



Made by Teamlingo

# Mi a CodeLingo?

- Programozási kérdéseket szolgáltat gyakorláshoz
- Azonnali visszajelzés a válaszokra
- Pontgyűjtés és szintlépés
- Több programozási nyelv támogatása (C#, JavaScript, Python)

# Elkészült funkciók

- Landing page
- Regisztráció és bejelentkezés
- Practice session indítása
- Kérdések lekérése és megjelenítése
- Session befejezés és eredmény
- Admin felület
- Felhasználó oldal
- Hosting
- Leaderboard



# Nem készült el és eltérések

- Szerencsére minden elkészült
- Eltérések:
  - Kérdés típus választás

# Extra funkciók

- Swagger dokumentáció minden endpointhoz
- Seed adatok automatikus betöltése induláskor
- Részletes API specifikáció dokumentáció



## Welcome back, JohnSmith!

Continue your coding journey.

Questions Answered: 0 | Weekly Accuracy: 0% | Streak:  
0 days

[Start Practice](#)

JavaScript

```
function fibonacci(n) {  
  if (n ≤ 1) return n;  
  
  let prev = 0, curr = 1;  
  
  for (let i = 2; i ≤ n; i++) {  
    let next = prev + curr;  
    prev = curr;  
    curr = next;  
  }  
  
  return curr;  
}  
  
// Calculate 10th Fibonacci number  
console.log(fibonacci(10)); // 55
```

# Oktató Bemutató

---

# **Használt technológiák**

- Backend: ASP.NET, Entity Framework Core, Identity
- Frontend: Angular, Bootstrap
- Adatbázis: In-memory (fejlesztés), SQL Server (prod terv)
- Verziókövetés: Git + GitHub



- Backend (NuGet):
  - Microsoft.AspNetCore.Identity.EntityFrameworkCore
  - Microsoft.EntityFrameworkCore.InMemory
  - Microsoft.EntityFrameworkCore.SqlServer
  - Microsoft.EntityFrameworkCore.Proxies
  - Microsoft.AspNetCore.Authentication.JwtBearer
  - System.Text.Json
  - Swashbuckle.AspNetCore
- Frontend (NPM):
  - @angular/core, @angular/router, @angular/forms
  - @angular/common/http
  - bootstrap
  - Rxjs

# **Fejlesztés menete**

1. Epicek és User Story-k definiálása
2. Sprint 1 – Api + Design és landing page
3. Sprint 2 – Backend + Frontend implementáció
4. Sprint 3 – Hibák kisímitása + Hosting

# Verziókövetési stratégia

- GitFlow: main, develop, feature/\*, *bugfix*/\*
- PR szabály: minimum 3 reviewer (Architect + PO + 1 dev)
- Sprint végén: develop → main merge

- Szakmai oldal:
  - FE–BE párok az API kontraktusoknál
  - Architect review minden PR-nél
  - Dokumentáció-vezérelt fejlesztés
- Emberi oldal:
  - Heti 2 meeting (vasárnap + szerda)
  - Mindenki aktívan részt vett
  - Nyitott kommunikáció Teams-en

# Munkamegosztás

- Szczuka Bendegúz | Architect | ~ 35%
- Tolnai Olivér | Frontend | ~ 11%
- Prohászka Ádám | Frontend | ~ 11%
- Kóbori Bence | Frontend | ~ 11%
- Homonnai Mihály | Backend | ~ 11%
- Zsebeházi Máté | Backend | ~ 11%
- Tisza Tamás | Manager/PO | ~ 10%

# Probléma jegyzőkönyv és tanulságok

- Projekt teljesen új – Első szakasz issue tervezés
- Tickett összeakadás – Egy emberen legyenek a hasonló ticketek
- Ráépülő issuek elakadnak – Priorizálás
- Master-re nem mentek fel a dolgok az elején – Zárás előtt feltétel
- Sok extra dolog az apikban – Törekedjünk az egyszerűsége
- Burnup látszólag rosszul áll – Issuek legyenek rendszeresen lezárva

**Köszönjük a figyelmet!**

---