

# Atlas Mystère

PROJET NEXT.JS

Jérémy DUFLOT

Sélène BOULLÉE

## Description du projet

### Partie 1

#### Authentification complète

(Inscription, connexion et déconnexion)

→ Base de données, Next Auth

#### Page home

Page principale permettant de jouer aux jeux

→ Leaflet, base de données et GeoJson

#### Page Score

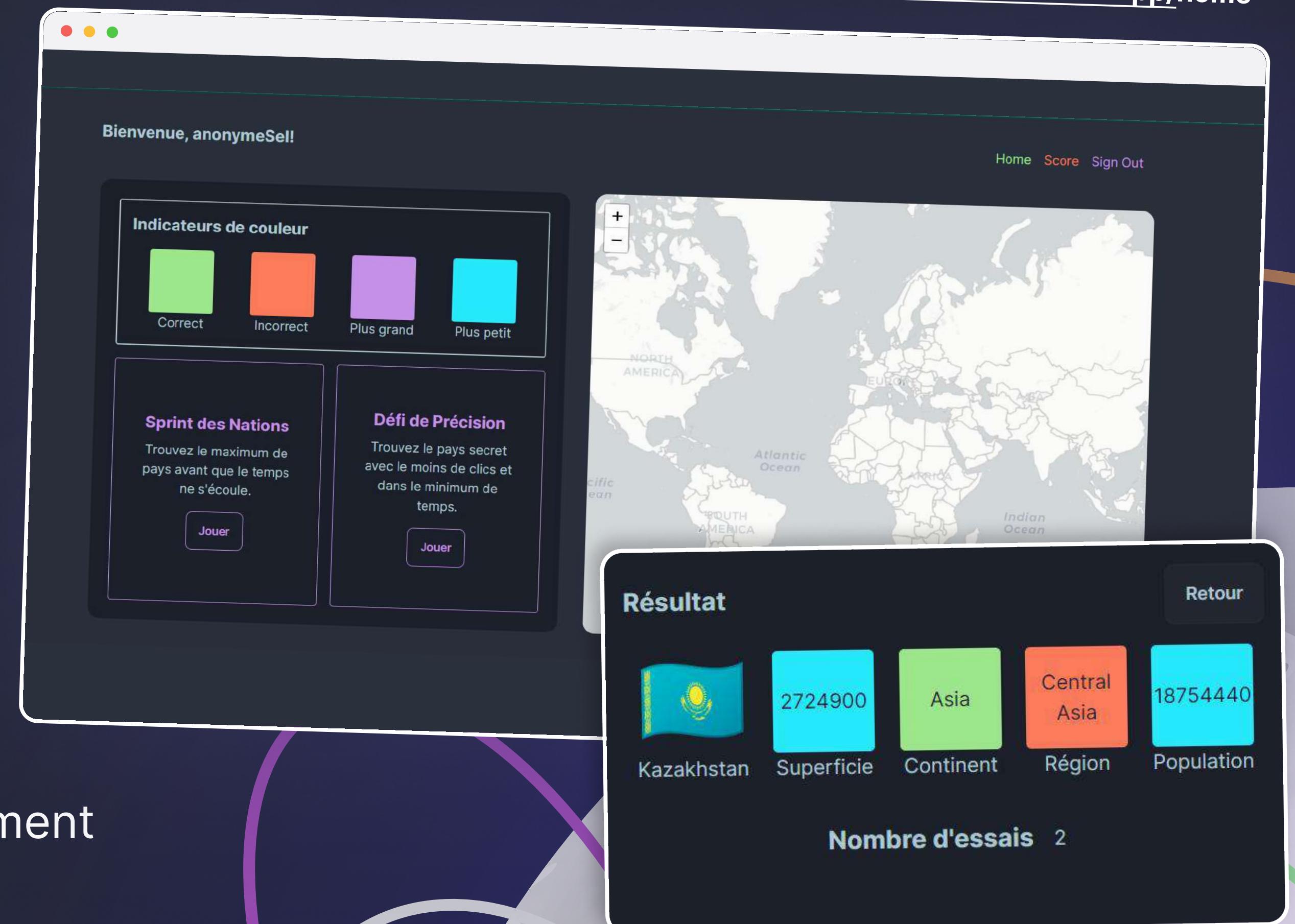
Page où s'affiche les scores sous forme de classement

Permet de voir le score des autres joueurs

→ Base de données

Voir la démo

<https://album-travel.vercel.app/home>



## Description du projet

### Partie 2

#### Sprint des nations

Course contre la montre où les joueurs doivent trouver autant de pays que possible en 90 secondes. Trouver un pays correct ajoute 30 secondes, mais se tromper en retire 5s.

#### Défi de précision

Être le plus précis possible pour trouver le pays en le moins de click possible

#### Sprint des Nations

Trouvez le maximum de pays avant que le temps ne s'écoule.

Jouer

#### Défi de Précision

Trouvez le pays secret avec le moins de clics et dans le minimum de temps.

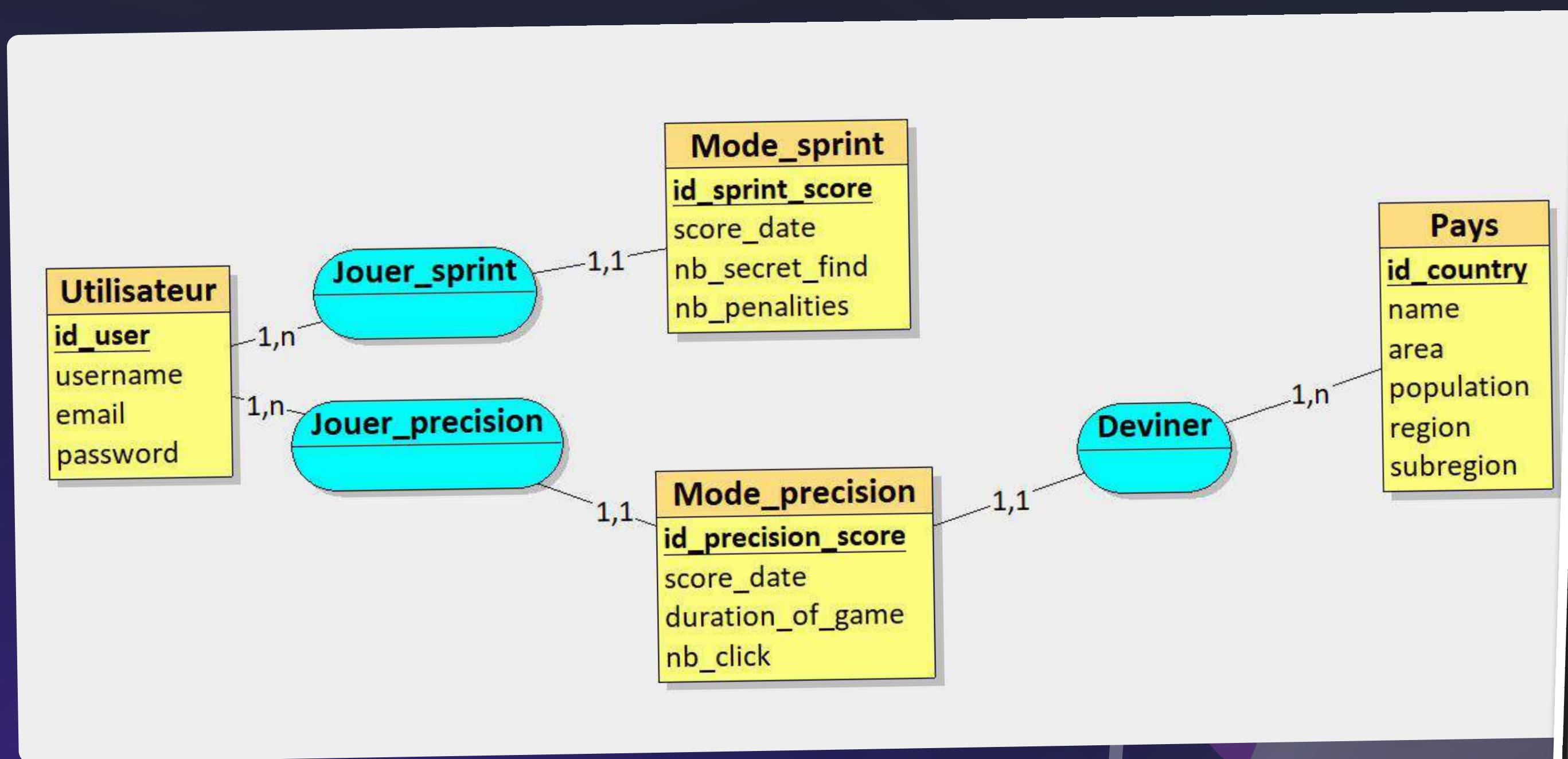
Jouer

Voir la démo

<https://album-travel.vercel.app/home>

## Base de donnée et API

Modèle conceptuel de données (MCD)



Utilisation de l'API countryapi.io pour alimenter les pays dans la BDD

```
async function seedCountries(client) {
  try {
    if (!process.env.NEXT_PUBLIC_API_KEY) {
      console.error('API key is not defined');
      throw new Error('API key is not defined');
    }

    console.log('Using API key:', process.env.NEXT_PUBLIC_API_KEY);

    // Récupère tous les pays depuis l'API
    const response = await fetch(`https://countryapi.io/api/all?apikey=${process.env.NEXT_PUBLIC_API_KEY}`);
    const data = await response.json();

    console.log('API response:', JSON.stringify(data, null, 2));

    const countries = Object.values(data);

    if (!Array.isArray(countries) || countries.length === 0) [
      console.error('Expected an array of countries, got:', countries);
      throw new TypeError('Expected an array of countries');
    ]

    // Insère les pays dans la base de données
    const insertedCountries = await Promise.all(
      countries.map((country) =>
        client.sql`INSERT INTO countries (name, area, population, region, subregion, flag_link)
        VALUES (
          ${country.name},
          ${parseFloat(country.area)},
          ${country.population},
          ${country.region},
          ${country.subregion},
          ${country.flag.large}
        )
        ON CONFLICT (name) DO NOTHING;
      )
    );
  } catch (error) {
    console.error('Error seeding countries:', error);
  }
}
```

Voir la démo

<https://album-travel.vercel.app/home>



# Fonctionnalités

Récupérer les données de la table Countries à partir du nom du pays

```
export async function fetchCountry(name: string) {
  noStore();
  try {
    const data = await sql`SELECT countries.name, countries.population, countries.subregion, countries.region, countries.area, countries.flag_link
      FROM countries WHERE countries.name=${name}`;
    return data.rows as Countries[];
  } catch (error) {
    console.error('Database Error:', error);
    throw new Error('Failed to fetch countries.');
  }
}
```

Récupérer les données des modes de jeu pour réaliser des classements

```
export async function fetchSprintsScore() {
  noStore();
  try {
    const data = await sql`SELECT s.*, u.username
      FROM sprints s
      JOIN users u ON s.id_user = u.id_user
      ORDER BY s.nb_secret_find DESC, s.nb_penalities ASC LIMIT 50`;
    console.log(data);
    return data.rows as Sprints[];
  } catch (error) {
    console.error('Database Error:', error);
    throw new Error('Failed to fetch sprints.');
  }
}
```

```
export async function fetchPrecisionScore() {
  noStore();
  try {
    const data = await sql`SELECT p.*, u.username
      FROM precisions p
      JOIN users u ON p.id_user = u.id_user
      ORDER BY nb_click ASC, duration_of_game ASC LIMIT 50`;
    return data.rows as Precisions[];
  } catch (error) {
    console.error('Database Error:', error);
    throw new Error('Failed to fetch precision.');
  }
}
```

```
if (mapRef.current) {
  L.geoJSON(geojsonData, {
    style: function (feature) {
      return { color: "#cccccc", weight: 1, fillColor: "#ffffff", fillOpacity: 0.4 };
    },
    onEachFeature: function (feature, layer: GeoJSON) {
      // Click on a country
      layer.on('click', async function () {
        if (!activePlay) return;

        console.log("clicked on", feature.properties.name);

        if (lastCountryClickedRef.current !== feature.properties.name) {
          lastCountryClickedRef.current = feature.properties.name;

          // test si victoire
          if (feature.properties.name !== secretCountry?.name) [
            layer.setStyle({ fillColor: "#FF7D5C", fillOpacity: 0.6 });
          ] else {
            layer.setStyle({ fillColor: "#9FE88D", fillOpacity: 0.4 });
          }

          try {
            let data = await fetchCountry(feature.properties.name);
            setCountryClicked(data[0]);
            console.log("Country data:", data[0]);
          } catch (error) {
            console.error("Error fetching country data:", error);
          }
        }
      });
    }
  });
}
```

Voir la démo

<https://album-travel.vercel.app/home>

## Solutions techniques proposées

Voir la démo



<https://album-travel.vercel.app/home>

### Partie 1

#### API REST



##### API RestCountries :

- Problème rencontré : Certificat SSL non renouvelé, API rendue payante sous un autre domaine.

##### Gestion des requêtes :

- Limite de 250 requêtes par inscription.
- Optimisation : Vérification du dernier pays cliqué pour éviter des requêtes inutiles.

##### Sécurité :

- Stockage de l'API\_KEY dans un fichier .env.local pour sécurité.

##### Optimisation :

- Décision d'organiser les données dans une table pour réduire les requêtes à long terme.

#### Conflit nom des pays

##### Limite dans les pays pris en compte par l'api GeoJson

Tri les pays de moins de 500 000 habitants

##### Noms différents entre les données GeoJson et la base de donnée

Création d'un script pour renommer les données en bdd

```
async function updateCountryNames(client) {
  try {
    const updateQueries = [
      client.sql`UPDATE countries SET name = 'Swaziland' WHERE name = 'Eswatini'`,
      client.sql`UPDATE countries SET name = 'United States of America' WHERE name = 'United states'`,
      client.sql`UPDATE countries SET name = 'Czech Republic' WHERE name = 'Czechia'`,
      client.sql`UPDATE countries SET name = 'Macedonia' WHERE name = 'North Macedonia'`,
      client.sql`UPDATE countries SET name = 'Democratic Republic of the Congo' WHERE name = 'DR Congo'`,
      client.sql`UPDATE countries SET name = 'The Bahamas' WHERE name = 'Bahamas'`,
      client.sql`UPDATE countries SET name = 'United Republic of Tanzania' WHERE name = 'Tanzania'`,
      client.sql`UPDATE countries SET name = 'Guinea Bissau' WHERE name = 'Guinea-Bissau'`,
      client.sql`UPDATE countries SET name = 'East Timor' WHERE name = 'Timor-Leste'`,
      client.sql`UPDATE countries SET name = 'Republic of Serbia' WHERE name = 'Serbia'`,
    ];
  }
}
```

## Solutions techniques proposées

### Partie 2

Voir la démo



<https://album-travel.vercel.app/home>

#### Authentification

##### Nomenclature des dossiers :

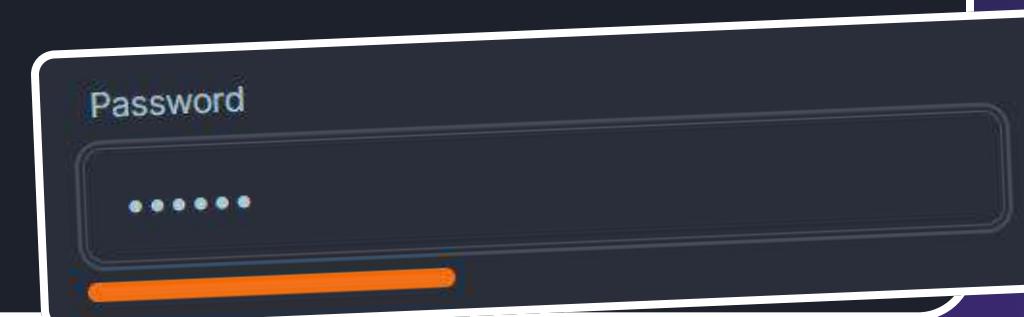
- Dossier src, mauvaise redirection côté serveur  
→ Migration des dossier en dehors de src

##### Disfonctionnement de session avec next-auth :

- Problème de hook useSession() avec cette version de next.js.  
→ création d'une session manuellement

##### Sécurisation du mot de passe :

- Installation du module zxcvbn
- Ajout d' obligation d' au moins un caractère spéciale et une majuscule



#### Doublons useEffect

Comportements inattendus de useEffect causés par le mode strict de React 18.0

→ Condition d'initialisation pour empêcher l'exécution en boucle.

#### Déploiement Vercel

- Soucis de caractères spéciaux comme l'apostrophe
- Conflit avec les hooks, modules
- Pages vides

```
<p className="text-center">Trouvez le maximum de pays avant que le temps ne s'écoule.</p>
```

# Atlas Mystère

## Et si nous avions plus de temps ....

### Indice image supplémentaire pour le défi de précision

Dans le défi de précision, si l'utilisateur dépasse 1min30s alors une image du pays s'affiche en haut

### Mode Histoire ou Géographie

Les indices pourraient selon le mode soit avoir rapport avec l'histoire du style

- Date de fondation (plus ou moins )
- Pays colonisateur
- Régime politique actuel
- Religion de la majorité

### OAuth

Pour la connexion, l'utilisateur à l'option d'inscription avec google ou autres providers (github, facebook..)

### Responsive

Adapter le design pour pouvoir jouer sur téléphone



Voir la démo

<https://album-travel.vercel.app/home>



# Merci

Pour votre écoute

Jérémy DUFLOT

Sélène BOULLÉE