OM azonosító: 203060/015
Felnőttképzési nyilvántartás száma: E-001290/2015

Cím: 3525 Miskolc, Palóczy László utca 3.
Telefon: +36 46 500 930
E-mail: kando@kkszki.hu
Weblap: https://www.kkszki.hu

Miskolci SZC Kandó Kálmán Informatikai Technikum

ISLANDERS

Iskolai projekt

Szigili Edit

Lekner Norbert

Mészáros Balázs

Miért játék?

- A rohanó világban elég sok stressz gyűlik össze. A játékok ezt tudják enyhíteni.
- Mobil app és böngészős elérhetőség egyaránt lehetséges
- Könnyű kezelhetőség és egyszerű játékmenet
- Nem igényel sok idő ráfordítást
- Bárhonnan, bármikor elérhető

Játék ismertető

Sziget választás

Csata

Expedíció

Építkezés

Fejlesztés

Begyűjtés

Program fő részei

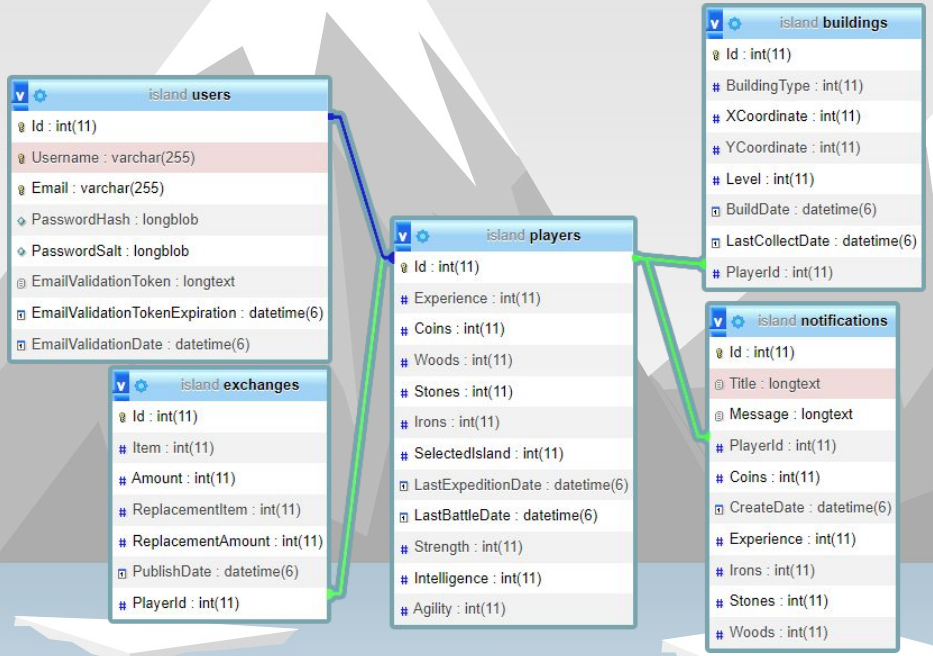
Belépés / Regisztráció / Jelszó emlékeztető
Saját profil
Menedzsment
Építés
Csata
Expedíció
Útmutató
Piac
Értesítések
Zene / hangok
Kilépés

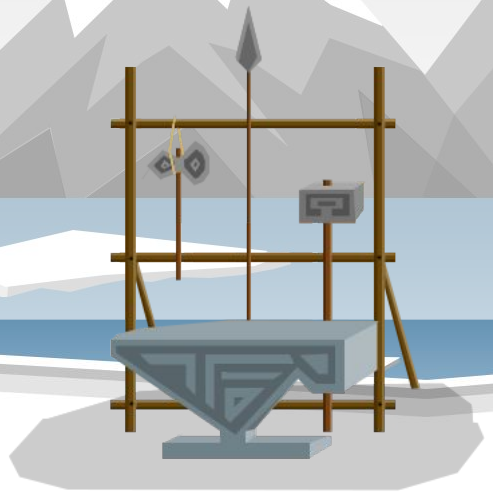
Program felépítése

- Adatbázis
  - Mysql + apache
  - Entity framework
- Backend
  - ASP.NET Core
  - Swagger
  - WebAPI / service / authentication
- Frontend
  - React Js
  - Bootstrap + Npm packages

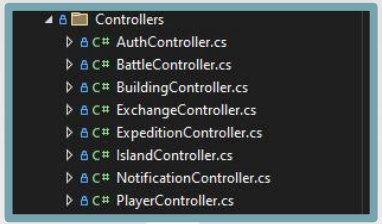
Adatbázis

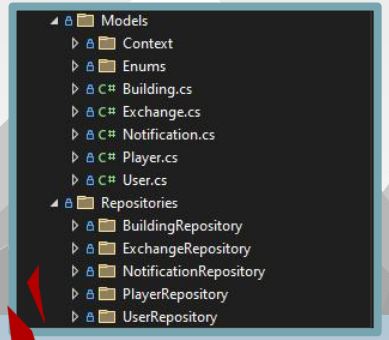
Mysql
Model first
Entity framework
5 tábla





Backend





ASP .NET 6.0 Web API

3 rétegű architektúra

Dependency Injection Design Pattern

Async

Mailkit

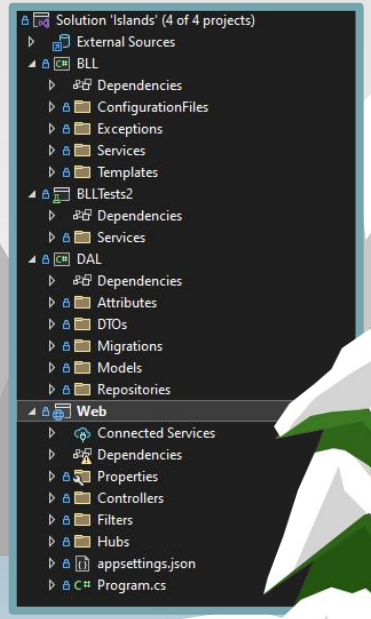
Websocket (SignalR)

JSON Web Tokens (JWT)

Titkosított jelszavak

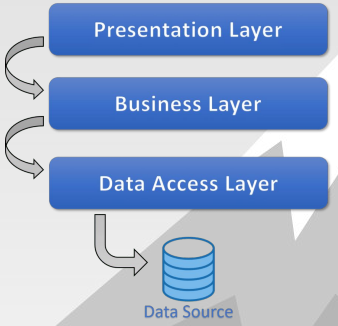
Entity Framework

Json fájlok a statikus adatokhoz



Norbert Backend

- 3 rétegű architektúra
  - Data Access Layer, Business Logic Layer, Presentation Layer
  - Előnyök a hagyományos tervezési mintákkal szemben: átlátható, karbantartható, könnyebb tesztelni
- Dependency Injection Design Pattern
  - Interface, mint függőség - könnyen cserélhető az implementáció anélkül, hogy az osztály viselkedése a külvilág számára megváltozna

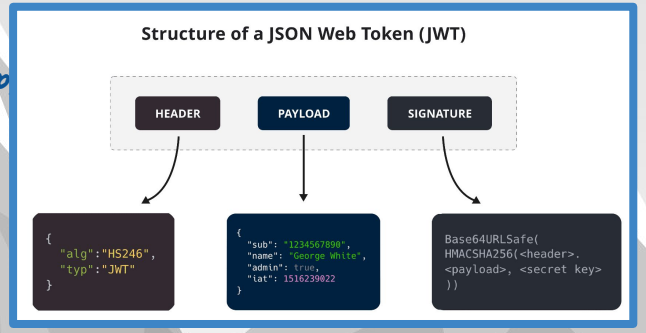


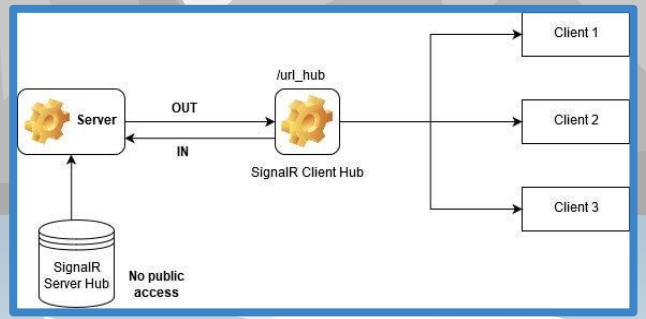

```
namespace DAL.Repositories.UserRepository
{
    6 references
    public interface IUserRepository
    {
        4 references
        Task<User> GetUserByUsernameAsync(string username);
        3 references
        Task<User> GetUserByEmailAsync(string email);
        2 references
        Task<User> GetUserByValidationTokenAsync(string token);
        2 references
        Task AddUserAsync(User user);
        5 references
        Task UpdateUserAsync(User user);
        2 references
        Task<DateTime> GetEmailValidationDateByUsernameAsync(string username);
    }
}
```

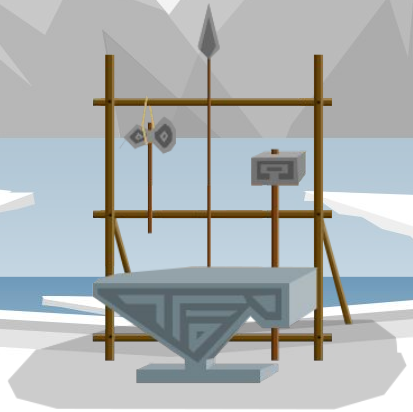
```
// Dependency injections
builder.Services.AddScoped<IAuthService, AuthService>();
builder.Services.AddScoped<IBattleService, BattleService>();
builder.Services.AddScoped<IBuildingService, BuildingService>();
builder.Services.AddScoped<IConfigurationService, ConfigurationService>();
builder.Services.AddScoped<IEmailService, EmailService>();
builder.Services.AddScoped<IExchangeService, ExchangeService>();
builder.Services.AddScoped<IExpeditionService, ExpeditionService>();
builder.Services.AddScoped<IIslandService, IslandService>();
builder.Services.AddScoped<INotificationService, NotificationService>();
builder.Services.AddScoped<IPlayerService, PlayerService>();
builder.Services.AddSingleton<IRandomProvider, RandomProvider>();

builder.Services.AddScoped<IBuildingRepository, BuildingRepository>();
builder.Services.AddScoped<IExchangeRepository, ExchangeRepository>();
builder.Services.AddScoped<INotificationRepository, NotificationRepository>();
builder.Services.AddScoped<IPlayerRepository, PlayerRepository>();
builder.Services.AddScoped<IUserRepository, UserRepository>();
```

Norbert Backend

- JSON Web Tokens (JWT):
  - Bejelentkezés után a felhasználó egy aláírt tokent kap amit az authorization során a szerver hitelesít
- Websocket (SignalR):
  - kétirányú kommunikációt tesz lehetővé a szerver és a böngésző között
  - a HTTP-vel ellentétben fenntartja a kapcsolatot







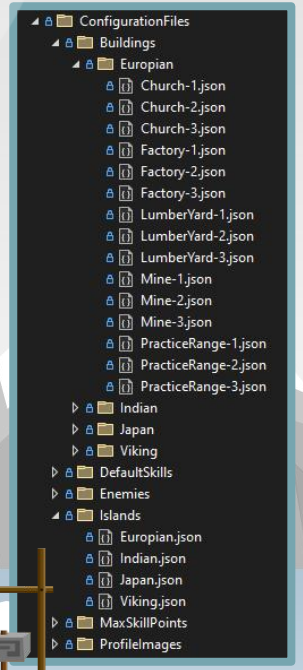
Edit Backend

Értesítések

```
2 references
public async Task RemoveNotificationAsync(Notification notification)
{
    _context.Notifications.Remove(notification);
    await _context.SaveChangesAsync();
}
```

```
2 references
public async Task<List<NotificationDto>> GetAllNotificationsByUsernameAsync(string username)
{
    return await _context.Notifications
        .Include(notification => notification.Player)
            .ThenInclude(player => player.User)
        .Where(notification => notification.Player.User.Username == username)
        .Select(notification => new NotificationDto()
        {
            Id = notification.Id,
            Title = notification.Title,
            Message = notification.Message,
            Woods = notification.Woods,
            Stones = notification.Stones,
            Irons = notification.Irons,
            Coins = notification.Coins,
            Experience = notification.Experience,
            IsOpened = notification.IsOpened,
            CreateDate = notification.CreateDate,
        })
        .OrderByDescending(notification => notification.CreateDate)
        .ToListAsync();
}
```

Balázs Backend

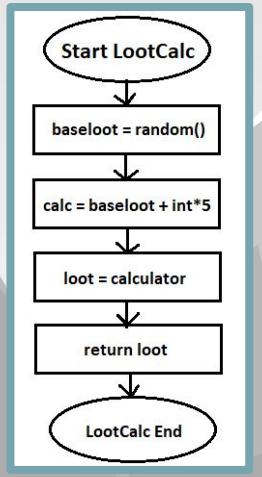


Csata és expedíció alapja

Expedíció service és controller

Configok beállítása

Balance



Frontend

- node_modules
- public
- src
  - components
  - contexts
  - fonts
  - game-helper
  - models
  - pages
  - sounds
  - App.js
  - App.test.js
  - index.css
  - index.js
  - logo.svg
  - reportWebVitals.js
  - setupTests.js
- .gitignore
- package-lock.json
- package.json
- README.md

- contexts
  - GameFieldContext.js
  - HudContext.js
  - IslandContext.js
  - NotificationContext.js
  - UserContext.js
- fonts
- game-helper
- models
- pages
  - battle
  - email-verification
  - error
  - expedition
  - island
  - login
  - management
  - market
  - myprofile
  - notifications
  - pwreset
  - register
  - select-island
  - sell
  - single-notification
  - tutorial

React JS
Bootstrap
Rxjs
Axios
Moment
React-cookie
React-simple-animate
React-hook-form
Use-sound

- components
  - alert-modal
  - base-modal
  - expedition-interface
  - expedition-result-modal
  - experience-bar
  - float-alert
  - game-field
  - hud
  - islands-battle
  - layout
  - loading-screen
  - management-abilities
  - management-island
  - management-unbuilt-buildings
  - profile-image
  - resend-verify-email-form
  - unbuilt-building

Norbert Frontend

Sziget és sziget választás

Épületek

Csata

Menedzselés

Login

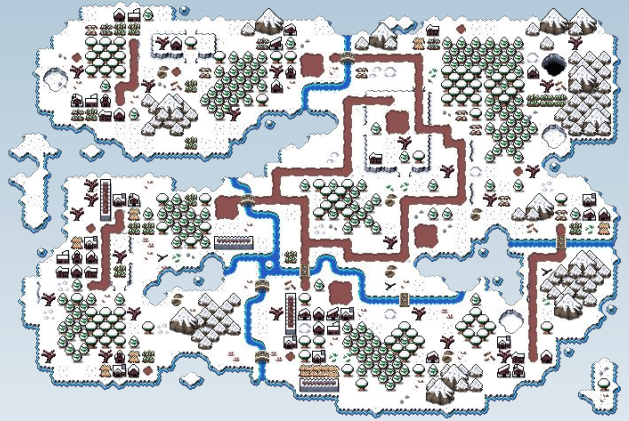
Email megerősítés

Gamefield

Értesítések

Animációk

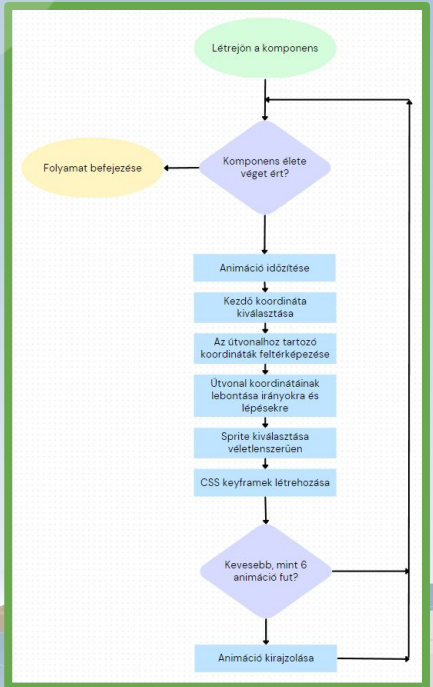
RXJS



Norbert Frontend

- Értesítések:
  - websocket csatlakozás
  - az új értesítések azonnal megjelennek, ikon is jelzi a hudban
  - részletes nézet
- Gamefield:
  - Nagyítás, mozgatás, érintés eventek
  - animációk renderelése
  - objektumok megjelenítése rácsszerkezetben

Norbert Frontend



- RXJS (javascript reaktív programozási könyvtár):
  - épület termelés, eventek, reagálás http kérésekre
- Animációk

```
const urls = [
  `${process.env.REACT_APP_API_BASE}/api/Island/GetIsland`,
  `${process.env.REACT_APP_API_BASE}/api/Building/GetAllBuildings`,
  `${process.env.REACT_APP_API_BASE}/api/Building/GetAllUnbuiltBuildings`,
];

const requests$ = urls.map((url) => from(axios.get(url)));

forkJoin(requests$).subscribe({
  next: (response) => {
    const island = response[0].data;
    const buildings = response[1].data;
    const unbuiltBuildings = response[2].data;

    this.setState((state) => ({
      ...state,
      isIslandInitialized: true,
      buildings: buildings,
      unbuiltBuildings: unbuiltBuildings,
      island: {
        spritePath: island.spritePath,
        buildableLocations: island.buildableLocations,
        npcRoutes: island.npcRoutes,
        npcSprites: island.npcSprites,
      },
    }));
  },
  error: () => {
    alert("Nem sikerült kapcsolódni a szerverhez!");
  },
});
```

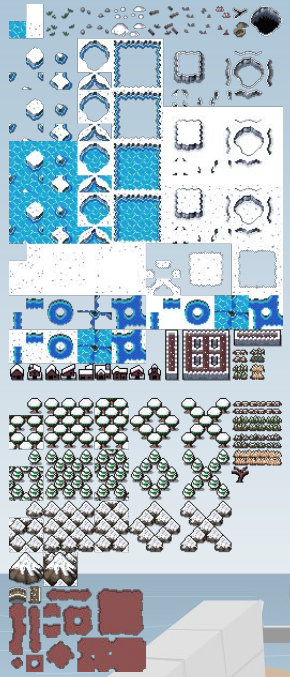
```
this.touchMove$ = new Subject();
this.touchStart$ = new Subject().pipe(
  switchMap(() =>
    this.touchMove$.pipe(
      pairwise(),
      map(([previousEvent, currentEvent]) => {
        const previousTouch =
          previousEvent.touches[0] || previousEvent.changedTouches[0];
        const currentTouch =
          currentEvent.touches[0] || currentEvent.changedTouches[0];
        const movementX = currentTouch.clientX - previousTouch.clientX;
        const movementY = currentTouch.clientY - previousTouch.clientY;

        return [movementX, movementY];
      })
    )
  )
);
```

Edit Frontend

Saját profil
Útmutató
Expedíció

```
  setConfirmPassword(event.target.value);
}

function submitHandler(event) {
  event.preventDefault();

  if (password !== confirmPassword) {
    setErrorMessage("Nem egyeznek a megadott jelszavak!");
    return;
  }

  axios
    .put("https://localhost:7276/api/Auth/ResetPassword", {
      password: password,
      confirmPassword: confirmPassword,
    })
    .then(() => {
      setUserLoggedOut();
      navigate("");
    })
    .catch(() => {
      setErrorMessage("Nem sikerült kapcsolódni a szerverhez.");
    });
}

useEffect(() => {
  setIsHudDisplayed(true);
}, []);

return (
  <Layout navigations={[]} title="Saját profil">
    <div className="myprofile-data flex-column">
      <div className="myprofile d-flex justify-content-center">
        <ProfileImage />
      </div>
      <div className="Profile d-flex justify-content-center">
        <form id="myprofile-form" className="row" onSubmit={submitHandler}>
          <div class="User justify-content-center">
            <h2>{user.username}</h2>
            <h2>{user.email}</h2>
            <div class="reset-passwordjustify-content-center bg-bleur">
              <h3 class="text-profile text-center">Jelszó módosítása:</h3>
              <label class="text-label col-sm-12 text-center">
                Új jelszó:
```

Balázs Frontend



UI / UX

Regisztráció

Jelszó emlékeztető

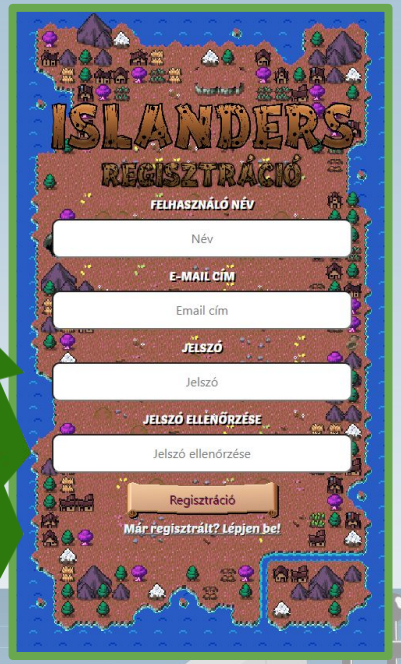
Belépés

Piac

Értesítések

Zene / hangok

Balázs Frontend



Regisztráció
React-hook-form
Hibakezelés

```
<div className="justify-content-center form-group row pb-1">
  <label className="col-form-label text-center register-label">
    JELSZÓ
  </label>
  <input
    className="register-input"
    name="password"
    type="password"
    placeholder="Jelszó"
    id="password"
    {...register("password", {
      required: true,
      max: 40,
      min: 8,
      maxLength: 40,
      pattern:
        /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,40}$/i,
    })}
  />
  {errors.password?.type === "pattern" && (
    <span className="reg-error-msg text-center">
      A jelszónak 8 - 40 karakter hosszúnak kell lennie.
    </span>
  )}
  {errors.password?.type === "pattern" && (
    <span className="reg-error-msg text-center">
      A jelszó tartalmazzon egy kisbetűt, egy nagybetűt és egy
      számot.
    </span>
```

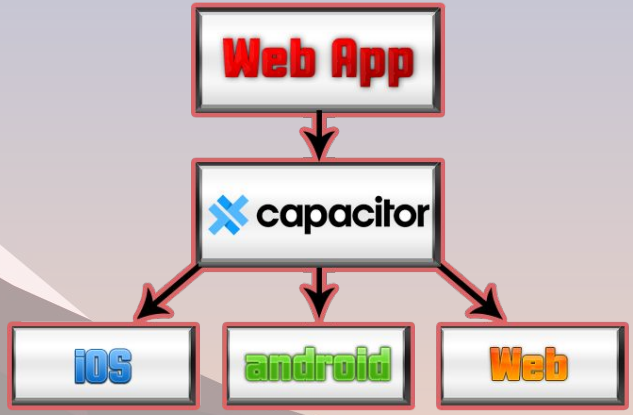
Balázs Frontend

- Piac
  - Hirdetések megjelenítése
  - Saját hirdetések kezelése
  - Hirdetés feladása
- Zene / hangok
  - Hang lejátszás
  - Zene lejátszás

```
import useSound from 'use-sound';

import boopSfx from '../../sounds/boop.mp3';

const BoopButton = () => {
  const [play] = useSound(boopSfx);

  return <button onClick={play}>Boop!</button>;
};
```

Mobil App

Capacitor
Android studio
Reszponzív dizájn

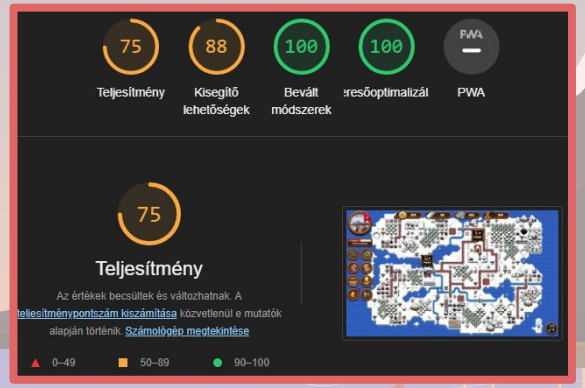


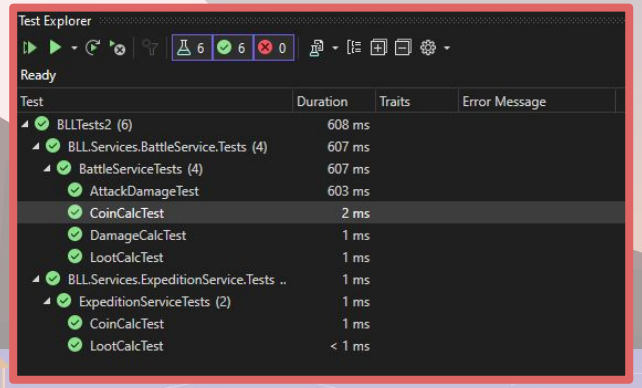
Tesztelés

Csata és expedíció függvényei

Frontend tesztelése

Frontend teszt (Lighthouse)

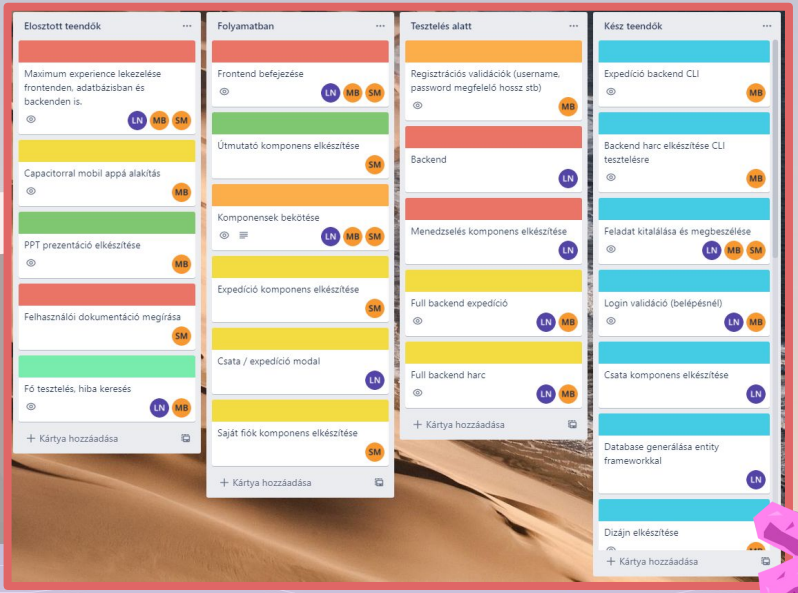




Csapatmunka

Github (2 repository)
Trello
Discord
Messenger
Google Drive

public	touch island moving	2 days ago
src	UnbuiltBuilding javítás	12 hours ago
.gitignore	Initialize project using Create React App	4 months ago
README.md	Initialize project using Create React App	4 months ago
package-lock.json	touch island moving	2 days ago
package.json	touch island moving	2 days ago



Publikáció

SharkASP.net
Firebase
Google Drive

Jövőbeli fejlesztési lehetőségek

- Mikrotranzakciók
  - Csata / expedíció gyorsítás
  - Profilkép változtatás
- Rendes mobil app készítése
  - React native-ra váltás
  - Külön mobilapp kotlin / java nyelvre átírás
- Új feature-k bevezetése
  - Több épület lerakása
  - Saját sziget építése alapanyaggal
  - Összefogás más szigetekkel (Guildek) és egymás szigeteinek látogatása
  - Hosszú expedíciók (Pl. barlangon haladva 4 kis ellenfél és egy fő boss)

ISLANDERS

Köszönjük a figyelmet!