
4   try {
5     const user = await User.findOne({ username }).select('friends');
6     if (!user) {
7       return res.status(404).json({ msg: 'User not found' });
8     }
9     res.json(user.friends);
10  } catch (err) {
11    console.error('Error fetching user friends:', err.message);
12    res.status(500).json({ error: err.message });
13  }
14});
```

4.2. kódrészlet. Szerveroldali REST API-végpont: barátlista lekérdezés (routes/users.js).

WebSocket (Socket.IO)

A valós idejű funkcionálitások implementálása WebSocket-ekkel történt. A WebSocket-kapcsolat lehetőséget ad, hogy egyetlen "kézfogás" után a kliens és szerver között egy folyamatos, kétirányú adatáramlás jöjjön létre, vagyis a szerver kezdeményezhet anélkül is kommunikációt, hogy a kliens erre explicit kérést küldene. Ezen a kétirányú csatornán lehet különböző eseményeket küldeni (emit) és fogadni (on), így minimalizálva a késleltetést. A TingBuDong projektben Socket.IO van használva többek között a valós idejű játékok működésénél, chatelési funkcióknál, vagy éppen a baráti jelölések küldésénél és fogadásánál. Kliensoldalon egyetlen egy Socket.IO kapcsolat lesz létrehozva és használva az említett összes funkcionálitásnál (lásd a 4.3. kódrészlet).

```
1 //client/src/socket.js
2 import { io } from 'socket.io-client';
3 import serverUrl from './constants/serverUrl.js';
4
5 const socket = io(serverUrl);
6
7 export default socket;
```

4.3. kódrészlet. Kliensoldali egyetlen Socket.IO kapcsolat.

Az előző példát folytatva, ha sikerült a barátok listáját lekérni, akkor a felhasználó eldönt-

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

heti, hogy melyik barátjával szeretne játszani, mivel a játékok meghívás alapján működnek. A meghívó elküldésekor a handleFriendClick függvény fut le (lásd a 4.4. kódrészlet), ahol először ellenőrizve van, hogy a választott barát jelenleg online elérhető-e. Ha elérhető, akkor a socket.emit('inviteToGame', ...) hívással egy azonnali, WebSocket-alapú üzenet megy a szerverre, ami tartalmazza a meghívó küldőjének (fromUser) és a célzott barát (toUser) felhasználóneveit, valamint a választott játéktípus (gameType). A hívásnál a fenti 4.3. kódrészletben megjelenített socket kapcsolat van használva.

```
1 //client/src/components/PlayWithFriend.js
2 const handleFriendClick = (friend) => {
3     if (!friend.isActive) {
4         alert(` ${friend.name} is currently not active. Try another friend.`);
5         return;
6     }
7
8     socket.emit('inviteToGame', {
9         fromUser: username,
10        toUser: friend.name,
11        gameType,
12    });
13};
```

4.4. kódrészlet. Kliensoldali handleFriendClick függvény: játékmeghívó küldése Socket.IO kapcsolaton keresztül.

Szerveroldalon a socket.on("inviteToGame") eseménykezelő (lásd a 4.5. kódrészlet) veszi át a meghívót a benne lévő információval együtt, majd ez alapján megkeresi a meghívó fogadójának socket azonosítóját. Ha a fogadó aktív, akkor erre az azonosítóra továbbküldi a meghívást, ugyanazzal az információval: io.to(toSocketId).emit("gameInvite", fromUser, toUser, gameType). Kliensoldalon a továbbküldött meghívást a 4.6. kódrészletben látható listener fogja elkapni és feldolgozni.

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

```
1 //server/sockets/socket.js
2 socket.on("inviteToGame", ({ fromUser, toUser, gameType }) => {
3     console.log(
4         `Game invite from ${fromUser} to ${toUser} with gameType: ${gameType}`
5     );
6     const toSocketId = userSocketMap[toUser];
7     if (toSocketId) {
8         io.to(toSocketId).emit("gameInvite", { fromUser, toUser, gameType });
9     } else {
10        const fromSocketId = userSocketMap[fromUser];
11        if (fromSocketId) {
12            io.to(fromSocketId).emit("friendInactive", toUser);
13        }
14    }
15});
```

4.5. kódrészlet. Szerveroldali inviteToGame eseménykezelő: játékmeghívás továbbítása.

```
1 //client/src/App.js
2 socket.on('gameInvite', ({ fromUser, toUser, gameType }) => {
3     if (toUser === username) {
4         setIncomingInvite({ fromUser, toUser, gameType });
5     }
6});
```

4.6. kódrészlet. Kliensoldali gameInvite eseménykezelő (játékmeghívás fogadása és feldolgozása).

4.2. Adatbázis-kezelés és adatmodell

A TingBuDong adatkezelésre és tárolásra egy MongoDB adatbázist használ, amelynek működési részleteiért a MongoDB Atlas felel. A csatlakozás a kapcsolati string segítségével jön létre, amelyben meg kell adni egy felhasználónév-jelszó párost, a klaszter nevét, amiben az adatbázis található, illetve egyéb egyedi beállításokat is meg lehet határozni. A végeredmény a következő struktúrához hasonlóan épül fel: `mongodb+srv://[username:password@]host1[/database][?options]`. A Mongoose az adatbázissal való kommunikációt és a sémaalapú adatkezelést biztosítja. minden entitás, amelynek saját kollekciója van az adatbázisban, egy-egy Mongoose-modellként van definiálva. A modellek előírják a dokumentumok szerkezetét, megadják a mezőket és ezek adattípusait,

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

illetve lehetővé teszik a default értékek vagy egyéb megkötések meghatározását is. A modellek a `server/models` mappában vannak elhelyezve, a következő nevekkel: `CompletedLessons.js` (lásd a 4.7. kódrészlet), `Dialogue.js`, `GroupChat.js`, `Lesson.js`, `Message.js`, `Milestone.js`, `Subscription.js`, `Topic.js`, `User.js` és `Word.js`. A sémákra épülő működés lehetővé teszi az automatikus adatvalidációt és a létrehozási/frissítési időbélyegek automatikus kezelését is, amely olvashatóságot és könnyített adatkezelést eredményez.

```
1 //server/models/CompletedLesson.js
2 const mongoose = require('mongoose');
3 const Schema = mongoose.Schema;
4
5 const CompletedLessonSchema = new Schema({
6   username: { type: String, required: true },
7   lessonKeyword: { type: String, required: true },
8   completedAt: { type: Date, default: Date.now }
9 });
10
11 module.exports = mongoose.model('CompletedLesson', CompletedLessonSchema);
```

4.7. kódrészlet. Mongoose-modell a teljesített leckék számítartására.

4.3. Felhasználókezelés, hitelesítés, biztonság

Minden alkalmazásban elengedhetetlen a kliensek adatainak megfelelő, biztonságos módon való tárolása. Az egyik legegyszerűbb hibaforrás az lehet, hogyha a felhasználói jelszavak nincsenek megfelelően titkosítva.

Jelszó tárolása

A TingBuDongban a jelszó hashelésére a bcrypt hash-függvény volt használva, amely minden jelszóhoz egy egyedi, véletlenszerűen generált "sót" ad (`bcrypt.genSalt(10)`). Ez a jelszóval együtt esik át a hash-elésen (`bcrypt.hash(password, salt)`), ezáltal két azonos jelszó is teljesen eltérő lesz a folyamat után. Az ily módon titkosított jelszó lesz eltárolva az adatbázisban, a bejelentkezéskor megadott jelszó helyességének ellenőrzése pedig a `bcrypt.compare(<beírt jelszó>, <titkosítva tárolt jelszó>)` függvénytel törénik.

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

Felhasználókezelés

A felhasználói session-kezelés JSON Web Tokenekkel (JWT) lett megvalósítva. Sikeres bejelentkezés esetén (POST /api/users/login végponton) a szerver egy 1 óra élettartamú tokent ad vissza, ez a token kliens oldalon a localStorage-ban lesz elmentve jwtToken néven. Ez fogja jelenteni, hogy a felhasználó be van jelentkezve. Ezek után minden kérés a böngésző beépített fetch függvénye helyett ennek egy kiterjesztett változatát fogja használni, a client/src/services/profileService.js fájlban definiált authFetch függvényt. Az authFetch hozzárendel egy header-t a kéréshez, ami tartalmazza a tokent.

Amikor a kérés megérkezik a szerveroldalra, a server/middleware/auth.js fájlban található auth middleware fogja ellenőrizni a header jelenlétét, majd a benne található token helyességét és lejárati idejét. Bármilyen hiba esetén HTTP 401 - Unauthorized státuszkódot küld vissza a kliensoldalnak. A token visszafejtése a jwt.verify() függvénnyel történik, ami paraméterként megkapja a tokent és a JWT runtime secret-et, ami egy környezeti változóként tárolt random string, ami minden JWT tokenhez hozzá van konkatenálva.

A kliens oldalra visszajövő HTTP 401 státusz esetében az authFetch kezeli a felhasználó kiléptetését: törli a tokent és a user objektumot a localStorage-ból, majd újratölteni az ablakot, ezáltal valid token hiányában a kliens átirányítódik a kezdőoldalra. Bármilyen más státuszkód esetén visszatéríti a szerver válaszát, hogy az a továbbiakban felhasználható legyen az oldal működésénél.

4.4. Chat funkciók: chatszobák és AI API implementáció

Privát- és csoportos chatszobák

Az applikáció 3 különböző típusú chatszobát engedélyez: privát szoba egy baráttal, csoporthoz kötött szoba több baráttal, illetve chatbot-tanár szoba. A privát szobák automatikusan létre vannak hozva, ezzel szemben a csoporthoz kötött szobákat explicit módon lehet elkészíteni a Create Group Chat feliratú gombra kattintva. Ekkor a háttérben a szerveroldal routes/groupChat.js fájljában található /create REST végpont lesz meghívva. A csoporthoz kötött szoba létrehozásának két kritériuma van: a szobának rendelkeznie kell egy névvel és a létrehozó személlyel együtt legalább 3 tagja kell legyen. A szobák beazonosításához és kezeléséhez a csoport MongoDB objektumának _id-ja (groupId névvel használva), privát szobáknál pedig a chatRoomId (struktúra:

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

felhasználóA-felhasználóB) van használva, mindenkorban ezek az értékek egyediek.

Chatszoba megnyitásakor le lesznek kérve a múltbeli üzenetek (ha nagyon sok van, akkor az utolsó 100 darab) a GET /api/messages/<azonosító> REST végponton, majd ezek el lesznek mentve a Redis cache-be messages:<chatRoomId> kulcs alatt, ezáltal a következő 1 óra alatt történő csevegés betöltésnél nem kell újból adatbázislekérést végezni. Üzenetküldésnél az automatikus frissülés Socket.IO-val történik, a sendMessage vagy sendGroupMessage események egyikével. Az üzenet előbb az adatbázisba lesz elmentve, majd frissítve lesz a cache, hogy az aktuális információt tartalmazza. Ilyenkor a cache lejáratú ideje mindig vissza van állítva az eredeti 1 órára. Ezen műveletek végeztével ki lesz küldve a receiveMessage esemény a fogadó-félnek vagy -feleknek.

AI API implementáció

A chatbot tanár az egyetlen csevegőszoba, amelynél nincsenek megjelenítve a múltbeli üzenetek, minden beszélgetés külön van kezelve. Megnyitáskor ezért egy reset kérés kerül ki-küldésre a szerveroldalnak, amely törli az előzményeket és visszaküld egy üdvözlőszöveget, ami tartalmaz egy random tematikát, amely köré a gyakorlat fog épülni. Két különböző nyelvi modell közül lehet választani. Az egyik az OpenAI gpt-3.5-turbo modellje, amely az OpenAI Cloud-ban fut. Az ide irányuló kérések a <https://api.openai.com/v1/chat/completions> végpontra lesznek elküldve egy egyedi API kulccsal együtt. Használata nem ingyenes, viszont magas minőségű válaszokat nyújt. Az OpenAI modelljei között való választáskor fontos szempont volt az ár-érték arány.

A másik elérhető nyelvi modell az alkalmazásban a Gemma 2B. Ez az OpenAI modellel ellentétben lokálisan van hasztolva az Ollama futtató környezettel, az ide irányuló hívások az OLLAMA_URL/v1/chat/completions végpontra mennek. Mivel a modell lokálisan futatható, a használata költségmentes, azonban a hardverkapacitás korlátozottá teszi a modellválasztást, így jelenleg a válaszok minősége és a válaszidő is kevésbé optimális. A jövőben viszont ezen lehet változtatni, amint a futtatóeszköz a Gemma nagyobb modelljeit is elbírja, akkor csupán árnyalatnyi különbség lesz a két modell válaszai között, jelenleg viszont az OpenAI jobb eredményeket produkál.

Minden új beszélgetés kezdetén a chatbot egy utasítássort (promptot) kap meg, ami leírja, hogy hogyan kell viselkednie a chatbotnak, válaszainak hangnemét és a tanítási logikát ("You

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

are a patient, structured Chinese tutor for HSK 1 beginners..."). A prompt alapján a gyakorlat menete a következő: témaválasztás, ami köré épül az óra, egy új szó bemutatása (karakter, pinyin, jelentés), 1-2 egyszerű gyakorlat (például: mondatkiegészítés, fordítás) küldése, majd az erre kapott válasz kiértékelése. Ha ezek a lépések megtörténtek, akkor ismétli a folyamatot az adott tematikához tartozó másik szóval.

4.5. WebRTC integráció

A párbeszéd- és kiejtés gyakorlására készült Dialogue Game játéknál a két fél között valós idejű audiohívás jön létre a játék ideje alatt. Ez a böngészők beépített WebRTC API-ja segítségével van megvalósítva, melyet Socket.IO-val történő signaling egészít ki. A hangalapú kommunikáció beépítésének célja az interaktivitás és a közösséggel való kapcsolódás növelése volt.

A DialogueGame komponens betöltésekor az első lépés az engedélykérés a felhasználótól a mikrofonjához való hozzáféréshez (lásd a 4.8. kódrészlet). Ha ez elfogadásra kerül, akkor a böngésző visszaad egy MediaStream objektumot a kliensoldalnak, ami a globalLocalStream globális változóban lesz tárolva.

```
1 //client/src/components/DialogueGameRoom.js
2 navigator.mediaDevices.getUserMedia({ audio: true })
3   .then(stream => {
4     globalLocalStream = stream;
5     createPeerConnection(otherUser);
6   });
```

4.8. kódrészlet. Mikrofon engedélykérés és válasz elmentése.

A createPeerConnection függvény egy RTCPeerConnection példányt hoz létre, benne a konfigurált STUN-szerverrel (lásd a 4.9. kódrészlet). A kapcsolat a globalPeerConnection nevű globális változóban lesz mentve, majd hozzá lesz adva a fenti (4.8) kódrészletben létrehozott globalLocalStream változóban eltárolt összes audio-sáv (track). A hívás során ehhez a példányhoz fog kötődni az összes offer, answer és ICE (Interactive Connectivity Establishment) candidate esemény.

Az RTCPeerConnection példányosításánál ICE szervernek a Google STUN (Session Tra-

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

```
1 //client/src/components/DialogueGameRoom.js
2 globalPeerConnection = new RTCPeerConnection({ iceServers: [{ urls:
3   ↪ 'stun:stun.l.google.com:19302' }] });
4 globalLocalStream.getTracks().forEach(track =>
  ↪ globalPeerConnection.addTrack(track, globalLocalStream));
```

4.9. kódrészlet. WebRTC kapcsolat létrehozása.

versal Utilities for NAT) szervere van megadva. A STUN szerver egyszerűen lekérdezi és visszaküldi a kliens publikus IP-címét és portját, lehetővé téve a peer-to-peer közvetlen kapcsolat kialakítását, amin keresztül a médiatartalom lesz szállítva.

A WebRTC jelzések feldolgozását Socket.IO eseménykezelők végzik, a bejövő üzeneteket nem módosítják vagy értelmezik, hanem kikeresik a fogadószemély socket ID-ját a userSocketMap[toUser] leképezéssel, majd továbbítják az üzeneteket.

Az offer esetében a következőképpen néz ki a kód (lásd a 4.10. kódrészlet):

```
1 //server/sockets/socket.js
2 socket.on("webrtc-offer", ({ toUser, description }) => {
3   const targetId = userSocketMap[toUser];
4   if (targetId) {
5     io.to(targetId).emit("webrtc-offer", {
6       fromUser: socket.username,
7       description
8     });
9   }
10});
```

4.10. kódrészlet. Szerveroldali WebRTC signaling: offer feldolgozása és továbbítása.

A fenti példához hasonlóan működik az webrtc-answer és a webrtc-candidate eseménykezelője is.

Amint a hívás véget ér (a játék befejeződik vagy valamelyik fél bontja a kapcsolatot), a RTCPeerConnection.close() metódussal le lesz zárva a kapcsolat, illetve a mikrofon használati engedélye is vissza lesz állítva az eredeti állapotra.

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

4.6. Nyelvspecifikus funkciók implementálása (pinyin, hangfelismerés)

A Dialogue Game játék az audióhívás mellett a kiejtés helyességének tesztelésére készült. Amint a kiküldött játékfelkérést elfogadja a meghívott fél, a játék inicializálásakor a joinDialogueGame esemény elküldődik a szervernek a gameId-val együtt. A szerver válaszként visszaküld egy dialógus objektumot az adatbázisból, ami tartalmazza: a párbeszéd címét, a párbeszédben résztvevő két karaktert, a nyelvi nehézségi szintet és egy lines tömböt, amiben a párbeszéd mondatai vannak sorban tárolva. minden tömb elem 4 mezőből áll: a beszélő neve, a mondat kínai karakterekkel, a mondat romanizált formában és a mondat angolul.

A játékosok között ki lesz osztva a két karakter: az a személy, akinek a felhasználóneve a játékszóba azonosítójában első helyen szerepel, az lesz az a karakter, aki a párbeszédet kezdi (szoba azonosító kinézete: felhasználóA-felhasználóB, például: bob-jim). Az azonosítóban a második játékos lesz a második karakter. A kiejtésellenőrzést elindító gomb minden csak annak a játéknak jelenik meg, aki soron következik, ez az isMyTurn konstans segítségével van meghatározva, ami megnézi, hogy a játékos karaktere megegyezik-e az aktuális mondat karakterével. Ha megegyezik a két karakter, akkor megjelenik a mikrofon gomb a Start Recording felirattal.

A mondatot úgy kell formára hozni a segmentAndConvert függvény által, hogy létrejöjjön a célmondat, ami a helyes megoldást fogja jelenteni. A függvény működése: a kínai karakterekkel írt mondatot az írásjelek eltávolítása után először szavakra kell bontani, mivel a kínai nyelvben nem használnak szóközöket. Ez a szegmentálás a **js-jieba** package használatával van megvalósítva, a **jieba.cut(noPunct)** függénnyel, ahol a noPunct az írásjelek eltávolítása utáni mondat. Miután megtörtént a szavakra bontás, a **pinyin** package segítségével tónus nélküli romanizált tokenekké lesznek alakítva a szavak, majd a tokeneket összefűzi szavakká, tehát a végeredmény egy romanizált szavakból álló tömb lesz. A végső tömb a targetPinyin változóban lesz tárolva. Az imént leírt lépések egy konkrét példával szemléltetve:

"你好，老师" -> "你好老师" -> ["你好", "老师"] -> [[['ni'], ['hao']], [['lao'], ['shi']]] -> ["nihao", "laoshi"]

A Start Recording gombra kattintva kezdődik a kiejtés helyességének ellenőrzése. A beszédfelismerés a böngésző beépített Speech-to-Text (STT) motrálval történik. A **react-speech-**

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

recognition könyvtárral folyamatosan valós időben frissül egy transcript nevű változó, ami az átírt szöveget tartalmazza. Egy useEffect figyeli a transcript változót és minden alkalommal, amikor új karakterek érkeznek, a szöveg formázva lesz a segmentAndConvert függvényel, majd az így megkapott tömb a recognizedPinyin változóban lesz eltárolva. Ahhoz, hogy valós időben lehessen visszajelzést adni a felhasználónak, a recognizedPinyin tömb elemei páronként össze lesznek hasonlítva a targetPinyin-ben ugyanazon az indexen található szóval: ha a két szó egyezik, zöldre lesz színezve a felhasználói felületen a szó (correct CSS-osztály), ha pedig eltérnek, akkor pirosra (incorrect CSS-osztály). A useEffect segítségével minden új szó után le fog futni ez a kiértékelés.

Amikor a felhasználó leállítja a kiejtés ellenőrzést a Stop & Check gombbal, akkor a két teljes tömb is össze lesz vetve. Ha a mondatot teljes egészében helyesen sikerült kimondani, akkor a lineCompleted Socket.IO esemény küldődik el a szervernek, jelezve, hogy jöhet a következő sor: `socket.emit('lineCompleted', {gameId, newIndex: nextIndex});`. A szerver fogadja ezt az üzenetet, frissíti a párbeszéd állapotát, majd visszaküldi minden két játékosnak a következő párbeszéd sorra vonatkozó adatokat.

4.7. Konténerizálás

Az alkalmazás futtatásának megkönnyítése miatt az alkalmazás konténerizálva van, így lokálisan könnyen indítható Docker Compose segítségével. A docker-compose.yaml fájlban 5 különböző konténer van definiálva az applikáció fő részeinek: frontend, server, redis, ollama-gemma és rabbitmq. A Docker Compose általi indítás jelentősen megkönnyítette és felgyorsította a lokális fejlesztést és tesztelést.

A helyi fejlesztés mellett nem elhanyagolható a produkciós környezetre is gondolni, ami nél Docker helyett általában Kuberne test alkalmaznak. Kuberne tes használatával könnyen lehet skálázni az alkalmazás részeit futási időben a terheltség függvényében, ezáltal magas rendelkezésreállást lehet biztosítani. A Minikube segítségével lokálisan lehet futtatni egy Kuberne tes klasztert díjmentesen, lehetővé téve, hogy Kuberne tes környezetben is működőkéssé lehessen tenni az alkalmazást, anélkül, hogy valamilyen felhőalapú szolgáltatónak kelljen fizetni a hosztolásért.

A klaszteren belül Helm chartok segítségével lehet telepíteni az applikációt. A chartok a helm/templates mappában található sablonokkal vannak definiálva, az alkalmazás mind az 5

4. FEJEZET: A RENDSZER MEGVALÓSÍTÁSA

része rendelkezik service és deployment leírással. A konkrét behelyettesítési értékek (például Docker image repository-ja, MongoDB- és szerver URL) a values.yaml fájlból kerülnek beállításra, ezek a sablonokban paraméterekkel vannak jelölve konkrét értékek helyett. A Helm chartok segítségével egy felhőalapú Kubernetes környezetben könnyedén fel lehet építeni, futtatni, skálázni és karban tartani a teljes alkalmazást.

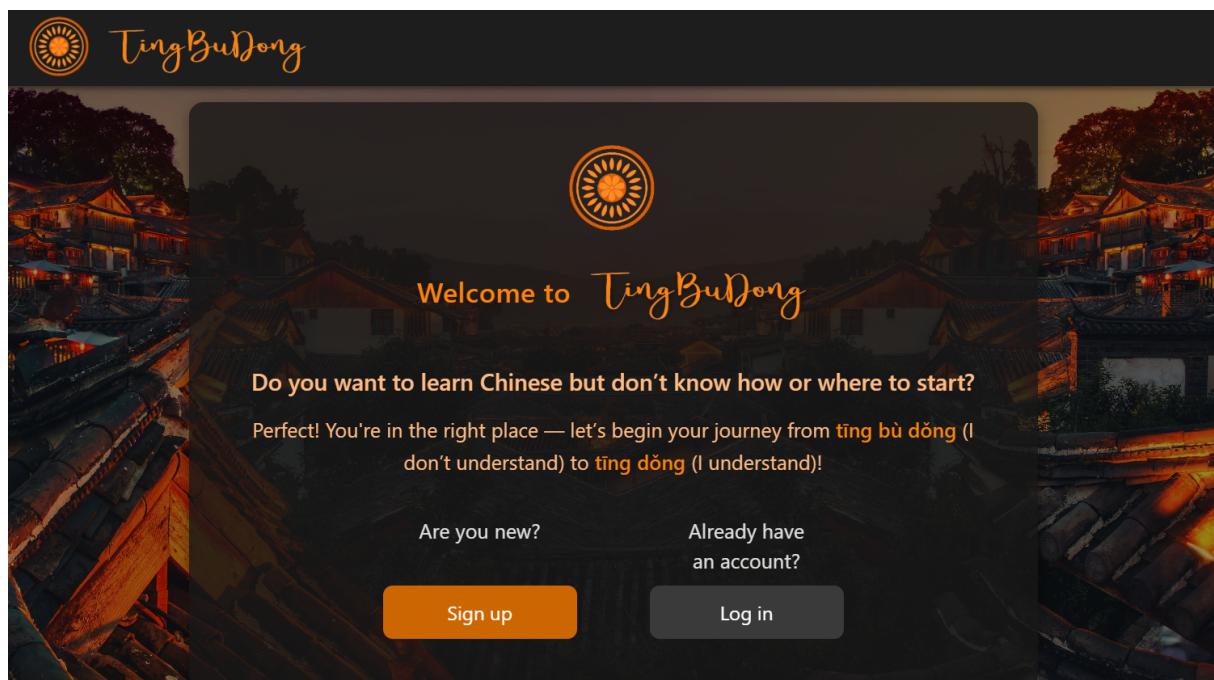
5. fejezet

A TingBuDong működése

Összefoglaló: A fejezet a TingBuDong alkalmazás működését követi végig, bemutatva a felhasználói felületet és az ezen elhelyezkedő elemek funkcionálitásait.

Kezdőoldal

Az TingBuDong applikáció megnyitásakor a felhasználót a Home kezdőoldal fogadja, amelyen egy rövid üdvözlőszöveg ismerteti az alkalmazás célját (lásd az 5.1. ábra). Két opció van felkínálva: gomb általi navigáció a regisztrációs oldalra és a bejelentkezés oldalra. A látogatónak választania kell a lehetőségek közül, annak függvényében, hogy rendelkezik-e már regisztrált profillal vagy sem.

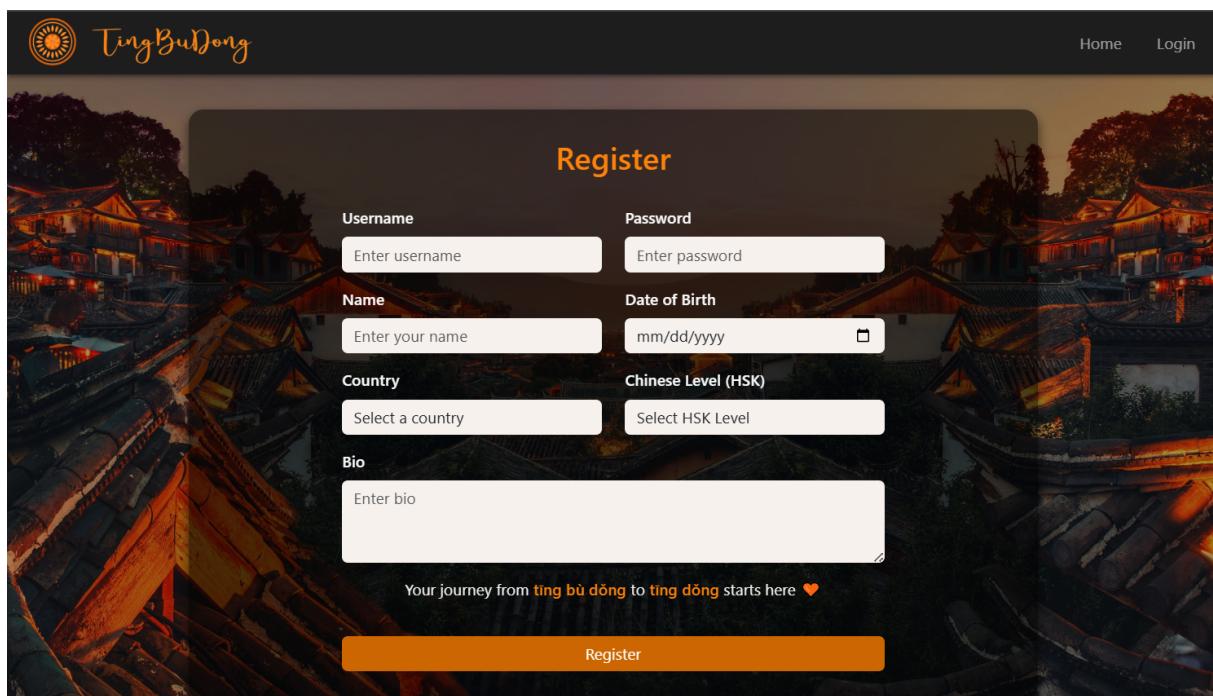


5.1. ábra. Külső kezdőoldal (Home).

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

Regisztráció

Új felhasználói fiók létrehozásához az 5.2. ábrán látható információkat kell megadni. A jelszó, felhasználónév, név és bemutatkozó szöveg rögzítése szövegmezők segítségével történik, az ország és a kínai tudás megadásánál előre meghatározott opciókból lehet választani egy dropdown listából, a születési dátumnál pedig egy beépített naptár widget áll rendelkezésre. Hiányos mezők esetén nem lehet tovább menni, validátor üzenet fog megjelenni az üresen hagyott mezőknél.

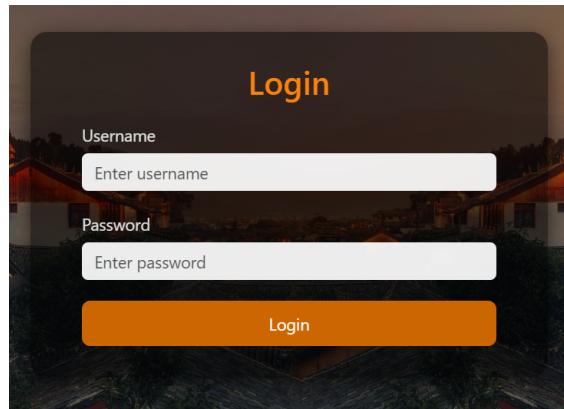


5.2. ábra. Regisztrációs oldal.

Bejelentkezés

A navigációs sávon vissza lehet váltani a kezdő- vagy bejelentkezési oldalakra. A bejelentkezés a felhasználónév és a jelszó alapján történik az 5.3. ábrán látható módon.

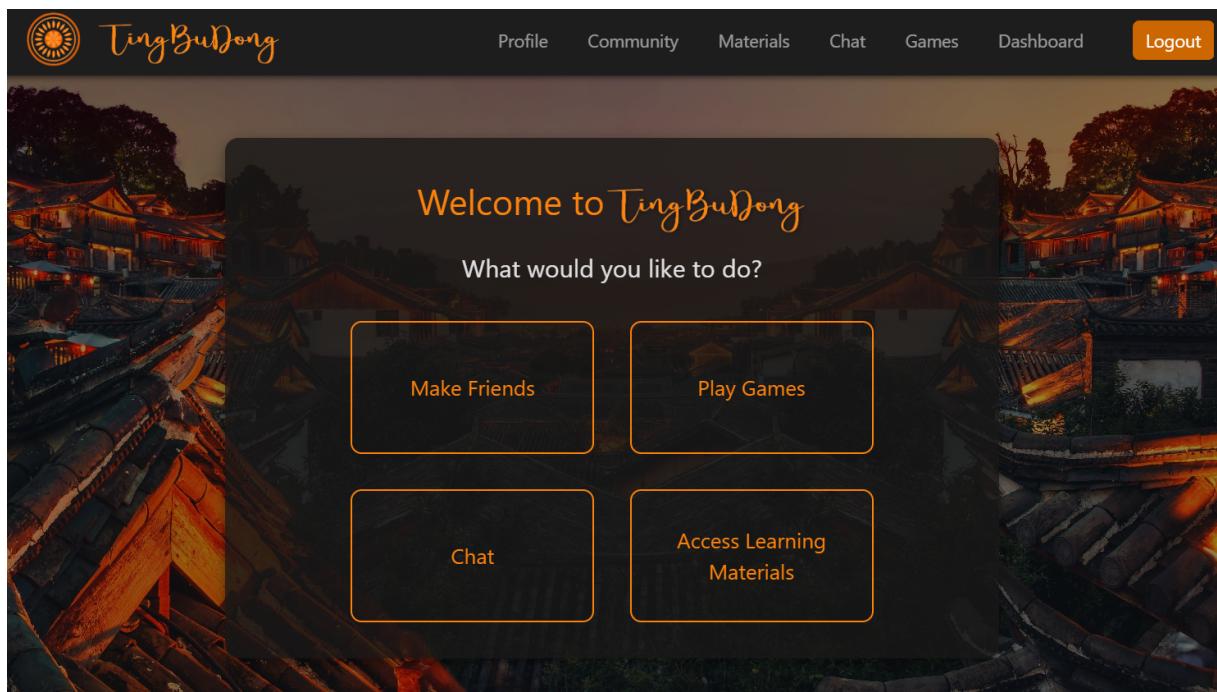
5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



5.3. ábra. Bejelentkezés oldal.

Kezdőoldal - bejelentkezés után

Sikeress autentikáció után egy másik kezdőoldal jelenik meg (LoggedInHome, lásd az 5.4. ábra), amely a rajta található gombokkal lefedi az alkalmazás főbb részeit, ezáltal ismertetve az applikáció nyújtotta lehetőségeket: barátkozás, játékok, csevegés és tananyagok. Mindegyik gomb az alkalmazás megfelelő részéhez navigál. Az alkalmazás különböző részei a navigációs sávon keresztül is elérhetők, ez a 4 opció mellett megjelenik még a profil oldal és a mérföldkövek megosztására szolgáló Dashboard is.



5.4. ábra. Belső kezdőoldal (LoggedInHome).

5. FEJEZET: A TINGBUONG MŰKÖDÉSE

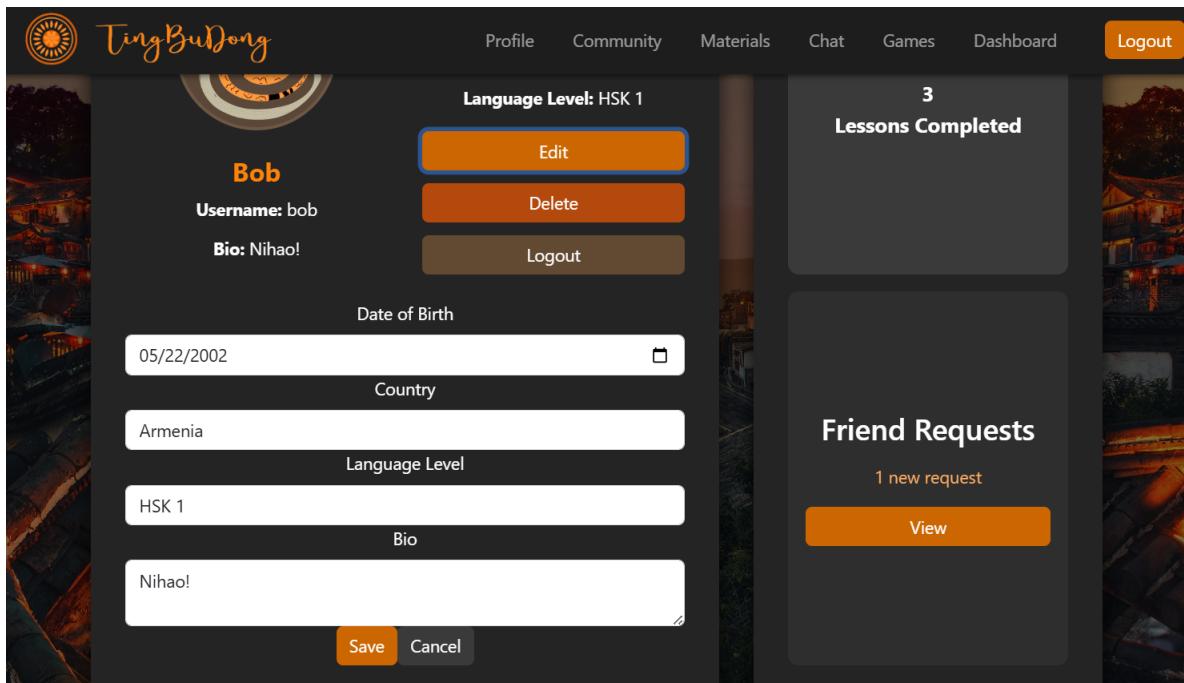
Profil

A navigációs sáv mentén haladva az első oldal a felhasználói profil (lásd az 5.5. ábra). A Profil oldal tartalmazza a felhasználó személyes adatait, amiket regisztrációkor megadott magáról. Itt lehetősége van ezeket az információkat módosítani (lásd az 5.6. ábra), de akár törölni is tudja a teljes felhasználói fiókját. Ezen funkcionálisok mellett az oldalon látható a profilkép is, a teljesített leckék száma és az esetlegesen bejövő baráti jelölések. Annak érdekében, hogy a diákok tudjanak egymáshoz kapcsolódni, de mégis el lehessen kerülni azt, hogy valamelyen másokra káros vagy sértő tartalmat osszanak meg, a profilképet nem lehet megváltoztatni. Az alkalmazás tervezésénél külön hangsúly volt fordítva arra, hogy a felhasználók interakciója valamennyire felügyelt módon történjen (például csak a barátokkal lehet üzenetet váltani vagy csoportos chatszobákat kialakítani), hogy el lehessen kerülni a negatív szándékú megnyilvánulásokat. A profilkép a születési év alapján kerül beállításra, mindenki a kínai horoszkópját ábrázoló állatos kép lesz a profiljához rendelve.



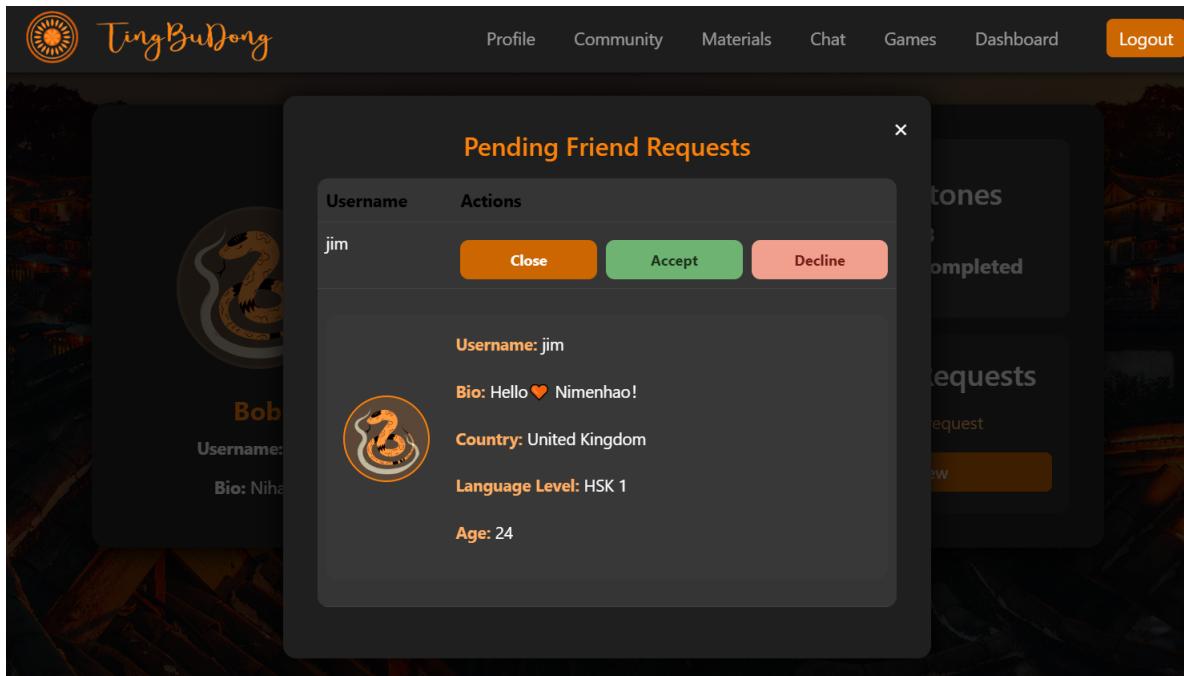
5.5. ábra. Felhasználói profil.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



5.6. ábra. Felhasználói profil: szerkesztés.

A baráti jelölések a View gomb megnyomásával egy pop-up ablakban jelennek meg (lásd az 5.7. ábra), amelyet lenyitva látni lehet a küldő profiladatait, illetve el lehet fogadni vagy el lehetutasítani a felkérést.

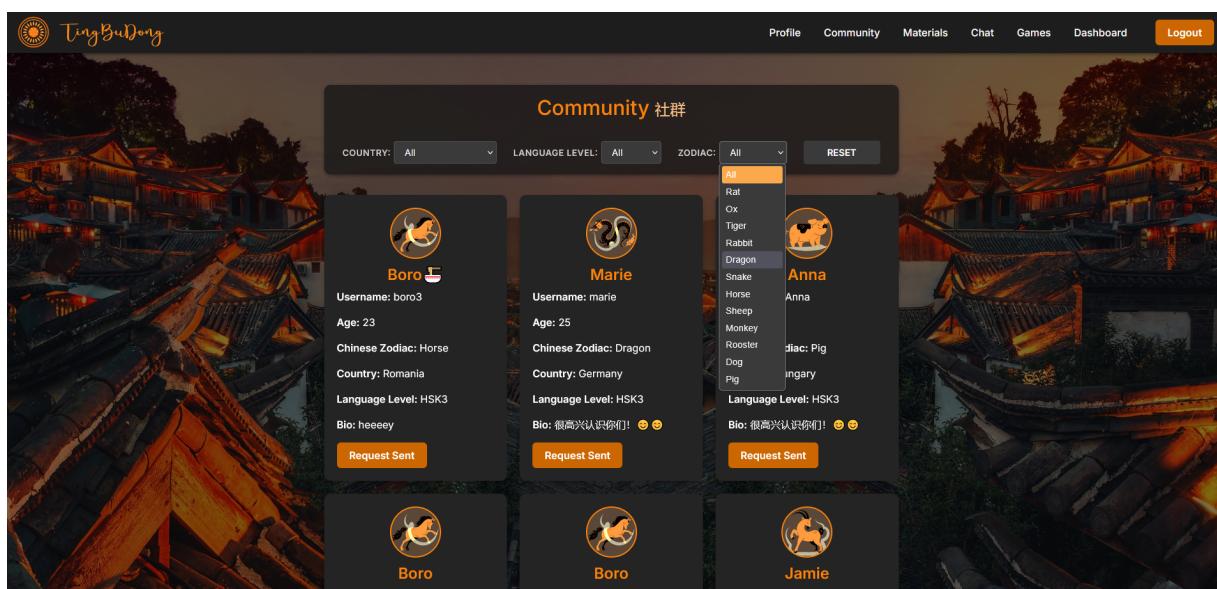


5.7. ábra. Felhasználói profil: baráti jelölések.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

Közösségek

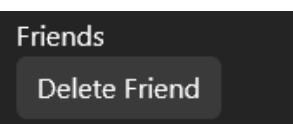
A következő a Community oldal, ami a többi felhasználó megismerésének ad teret. Az összes diák profiladata meg van itt jelenítve, továbbá lehetőség van szűrési műveletek alkalmazására is (lásd az 5.8. ábra). A filterezés történhet ország, nyelvtudási szint, kínai horoszkóp vagy ezek kombinációja szerint. Ha valamelyik profil szímpatikus a felhasználónak, már helyben el is küldheti a baráti felkérést, ilyenkor a *Send Friend Request* gomb felirata átvált *Request Sent*-re. Fordított esetben, ha a felkérés fogadóoldalán van a diák, valós időben fog frissülni a küldő profilja, a narancssárga gomb helyett egy zöld Accept és egy piros Decline gomb jelenik meg egymás mellett (lásd az 5.9a. ábra). Ha valakivel már ki lett alakítva a baráti státusz, akkor a Friends felirat van megjelenítve, valamint egy *Delete Friend* feliratú gomb (lásd az 5.9b. ábra) is látható.



5.8. ábra. Közösségek.



(a) Fogadott jelölés.



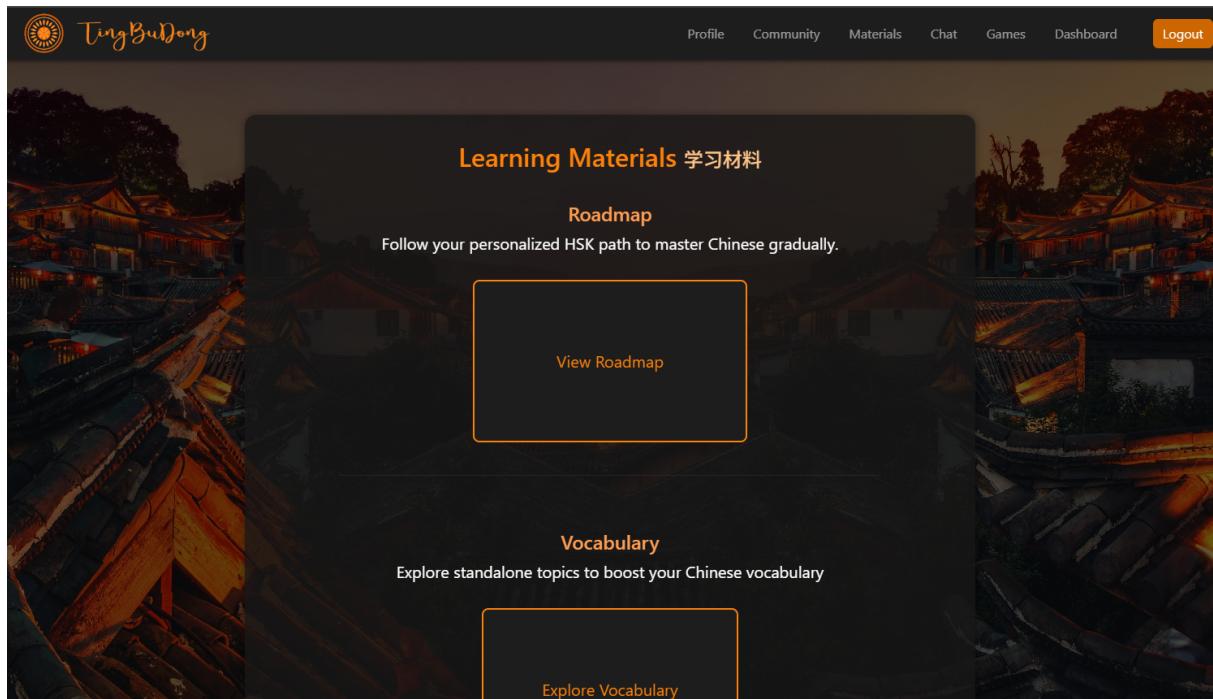
(b) Már barátok.

5.9. ábra. Különböző gombok baráti jelölések esetén.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

Tananyagok

A tananyagok oldalán a kezdőoldalhoz hasonlóan navigációs gombok találhatók, az egyik a Roadmap, a másik a Vocabulary oldalra vezet (lásd az 5.10. ábra).



5.10. ábra. Tananyagok oldala.

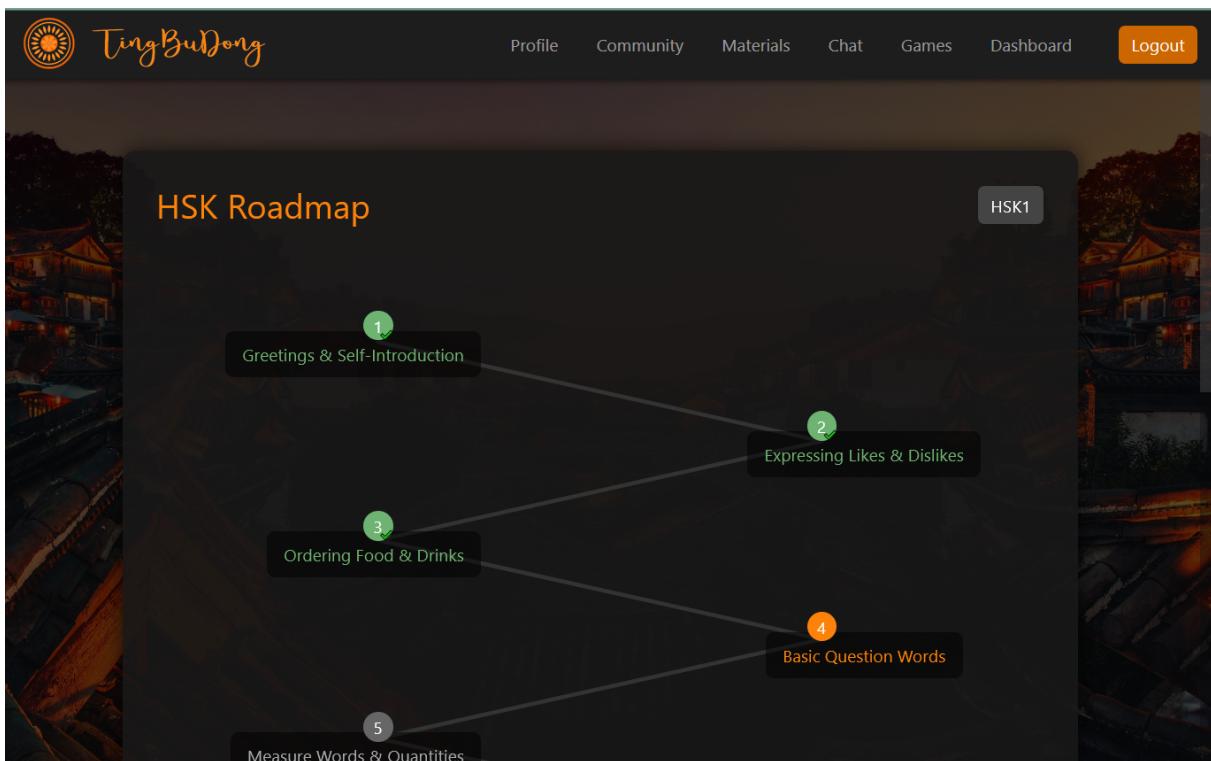
Roadmap

Az 5.11. ábrán látható Roadmap oldalon az adott HSK szinthez tartozó leckék találhatók lineáris elhelyezésben: zölddel vannak jelölve a teljesített leckék, narancssárgával az aktuális lecke, szürkével pedig a hátralevő leckék. A lecke címére kattintva lehet megtekinteni annak tartalmát, a jobb felső sarokban található filterrel pedig hozzá lehet férni a többi HSK szinthez tartozó tematikákhoz is.

Lecke

A leckék hasonló struktúrával rendelkeznek mind: először ismertetve vannak a fő nyelvtani és mondattani pontok, majd egy párbeszéd következik (lásd az 5.12. ábra). A párbeszéd mondatai 3 különböző módon is fel vannak tüntetve, azért, hogy minél könnyebben érhető legyen: kínai karakterekkel, a kínai karakterek romanizált változatával (pinyin), illetve angolul is. A mondat melletti hang ikonra kattintva a helyes kiejtést is meg lehet hallgatni.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



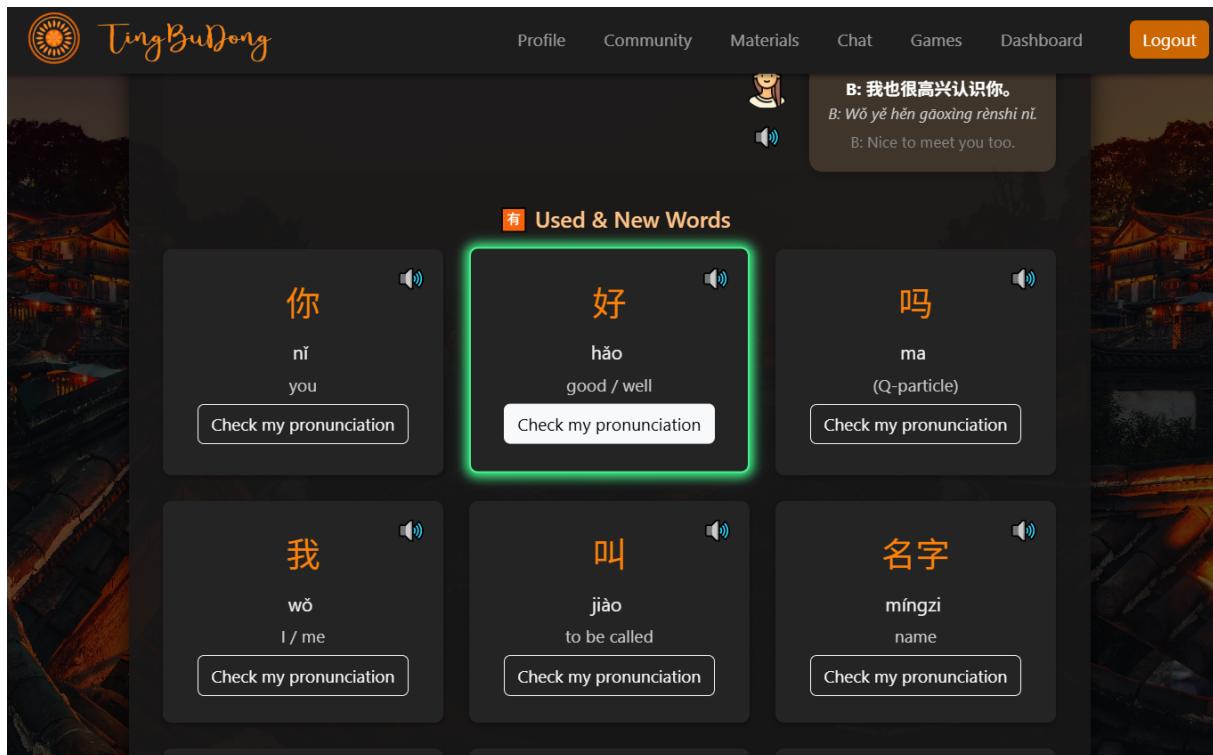
5.11. ábra. Roadmap oldal.

The screenshot shows the 'Greetings & Self-Introduction' lesson page. At the top right, there is a 'Logout' button. Below it, a 'Back to Roadmap' button is visible. The main title 'Greetings & Self-Introduction' is displayed, followed by the subtitle 'Say hello, introduce yourself, and ask someone's name.' A lightbulb icon indicates key structures: '你好' (nǐ hǎo) – hello, '我叫...' (wǒ jiào...) – my name is..., and '...吗?' (ma?) – yes/no question particle. Below this, a 'Practice Dialogue' section shows a conversation between two characters. On the left, a character says 'A: 你好!' (nǐ hǎo!) with the English translation 'Hello!' and a speaker icon. On the right, another character says 'B: 你好，我叫李华。' (nǐ hǎo, wǒ jiào Lǐ Huá.) with the English translation 'Hello, I call Li Huá.' and a speaker icon.

5.12. ábra. Lecke oldal: téma és párbeszéd.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

A párbeszédben található szavak egyesével is megjelennek az oldal alján a Used & New Words rész alatt. minden szókártyán 2 gomb található: az egyikkel a helyes kiejtést lehet meg-hallgatni, a másikkal pedig a diákok a saját kiejtésük helyességét tudják tesztelni. Abban az esetben, ha jól ejtik ki, a kártya egy zöld kerettel lesz ellátva 1 másodpercig (lásd az 5.13. ábra), ha nem, akkor pedig pirossal. Akárhány próbálkozás meg van engedve.



5.13. ábra. Lecke oldal: szókártyák.

Vocabulary

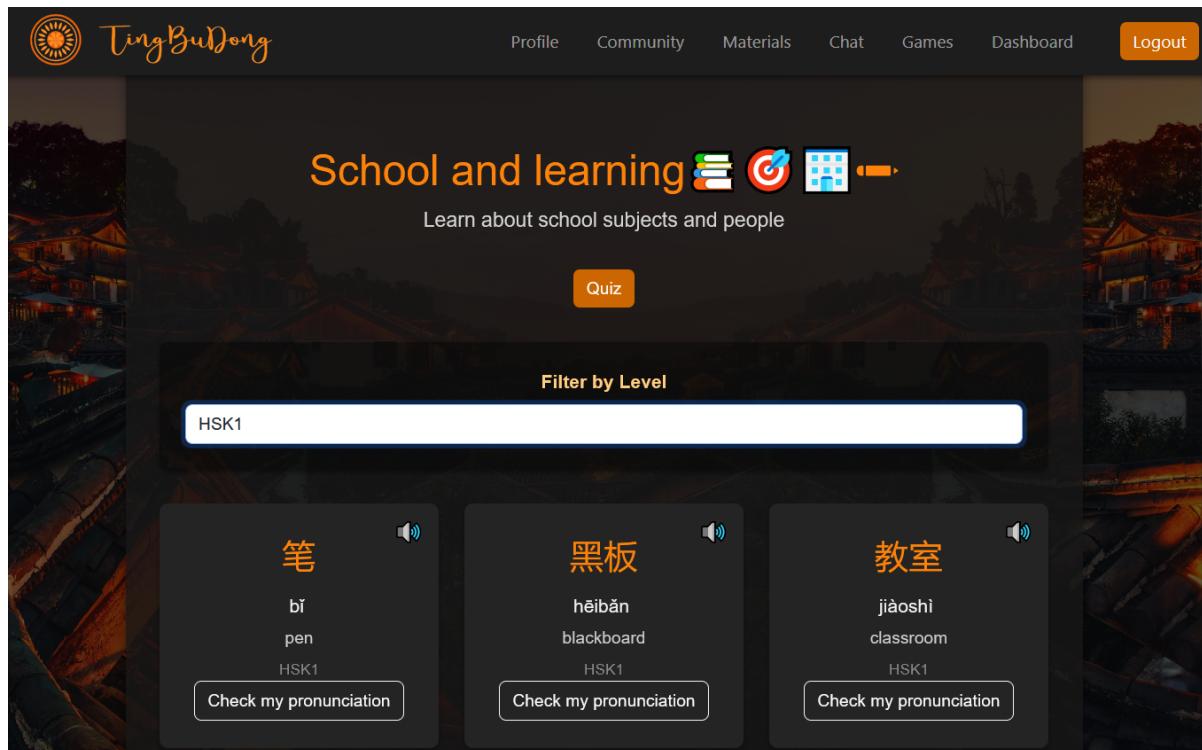
A szókincs gyarapításában a Vocabulary oldal van a diákok segítségére, ahol különböző tematikák közül lehet választani, ezek között megjelenik az iskola és tanulás, természet, gasztronómia, igék (lásd az 5.14. ábra). 9 kategória áll rendelkezésre.

Ha a diákok rámegy egy választott kategória oldalára, például az 5.15. ábrán található School and learning téma-re, akkor láthatja, hogy a kategóriához tartozó szavak szókártyákon vannak feltüntetve (a leckékhez hasonlóan), ábécé sorrendben. A szavak kiejtését meg lehet hallgatni, a saját kiejtést tesztelni, illetve a szavak HSK szint szerinti szűrésére is van lehetőség. Mindezek mellett, kvízkérdezések által is próbára teheti a diákok az elsajátított tudását, ezt a Quiz gombra kattintással lehet elérni.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



5.14. ábra. Vocabulary: Tematikák.



5.15. ábra. Vocabulary: iskola és tanulás tematika.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

A kvízben 4 opción közül az angol szó helyes kínai megfelelőjét kell kiválasztani (lásd az 5.16. ábra). A helyes válaszok száma a jobb felső sarokban van számítottartva, a teszt pedig kétféleképpen érhet véget: 10 egymást követő helyes válasz elérésekor, vagy ha a diákok helytelenül válaszolnak egy kérdésre. Mindkét esetben egy Game Over oldal jelenik meg, amelyről lehetőség van visszamenni a tematika oldalára, vagy új kvízt lehet kezdeni.

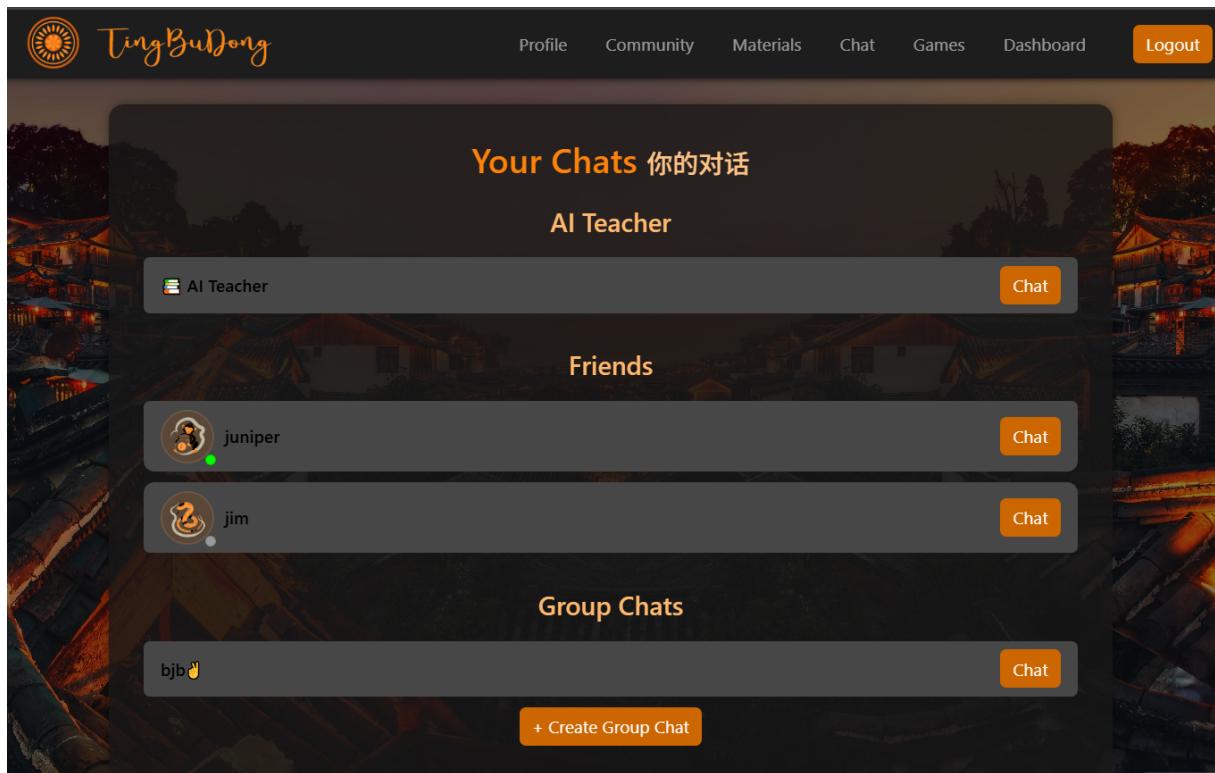


5.16. ábra. Vocabulary: több válaszlehetőséges kvíz.

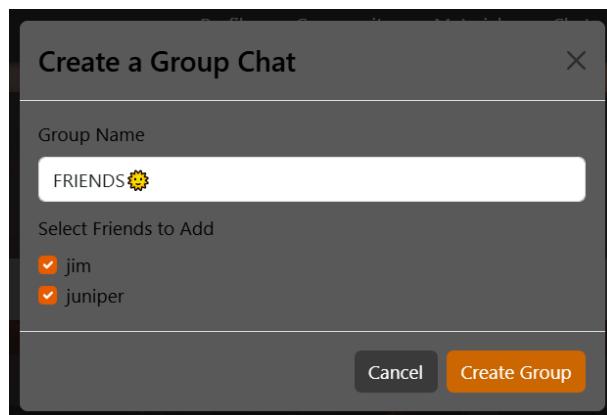
Csevegések

A csevegések 3 különböző módon történhetnek: lehet privát chatszobában üzenetet váltani a barátokkal, csoportos chatszobát kialakítani több baráttal, illetve igénybe lehet venni a beépített AI chatbotot, ami egy interaktív tanárként működik. Ennek a 3 kategóriának az elhelyezését az 5.17a. ábrán lehet megtekinteni, amelyen az is látszik, hogy a barátok online állapota színkódokkal van jelölve: szürke kör jelenik meg az inaktív felhasználók profilképe mellett, zöld pedig a jelenleg elérhető személyeknél. Valamennyi chatszobát a Chat feliratú gombra kattintással lehet megnyitni. A Create Group Chat gomb egy pop-up ablakot jelenít meg a csoportos chatszobák létrehozására (lásd az 5.17b. ábra).

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



(a) Chatszóbák megjelenítése.

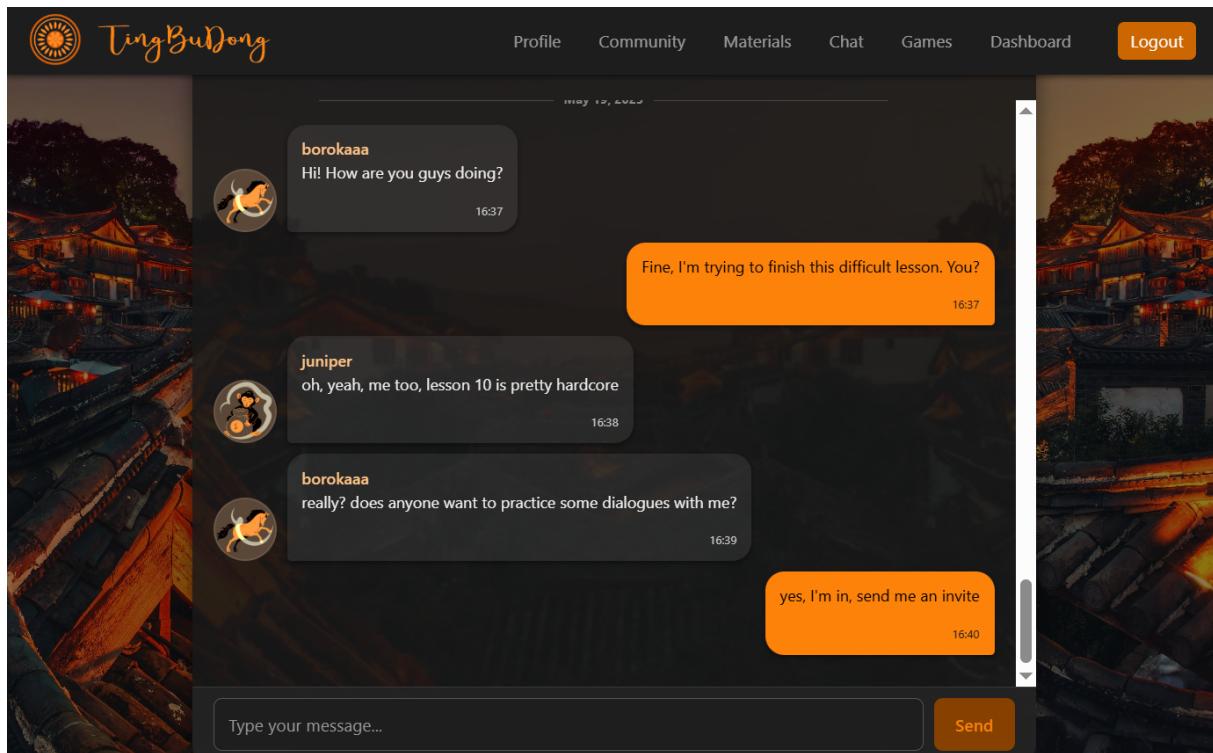


(b) Csoportos chatszóba létrehozása.

5.17. ábra. Chatszobák lista.

A chatszobába belépve hasonlóan néznek ki az 1-1 szobák és a csoportos szobák: narancssárgával jelenik meg jobb oldalon az aktuális felhasználó által küldött üzenet, a csevegő konténer ellenkező oldalán pedig profilképpel ellátva a barát/barátok által küldött üzenetek (lásd az 5.18. ábra). A chatszoba felépítése hasonló az összes széleskörben használt chatalkalmazáshoz, ezáltal is segítve az egyszerű, intuitív használhatóságot.

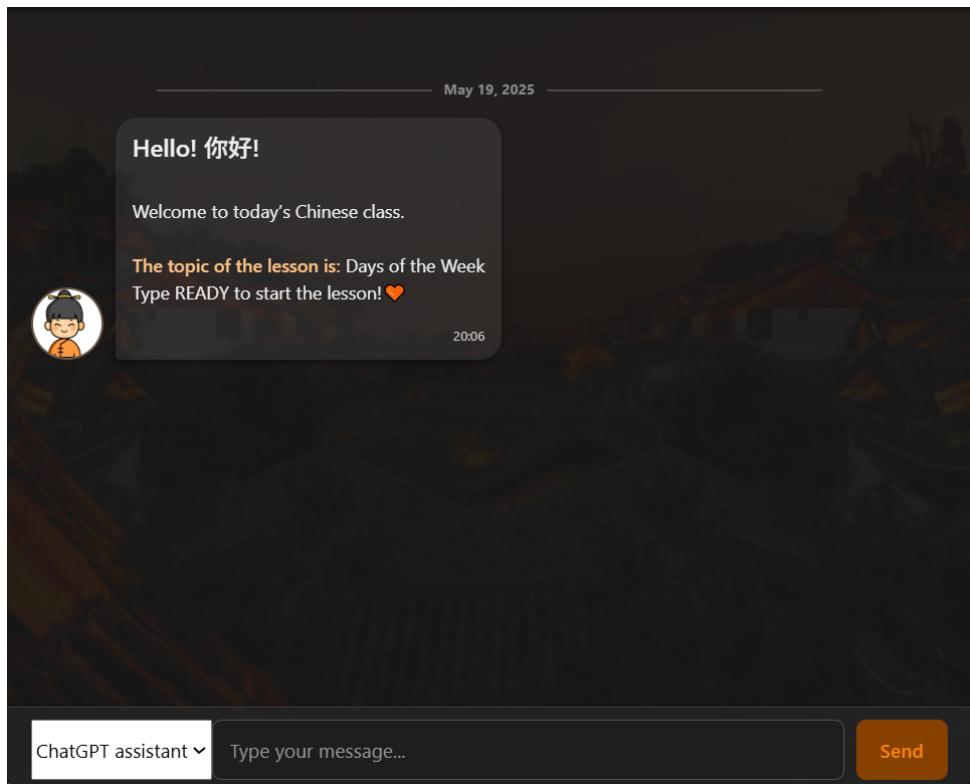
5. FEJEZET: A TINGBUDONG MŰKÖDÉSE



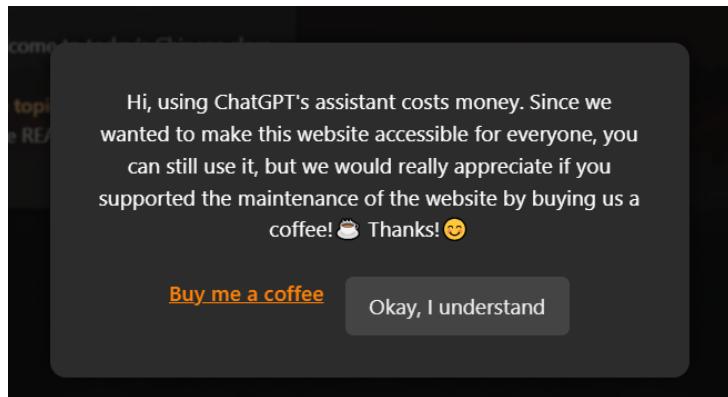
5.18. ábra. Chatszoba felépítése.

A chatbot csevegőszobánál egy üdvözlőszöveg található az MI tanártól, illetve felvezet egy random tematikát, ami köré a lecke épülni fog (lásd az 5.19a. ábra). A felhasználó a READY üzenet elküldésével tudja megkezdeni a "tanórát", ezután a tanár ismertet egy új szót, majd 1-2 kisebb, egyszerű gyakorló feladatot küld el a diáknak (például mondatalkotás, hiányzó szó beillesztése, többválaszos kvízkérdés). Két különböző nyelvi modellt lehet használni a chatbotnál: Gemma 2B és GPT 3.5 turbo. Mivel az OpenAI API applikációba való integrálása nem díjmentes, ezért az alkalmazás valós környezetben való fenntarthatósága péntbeli adományokkal valósulna meg. A támogatásra az alkalmazásnak létrehozott ko-fi oldalon van lehetőség (<https://ko-fi.com/tingbudong>), de mivel a TingBuDong egyik fő célja az, hogy bárki használhassa, a támogatás becsületkassza alapon működik, teljesen fakultatív. A támogatásra való felhívás egy pop-up ablakban jelenik meg, amikor a nyelvi modell beállításoknál a GPT modellt választja ki a felhasználó (lásd az 5.19b. ábra).

5. FEJEZET: A TINGBUDONG MŰKÖDÉSE



(a) Tanár chatbot csevegőszoba.



(b) Pop-up ablak az anyagi támogatásra való felhívásra.

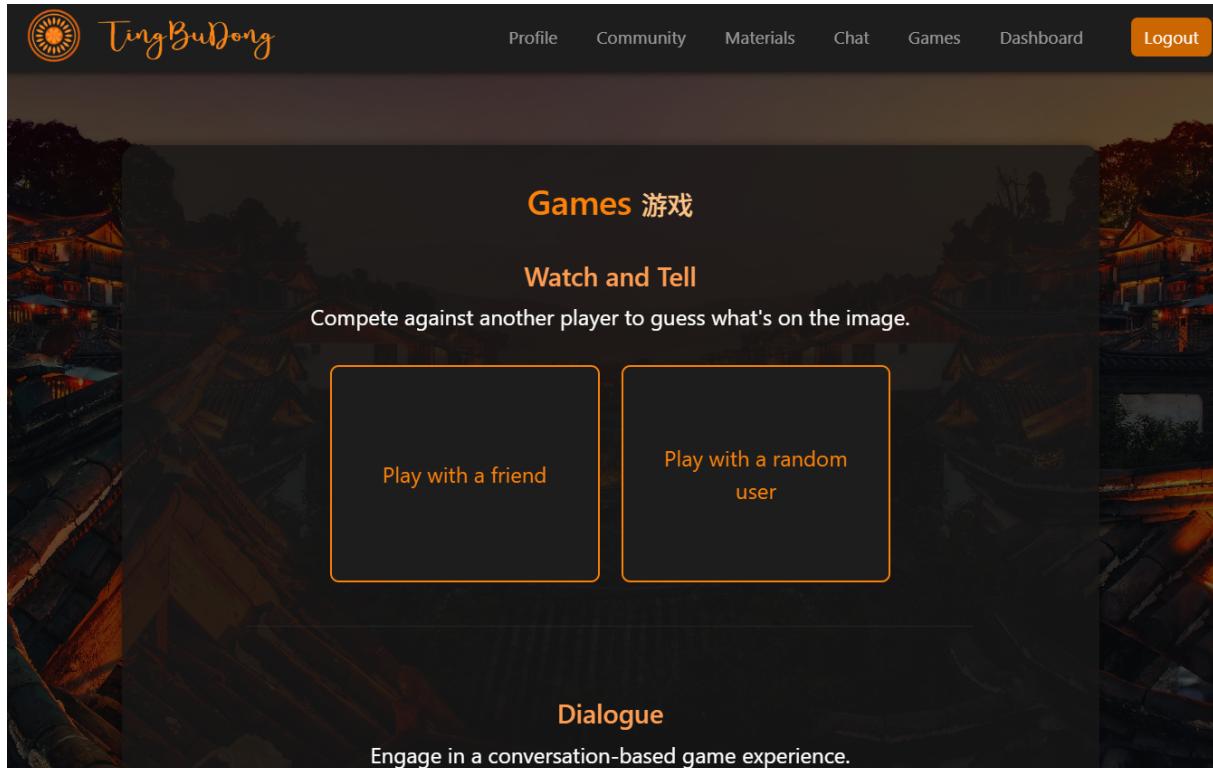
5.19. ábra. Tanár chatbot.

Játékok

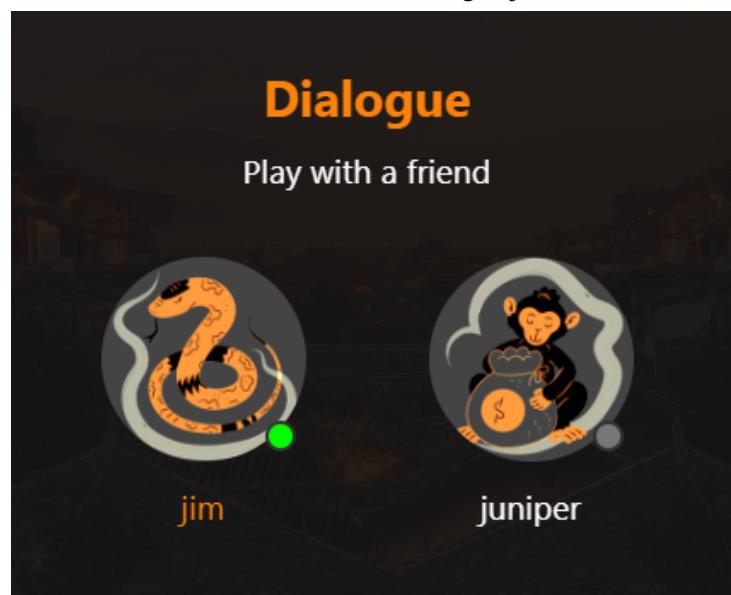
A TingBuDong applikáció két különböző nyelvtanuló játékot tartalmaz, amint ez az 5.20a. ábrán is látszik: a Watch and Tell, illetve a Dialogue játékok. Mivel ezek két személyre kézszültek, az egyik opción az, hogy a felhasználó valamelyik barátjával játsszon. Meghívás alapon működik, vagyis a Play with a friend gombra való kattintással látható lesz a barátok listája, az

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

elérhetőségi körrel együtt (lásd az 5.20b. ábra). Itt az aktuálisan elérhető barátok közül lehet meghívót küldeni valamelyiknek, majd amint beleegyezik a részvételbe, indul is a játék.



(a) Watch and Tell és Dialogue játékok.



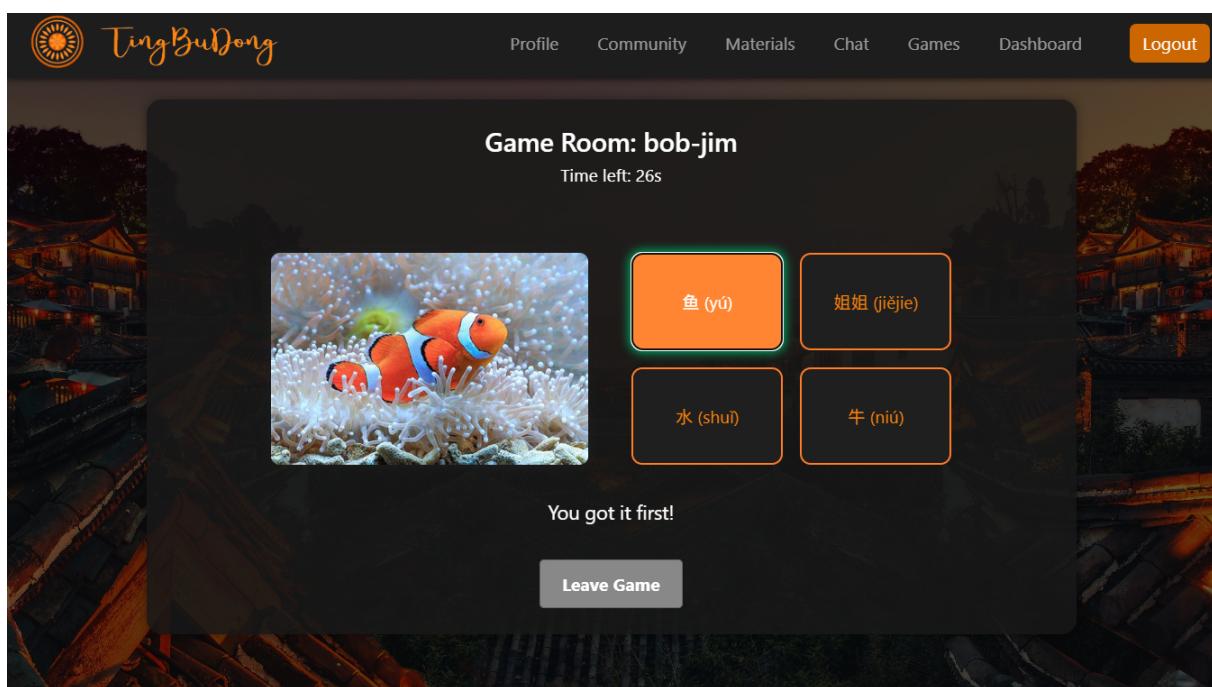
(b) Játékmeghívó küldése barátoknak.

5.20. ábra. Játékok oldal.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

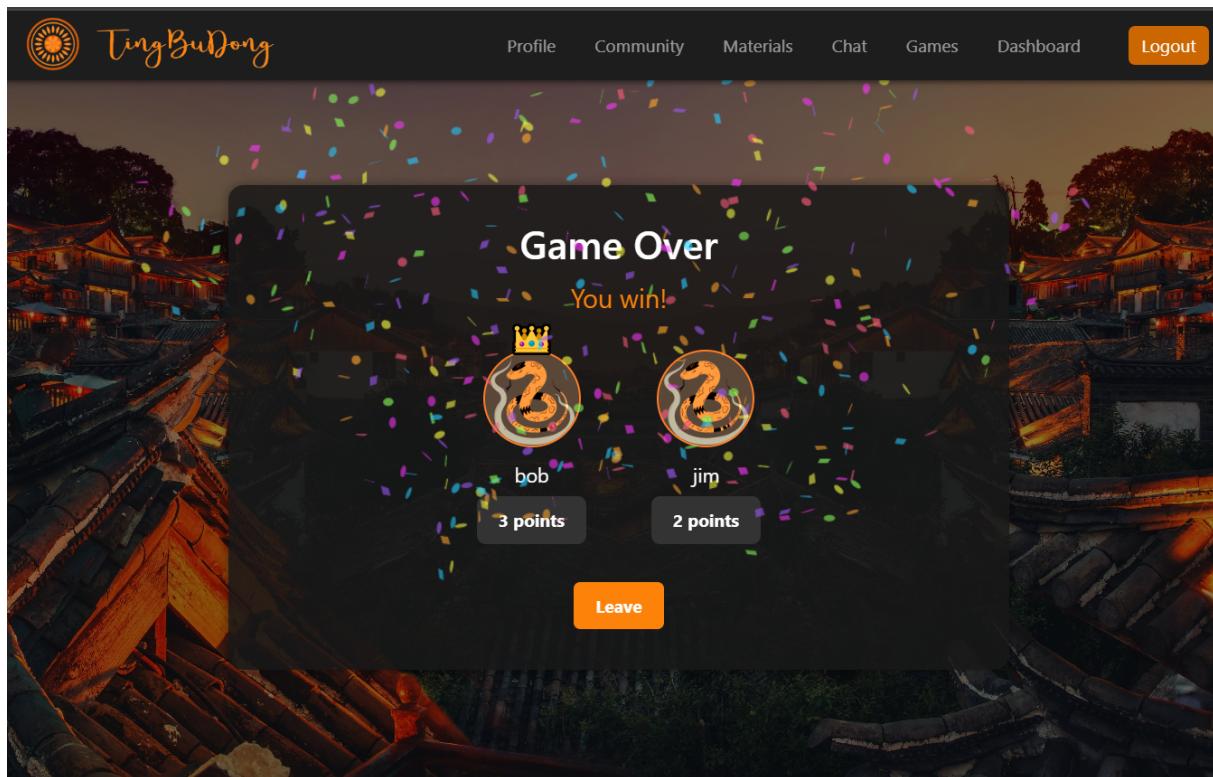
Watch and Tell

A Watch and Tell játék lényege a megjelent fényképen található tárgy, személy vagy tevékenység megnevezése. A 4 válaszlehetőség kínai karakterekkel, illetve ezeknek a romanizált változatával látható (lásd az 5.21. ábra). Az a játékos, aki a két résztvevő közül gyorsabban kattint a helyes válaszra, kap egy pontot. A megjelölt válasz körül egy színes keret jelenik meg: zöld, ha helyes a válasz; piros, hogyha helytelen; narancssárga, ha az ellenfél találta el hamarabb a megoldást. A játék 30 másodpercig tart, majd az idő leteltével ki lesz hirdetve az eredmény: a nyertes képernyőjén konfetti jelzi a győzelmet (lásd az 5.22. ábra). Ha a játékos egyetlen barátja sincs online, vagy ismeretlen emberekkel is össze szeretné vetni a tudását, akkor meghívhat egy jelenleg elérhető, szerver által random választott személyt is a játékra. Ez az egyetlen része az alkalmazásnak, aminek keretén belül direkt interakcióba lehet kerülni egy olyan személlyel, akiivel még nem áll baráti státuszban a felhasználó.



5.21. ábra. Watch and Tell játék.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE



5.22. ábra. Watch and Tell játék eredményhirdetés.

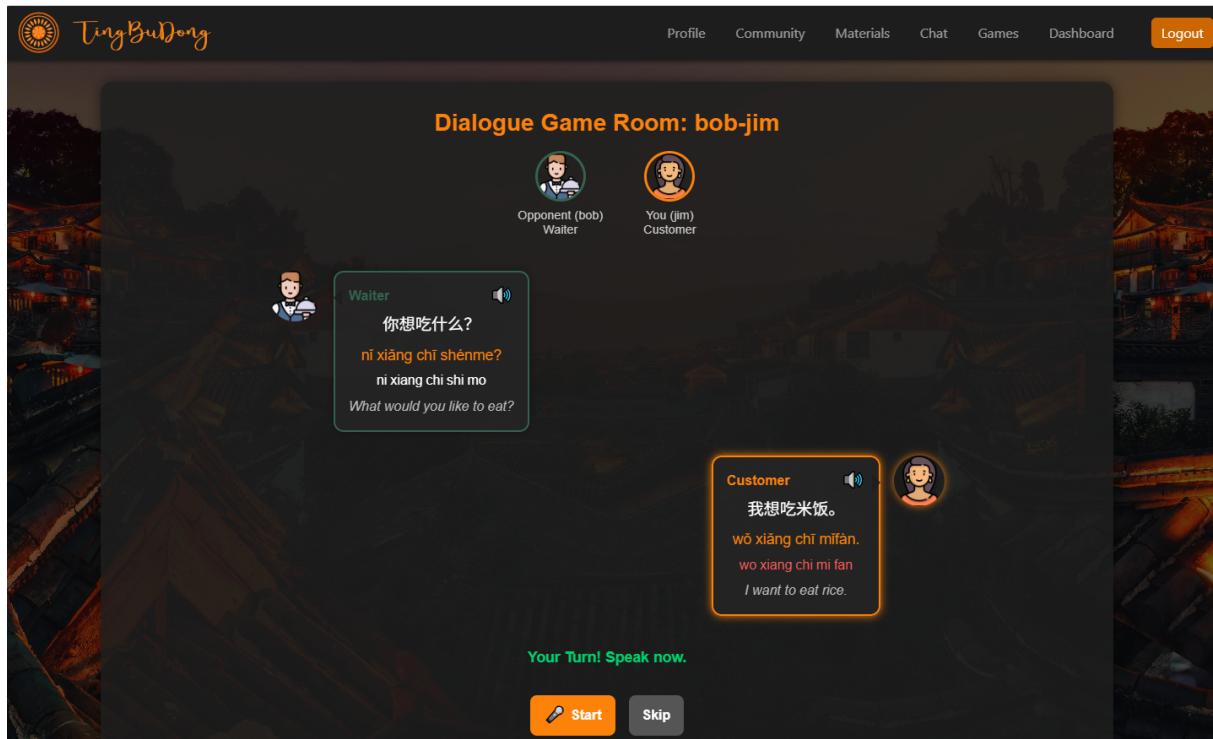
Dialogue

A Dialogue játék, ahogy a neve is utal rá, párbeszédek gyakorlására készült. Az előbbi kompetitív jellegű játékkal ellentétben, itt leginkább a kollaboráció alapszik a gyakorlat. Egy random módon választott párbeszéd két szerepköre el lesz osztva a két játékos között, majd mindenketen arra kell törekedjenek, hogy a saját karakterük mondatait helyesen ejtsék ki.

A játék teljes időtartama alatt egy hanghíváson keresztül hallja egymást a két résztvevő. A helyes kiejtés meghallgatható a hang ikonra kattintva. Amikor a diák készen áll a kiejtése tesztelésére, akkor a Start gombra kell kattintania (lásd az 5.23. ábra). Mikor befejezte a mondatot, akkor a Stop listening gombra nyomva állítja le a hangfelismerést. Valós időben a képen pirossal feltüntetett mondat szavai helyes kiejtés esetén egyesével zölddé változnak, így ha nem teljesen helyes a válasz, akkor pontosan lehet látni, hogy melyik szó nem volt megfelelően kimondva. Ha az egész mondat kiejtése helyes, vagy ha a Skip gombra kattint a játékos, akkor automatikusan megjelenik a következő mondat, ezáltal a másik játékos fog következni. Tehát eltűnik a mikrofonos gomb addig, amíg ismét nem az adott játékoson lesz a sor. A párbeszédek sokféle élethelyzetet, szituációt és szereplőpárost lefednek, van például pincér-vendég, orvos-páciens,

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

szülő-gyerek vagy tanár-diák beszélgetés is. Mivel a játékosok hanghívást folytatnak a játék ideje alatt, a biztonságos légkör fenntartása érdekében ezt csak barátokkal lehet játszani. Ha a felhasználó olyankor is szeretne gyakorolni, mikor egy barátja sincs online, akkor megteheti ezt az egy személyes verzióval, amikor az egyik szerepkört a számítógép tölti be, automatikusan kiejtve a második szereplő mondatait.



5.23. ábra. Dialogue játék.

Fórum és Dashboard

A **Fórum** oldalon (lásd az 5.24. ábra) a közösség tagjainak mérföldköveit lehet megtekinteni. Valós időben frissül, ha valaki teljesített egy leckét, valamelyik játékkal gyakorol, vagy legyőzte ellenfelét. A felhasználók neve narancssárgával ki van emelve, erre kattintva megjelenik a profiljuk egy popup ablakban, ahol baráti jelölést is lehet küldeni.

A **Dashboard** (lásd az 5.25. ábra) a tematikus kvízekből gyűjthető pontok alapján mutat egy rangsort. A felhasználónevekre kattintva itt is meg lehet tekinteni a profiladatokat. Mindkét oldal célja, hogy információt lehessen szerezni a többi felhasználó teljesítményéről.

5. FEJEZET: A TINGBU DONG MŰKÖDÉSE

The screenshot shows the 'HSK1 Forum 论坛' section of the TingBuDong website. At the top, there's a navigation bar with links for Profile, Community, Materials, Chat, Games, Forum, Dashboard, and Logout. Below the navigation, the title 'HSK1 Forum 论坛' is displayed, followed by 'Recent Milestones'. There are five entries in a list:

- finn** completed Lesson food_drink of HSK1
12/06/2025, 14:22
- bob** completed Lesson daily_verbs_1 of HSK1
12/06/2025, 14:21
- jim** has won a Watch and Tell game against **bibble**
12/06/2025, 13:54
- bibble** completed Lesson likes_dislikes of HSK1
12/06/2025, 12:18
- bibble** completed Lesson greetings of HSK1
12/06/2025, 10:21

5.24. ábra. Fórum oldal.

The screenshot shows the 'Dashboard 仪表板' section of the TingBuDong website. At the top, there's a navigation bar with links for Profile, Community, Materials, Chat, Games, Forum, Dashboard, and Logout. Below the navigation, the title 'Dashboard 仪表板' is displayed, followed by 'Leaderboard for food'. There is a table showing the top three players:

Rank	Username	Points
1 🏆	bibble	11
2 🏆	jim	9
3 🏆	bob	8

Below the food leaderboard, there is a section titled 'Leaderboard for nature'.

5.25. ábra. Dashboard oldal.

6. fejezet

Következtetések és továbbfejlesztési lehetőségek

A TingBuDong valós idejű alkalmazás fejlesztése során egy olyan platform jött létre, amely utat mutat a kínai nyelvtanulás iránt érdeklődő személyeknek. Az alkalmazás közösségi keretben teszi lehetővé a felhasználóknak, hogy kapcsolatba kerüljenek egy új nyelvvel és kultúrával, megismerkedjenek hasonló érdeklődésű emberekkel, egymást motiválva fejlődjenek. A valós idejű funkciók révén a tanulók barátjelöléseket küldhetnek, üzeneteket válthatnak privát vagy csoportos chatszobákban, közös játékokkal, kvízekkel, valamint szókártyákkal tesztelhetik kiejtésüket és nyelvtudásukat. Emellett olyan tananyagokhoz férhetnek hozzá, amelyek összhangban állnak a nyelvvizsga-elvárásokkal, így a TingBuDong által megszerzett kínai nyelvtudásuk a minden nap a gyakorlati hasznat hozhat.

Az alkalmazás tervezése és készítése során számos olyan ötlet és funkcionális felmerült, aminek beépítése javítaná a felhasználói élményt és növelné az alkalmazás hozzáadott értékét. Az ötletek sorrendiség nélkül a következők:

- Regisztráció lehetősége létező Google és Facebook fiókokkal, illetve Admin felület létrehozása,
- Gamifikált felhasználói felület, illetve több, akár csoportos (2+ személy) számára készült játékok,
- Hang- és videóhívás lehetősége automatikus feliratozással vagy hívás alatt közösen oldott feladatokkal,
- Mesterséges intelligencia magasabb szintű beépítése: személyre szabott visszajelzések, hibák számítartása, feladatok generálása,

6. FEJEZET: KÖVETKEZTETÉSEK ÉS TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

- Blog beépítése, amelyre tematikus ismeretterjesztő cikkek kerülnének fel a kínai szokásokról, gasztronómiáról, kultúráról,
- Karakterek helyes írásának módja grafikusan ábrázolva,
- Tananyagok minőségének és részletességének növelése.

A fent említett funkciók integrálásával tovább lehet csíszolni az applikációt, viszont a jelenlegi állapotában is ötvözi azokat az elemeket, amelyek szükségesek egy kezdő nyelvtanuló számára készült platformnál: tananyagok, közösség és az elsajátított tudás próbára tétele. Összegzésül tehát elmondható, hogy az alkalmazás egy járható utat biztosít ahhoz, hogy a tīng bù dōng-tól (nem értem) el lehessen jutni a tīng dōng-ig (értem).

Irodalomjegyzék

- [1] State Council of the People's Republic of China. Global statistics. https://english.www.gov.cn/statecouncil/ministries/202111/10/content_WS618b1b0dc6d0df57f98e4bf8.html, November 2021. Megtekintve: 2025-04-16.
- [2] Wen, X. Chinese language learning motivation: A comparative study of heritage and non-heritage learners. *Heritage Language Journal*, 8(3):333–335, 2011. doi: 10.46538/hlj.8.3.3. Fall 2011.
- [3] Loewen, S., Isbell, D. R., és Sporn, Z. The effectiveness of app-based language instruction for developing receptive linguistic knowledge and oral communicative ability. *Foreign Language Annals*, 53(2):209–233, 2020.
- [4] Duolingo. 2024 duolingo language report, December 2024. URL <https://blog.duolingo.com/2024-duolingo-language-report/>. Megtekintve: 2025-04-16.
- [5] Hadi Mogavi, R., Guo, B., Zhang, Y., Haq, E.-U., Hui, P., és Ma, X. When gamification spoils your learning: A qualitative case study of gamification misuse in a language-learning app. In *Proceedings of the Ninth ACM Conference on Learning @ Scale*, L@S '22, page 175–188, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391580. doi: 10.1145/3491140.3528274. URL <https://doi.org/10.1145/3491140.3528274>.
- [6] Chinestest. Hsk chinese proficiency test. <https://www.chinesetest.cn/Development>, 2025. Megtekintve: 2025-04-17.
- [7] Freepik. Chinese horoscope icons, 2025. URL https://www.flaticon.com/search?author_id=222&style_id=&type=standard&word=chinese+horoscope. Megtekintve: 2025-05-05.
- [8] OpenJS Foundation. Introduction to node.js, 2025. URL <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>. Megtekintve: 2025-04-29.
- [9] OpenJS Foundation and Express.js contributors. Express.js: Fast, unopinionated, minimalist web framework for node.js, 2025. URL <https://expressjs.com/>. Megtekintve: 2025-04-29.
- [10] Meta Platforms, Inc. React: The library for web and native user interfaces, 2025. URL <https://react.dev/>. Megtekintve: 2025-04-29.
- [11] Chacon, Scott and Straub, Ben. What is Git? URL <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>. Megtekintve: 2025-04-30.
- [12] GitLab Inc. Get started with git, 2025. URL https://docs.gitlab.com/topics/git/get_started/. Megtekintve: 2025-04-30.
- [13] Microsoft Corporation. Visual studio code – code editing. redefined, 2025. URL <https://code.visualstudio.com/>. Megtekintve: 2025-04-30.
- [14] Stack Overflow. Technology | 2024 developer survey, 2024. URL <https://survey.stackoverflow.co/2024/technology#most-popular-technologies-new-collab-tools-prof>. Megtekintve: 2025-

IRODALOMJEGYZÉK

04-30.

- [15] npm, Inc. About npm. URL <https://docs.npmjs.com/about-npm>. Megtekintve: 2025-04-30.
- [16] DB-Engines. Ranking of the most popular database management systems worldwide, as of june 2024. URL <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>. Megtekintve: 2025-04-30.
- [17] MongoDB, Inc. What is mongodb?, . URL <https://www.mongodb.com/company/what-is-mongodb>. Megtekintve: 2025-04-30.
- [18] MongoDB, Inc. Mongodb atlas documentation, . URL <https://www.mongodb.com/docs/atlas/>. Megtekintve: 2025-04-30.
- [19] Jesse Hall. Getting started with mongodb & mongoose. URL <https://www.mongodb.com/developer/languages/javascript/getting-started-with-mongodb-and-mongoose/>. Megtekintve: 2025-04-30.
- [20] Redis Documentation. Redis faq, 2025. URL <https://redis.io/docs/latest/develop/get-started/faq/>. Megtekintve: 2025-04-30.
- [21] Socket.IO Documentation. Socket.io v4 documentation, 2025. URL <https://socket.io/docs/v4/>. Megtekintve: 2025-04-30.
- [22] Pivotal Software, Inc. Rabbitmq: Messaging that just works, 2025. URL <https://www.rabbitmq.com/>. Megtekintve: 2025-05-01.
- [23] Google WebRTC Team. Webrtc: Real-time communication for the web, 2025. URL <https://webrtc.org/>. Megtekintve: 2025-05-01.,
- [24] Docker, Inc. What is docker?, 2025. URL <https://docs.docker.com/get-started/docker-overview/>. Megtekintve: 2025-05-01.
- [25] Docker, Inc. How docker compose works, 2025. URL <https://docs.docker.com/compose/intro/compose-application-model/>. Megtekintve: 2025-05-01.
- [26] The Kubernetes Authors. Overview, 2025. URL <https://kubernetes.io/docs/concepts/overview/>. Megtekintve: 2025-05-01.
- [27] The Helm Authors. Using helm, 2025. URL https://helm.sh/docs/intro/using_helm/. Megtekintve: 2025-05-01.
- [28] The Kubernetes Authors. Minikube documentation, 2025. URL <https://minikube.sigs.k8s.io/docs/>. Megtekintve: 2025-05-01.