

React and the Art of Gamification

Courtney Yatteau



START



Gamifying React Education

App.js

Download

Reset

Full Screen

```
1 import { useState } from 'react';
2
3 function Square({ value, onSquareClick }) {
4   return (
5     <button className="square" onClick={onSquareClick}>
6       {value}
7     </button>
8   );
9 }
10
11 function Board({ xIsNext, squares, onPlay }) {
12   function handleClick(i) {
13     if (calculateWinner(squares) || squares[i]) {
14       return;
15     }
16     const nextSquares = squares.slice();
17     if (xIsNext) {
18       nextSquares[i] = 'X';
```

Next player: X

1. Go to game start



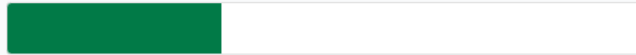
evidation

Hi Courtney!

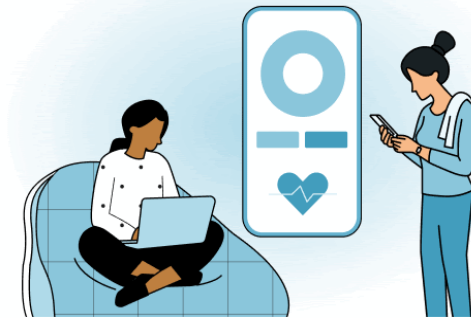
Keep up the good work and boost your rewards!

Your Week on Evidation

3,367 POINTS



Congrats, you earned 198 points this week! You're 34% towards \$10.

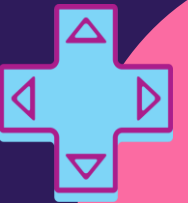


Did you know there's an app available?

+10 pts

Discover the Evidation app - enhance your experience today!

[Learn more](#)



⌕ Search (text or /regex/)

Context.Provider

t

V injectIntl +1

Context.Consumer

V Connect

Context.Provider

V

U

StarBalance Connect +1

Context.Provider

Context.Consumer

```
formatRelativeTime: f bound h() {}
formatTime: f bound m() {}
formatTimeToParts: f bound b() {}
formats: {number: {...}}
  number: {money: {...}, moneyRounded: {...}}
    money: {currency: "USD", style: "curre...
    moneyRounded: {currency: "USD", maximu...
formatters: {getDateTimeFormat: f bound o(...
  locale: "en-US"
messages: {accountCards.addCard.cardTermsN...
  onError: f onError() {}
  onWarn: f onWarn() {}
  textComponent: f Gn() {}
  timeZone: undefined
  wrapRichTextChunksInFragment: undefined
```

rendered by
react-dom@18.3.1

Starbucks

Summer on, Courtney ☀️

366★

STAR BALANCE

25

100

200

300

400

Context.Consumer 308px x 164px 1s

Rewards you can get with Stars

Answer a few questions to find something new

What type of drink are you looking for?

Iced

Cool off and uplift

Hot

Warm up and get going

Scan in store

Start an order

⌕ Search (text or /regex/)

key= ".0"

Anonymous key=" ".1" ForwardRef

ru key=" ".2"

styled.div

t

Context.Consumer

Context.Consumer

Context.Consumer

```
props
  children: f bound () {}

context
  value: {brand: {...}, breakpoint: {...}, breakpoint...
    brand: {primary: "#0f5fbc"}
    breakpoint: {large: 1280, medium: 1070, small...
    breakpoints: ["32em", "48em", "64em", "78em"]
    card: {backgroundColor: "#fff", borderRadius:...
    color: {error: "#eb3f54"}
    dropdown: {activeColor: "#f8f8f8", background...
    font: {primarySans: "'PublicSans', sans-serif...
    gridSize: "60rem"
    input: {borderColor: "#e3e3e3", color: "#2a3f...
    label: {color: "#5b6b7d"}
    palette: {announcement: "#0F5FBC", base: {...},...
    select: {borderColor: "#e3e3e3", color: "#2a3...
    space: [0, "0.25rem", "0.5rem", "1rem", "1.5r...
```

rendered by
react-dom@17.0.2

evidation

My Points: 336 ▾

336

10,000 ★

Context.Consumer 291px x 36px

Great job keeping your account active! Keep completing cards so your points never expire.

Learn More

Health, powered by you

Day Aug 8

0

Week Aug 4 - Aug 8

172

Month Aug 2024

220

Biking

Exerc...

Food ...

Healt...

Cards




Runni...

Refer friends to earn more points!

REFER

Walked 6,829 steps Yesterday • Fitbit +8

Tracked heart rate Yesterday • Fitbit +8

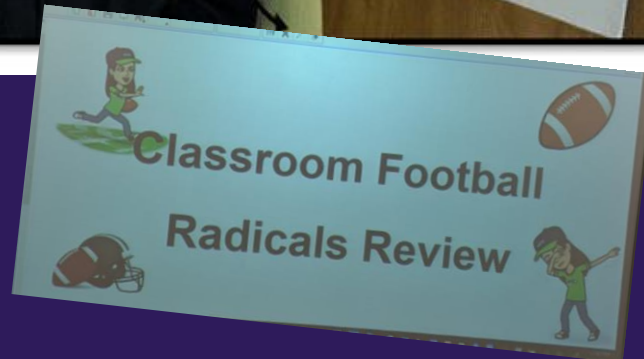


**Effective gamification creates
an ecosystem where users are
constantly motivated to
engage, improve, and achieve.
It's about harnessing the
power of game mechanics to
drive real-world results.**

- Sebastian Deterding

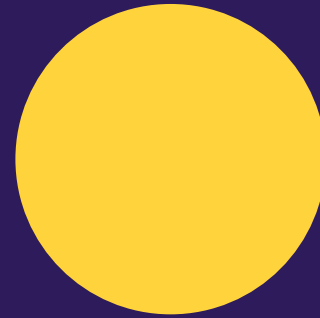
Courtney Yatteau

Developer Advocate, Esri





G.A.M.E.S.



G

A

M

E

S

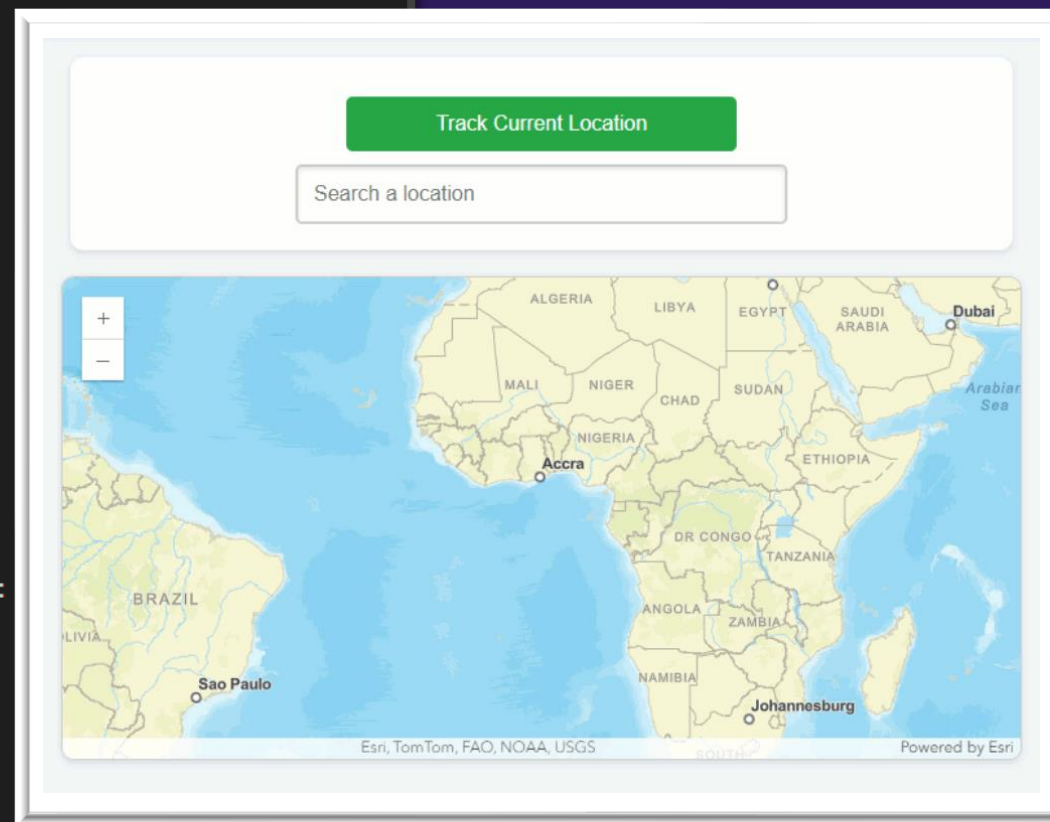
Gamified UI
Components

Gamified UI Components

```
const Badge = ({ badges }) => {
  const [earnedBadges, setEarnedBadges] = useState([]);
  const [selectedBadge, setSelectedBadge] = useState(null);

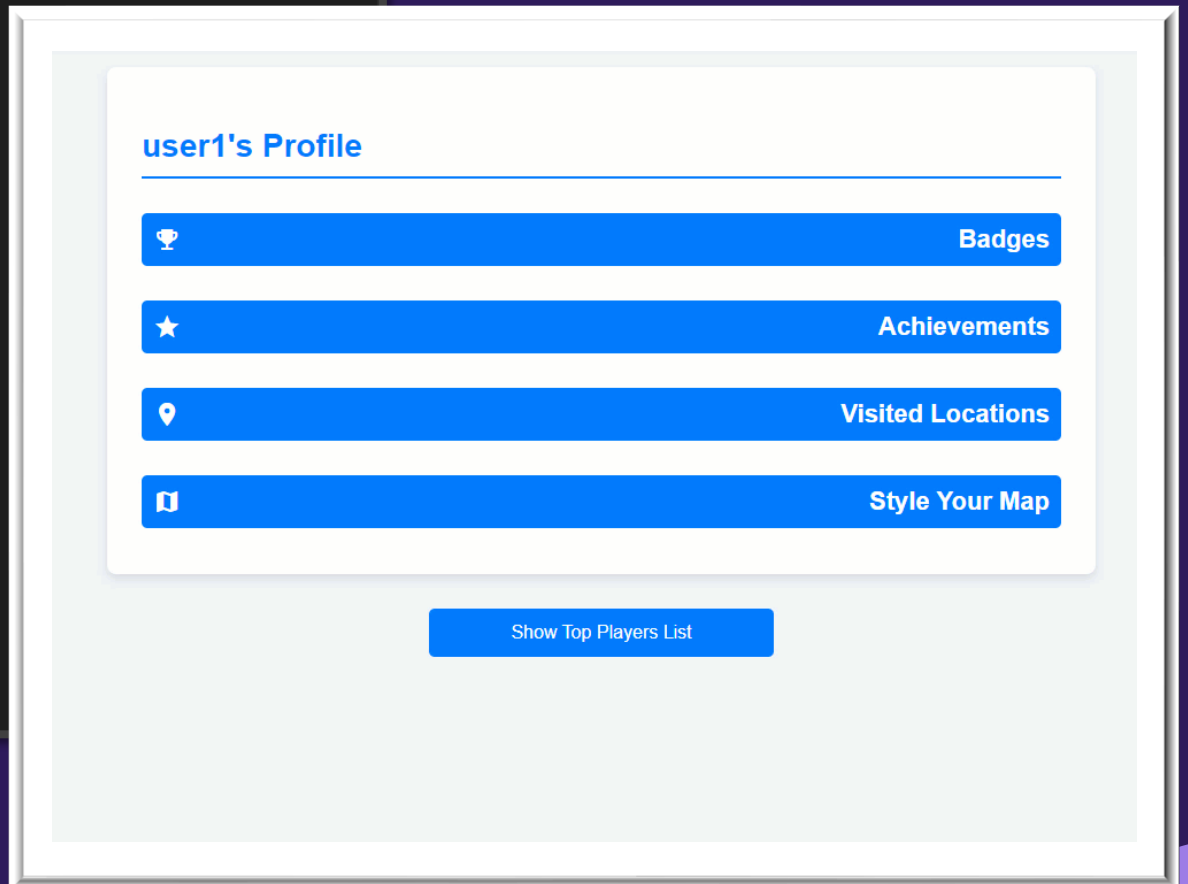
  const handleBadgeClick = (badge) => {
    setSelectedBadge(badge);
    setEarnedBadges((prev) => [...prev, badge]);
  };

  return (
    <div className="badges-container">
      {badges.map((badge, index) => (
        <div
          key={index}
          className={`badge ${earnedBadges.includes(badge) ? "badge-earned" : ""}`}
          style={{ backgroundColor: badge.color }}
          onClick={() => handleBadgeClick(badge)}
        >
          {badge.text}
        </div>
      ))}
      {selectedBadge && <BadgePopup badge={selectedBadge} onClose={() => setSelectedBadge(null)} />}
    </div>
  );
};
```



Gamified UI Components

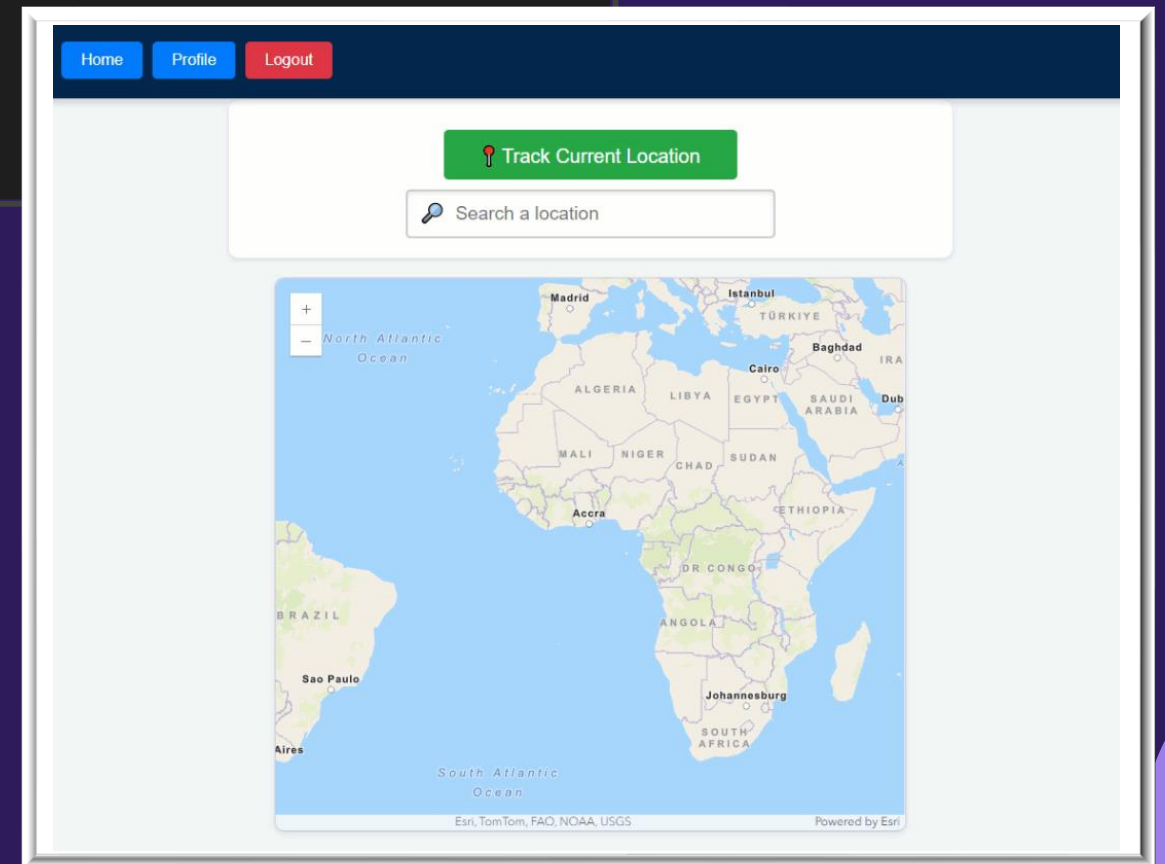
```
const ProgressBar = ({ points, maxPoints }) => {  
  const progress = Math.min((points / maxPoints) * 100, 100);  
  
  return (  
    <div className="progress-bar">  
      <div  
        className="progress-bar-fill"  
        style={{ width: `${progress}%` }}  
      ></div>  
      <span className="progress-bar-text">  
        {points} / {maxPoints} Points  
      </span>  
    </div>  
  );  
};
```



Gamified UI Components

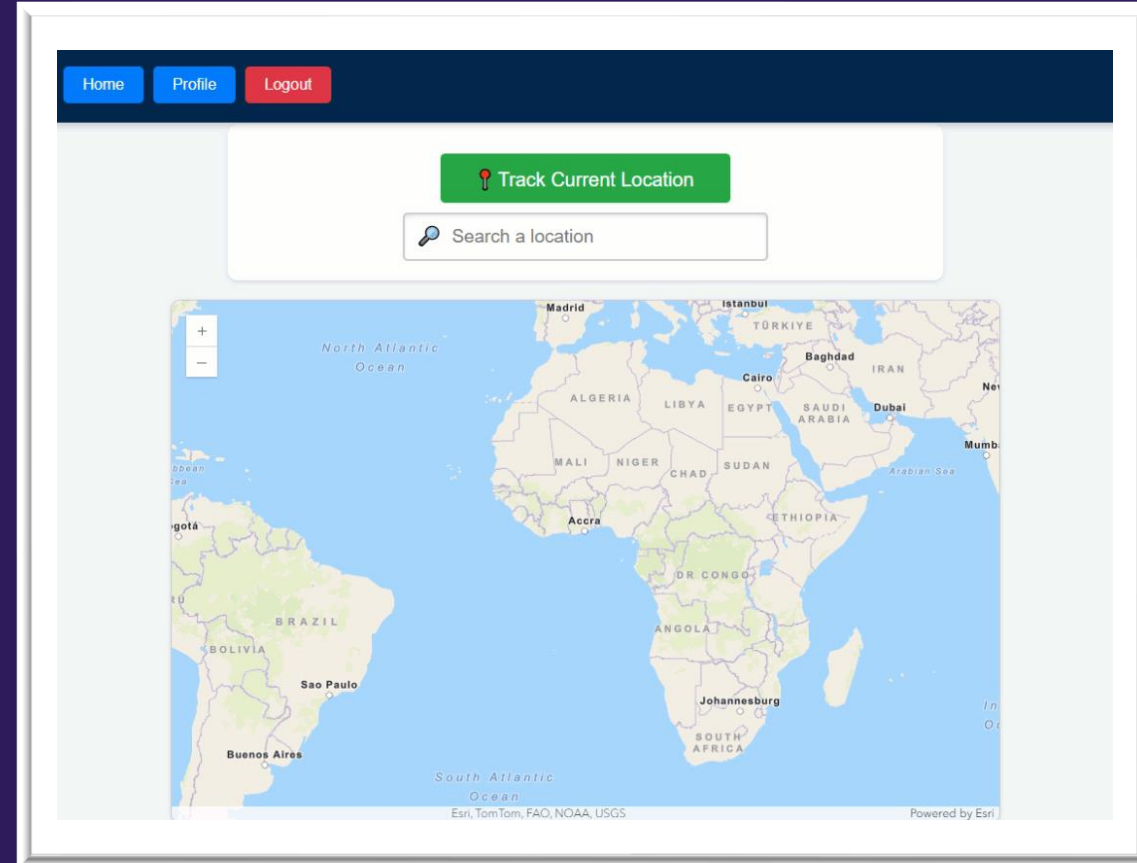
```
const showAchievementPopupMessage = (points, text) => {  
  setAchievementMessage(`Congrats! You've earned ${points} points for ${text}`);  
  setShowAchievementPopup(true);  
  setTimeout(() => {  
    setShowAchievementPopup(false);  
  }, 3000);  
};
```

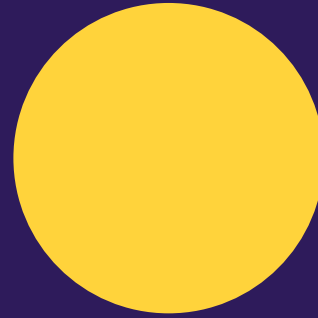
```
{earnedAchievements.map((achievement, index) => (  
  <div key={index} className="achievement-card">  
    <h4>{achievement.text}</h4>  
    <p>{achievement.description}</p>  
    <p>Points: {achievement.points}</p>  
  </div>  
))}
```



Gamified UI Components

```
const basemapOptions = [  
  {  
    id: "11b7300674584eb793129a808290d235",  
    name: "Default Basemap",  
    unlocked: true,  
  },  
  {  
    id: "456d1df3810e482b8abcb2aa0440d6ac",  
    name: "Valentine's Basemap",  
    unlocked: user.visitedLocations.length >= 1,  
  },  
  {  
    id: "f5023edabfee4dd68f2e3f87e2a6c14d",  
    name: "Popcorn Basemap",  
    unlocked: user.achievements.length >= 2,  
  },  
];
```





G

A

M

E

S

Gamified UI
Components

Advanced State
Control

Advanced State Control

```
import { createContext, useReducer, useContext } from "react";

const AppContext = createContext();


> const initialState = { ...
};

const reducer = (state, action) => {
>   switch (action.type) { ...
   }
};

export const AppProvider = ({ children }) => {
  const [state, dispatch] = useReducer(reducer, initialState);

  return (
    <AppContext.Provider value={{ state, dispatch }}>
      {children}
    </AppContext.Provider>
  );
};

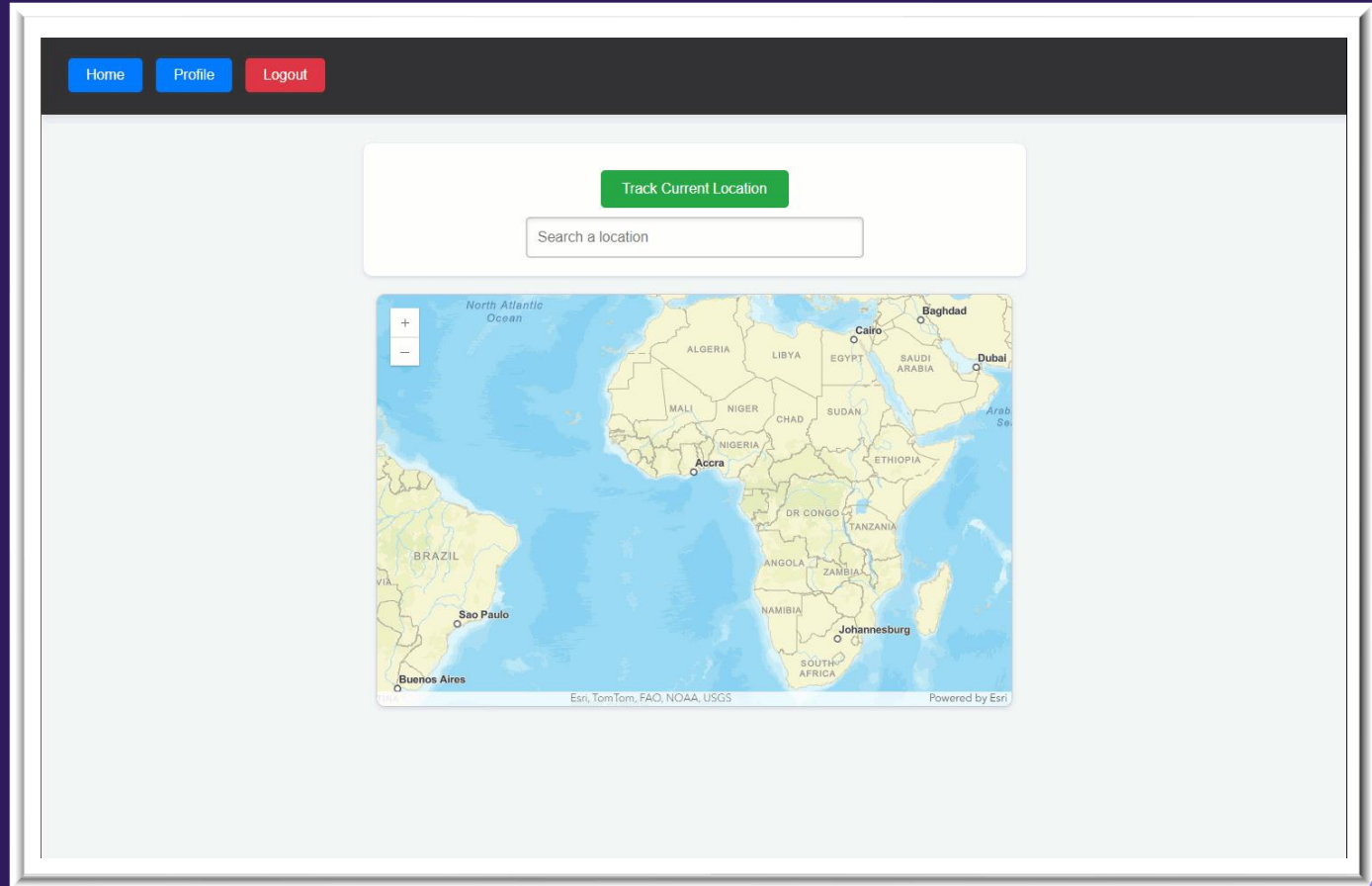
export const useAppContext = () => useContext(AppContext);
```



```
switch (action.type) {
  case "SET_LOCATION":
    return { ...state, location: action.payload };
  case "SET_LOCATION_INPUT":
    return { ...state, locationInput: action.payload };
  case "SET_SUBMITTED":
    return { ...state, submitted: action.payload };
  case "RESET":
    return initialState;
  case "SET_SELECTED_BADGE":
    return { ...state, selectedBadge: action.payload };
  default:
    throw new Error(`Unhandled action type: ${action.type}`);
}
```

Advanced State Control

```
const handleStartQuest = () => {  
  if (questEnabled) {  
    setQuestStarted(true);  
  }  
};  
  
const handleEndQuest = () => {  
  setQuestStarted(false);  
  // Reset other quest-related states  
};  
  
// UI elements controlled by questStarted  
{questStarted && (  
  <button onClick={handleEndQuest}>  
    End Quest  
  </button>  
)}
```





G

Gamified UI
Components



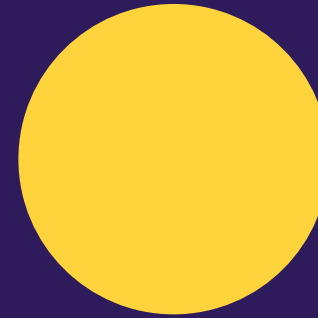
A

Advanced State
Control



M

Memoization
Techniques ... to
evolve into
Modern
Optimization



E

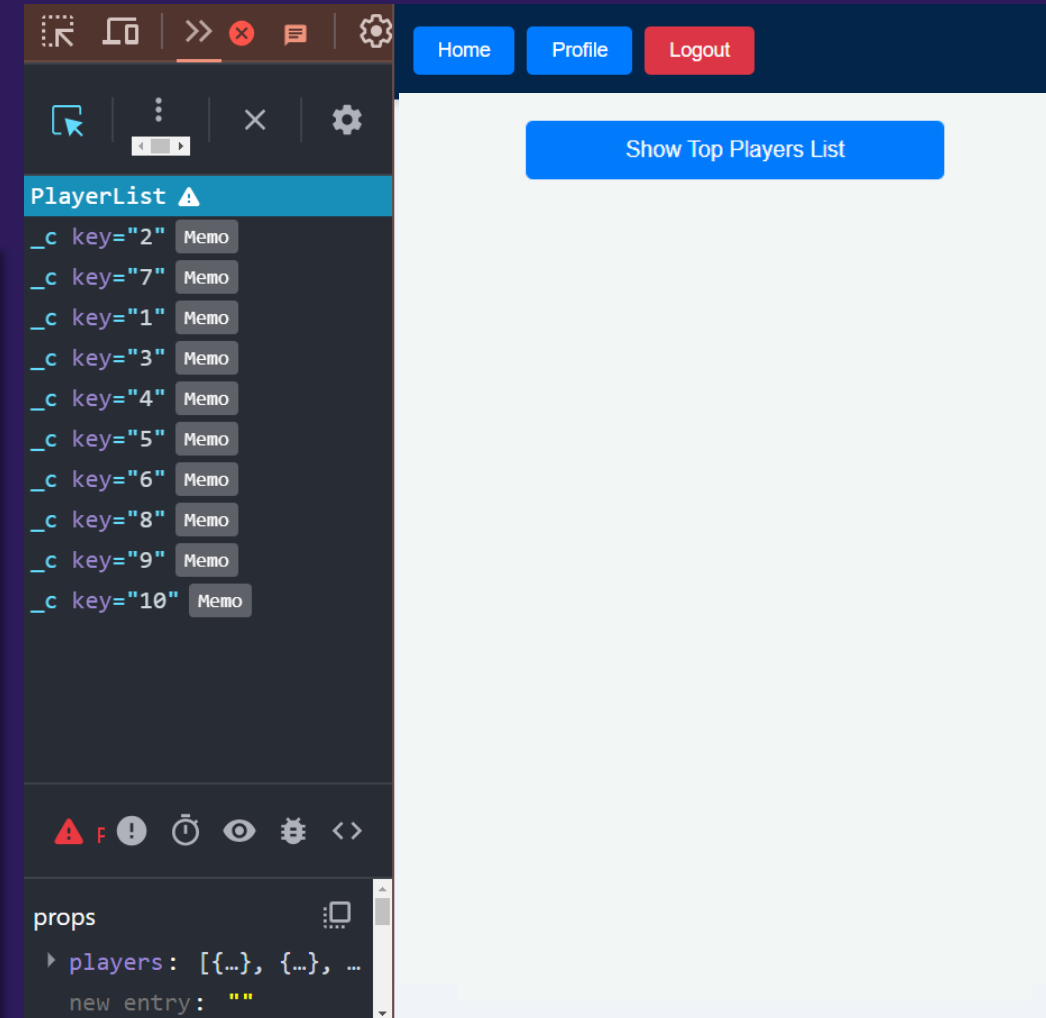


S

Memoization Techniques/Modern Optimization

```
const Player = React.memo(({ player }) => {  
  console.log(`Rendering ${player.name}`);  
  return (  
    <li>  
      {player.name}: {player.score}  
    </li>  
  );  
});  
  
export default Player;
```

```
import { useMemo } from 'react';  
import Player from './Player';  
  
const PlayerList = ({ players }) => {  
  const sortedPlayers = useMemo(() => {  
    return players.sort((a, b) => b.score - a.score);  
  }, [players]);  
  
  return (  
    <div className="player-list">  
      <h3>Top Players</h3>  
      <ul>  
        {sortedPlayers.map(player => (  
          <Player key={player.id} player={player} />  
        ))}  
      </ul>  
    </div>  
  );  
};  
  
export default PlayerList;
```

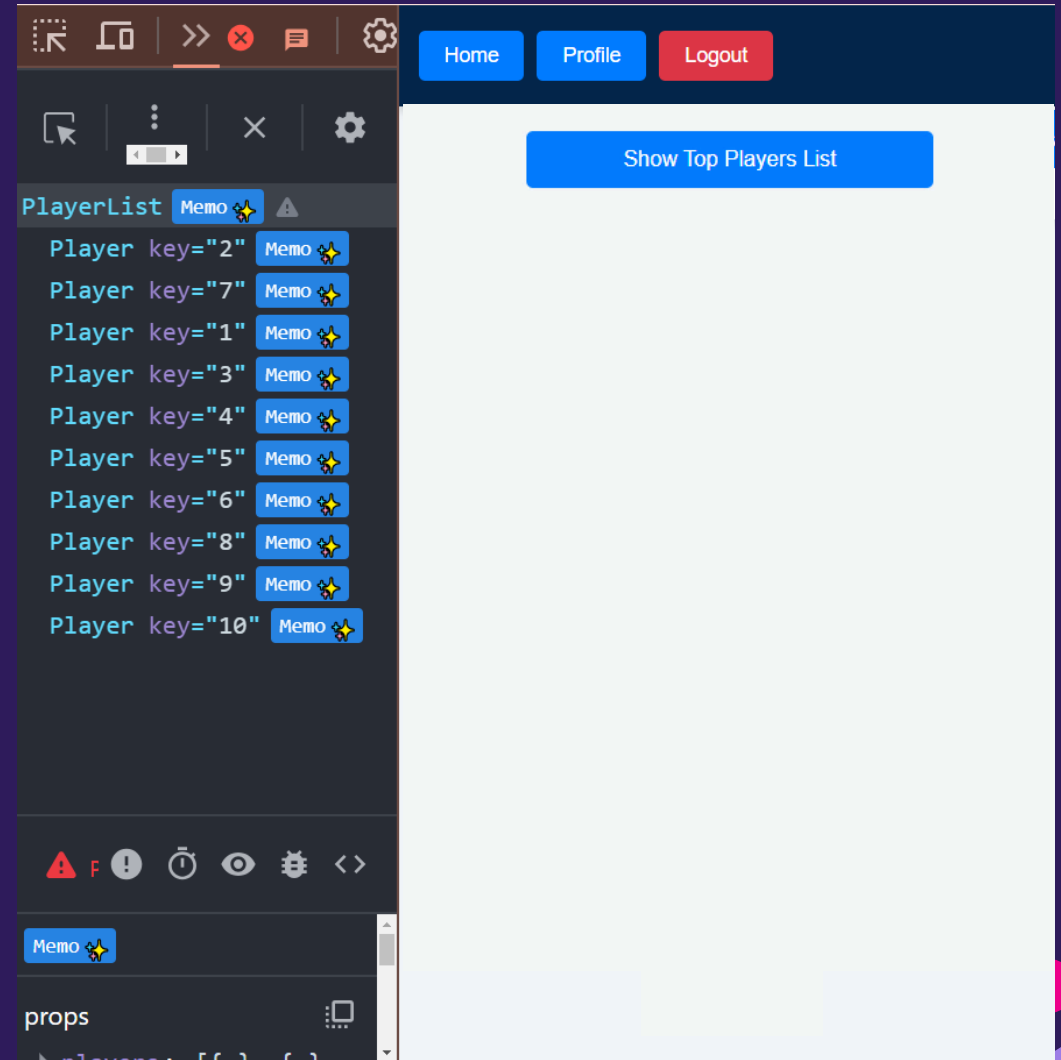


Memoization Techniques/Modern Optimization

```
import useMemo from 'react';
import Player from './Player';

const PlayerList = ({ players }) => {
  const sortedPlayers = useMemo() => {
    return players.sort((a, b) => b.score - a.score);
  }, [players]);
```

```
const Player = React.memo(({ player }) => {
  console.log(`Rendering ${player.name}`);
  return (
    <li>
      {player.name}: {player.score}
    </li>
```





G

Gamified UI
Components



A

Advanced State
Control



M

Memoization
Techniques ... to
evolve into
Modern
Optimization



E

Efficient
Rendering



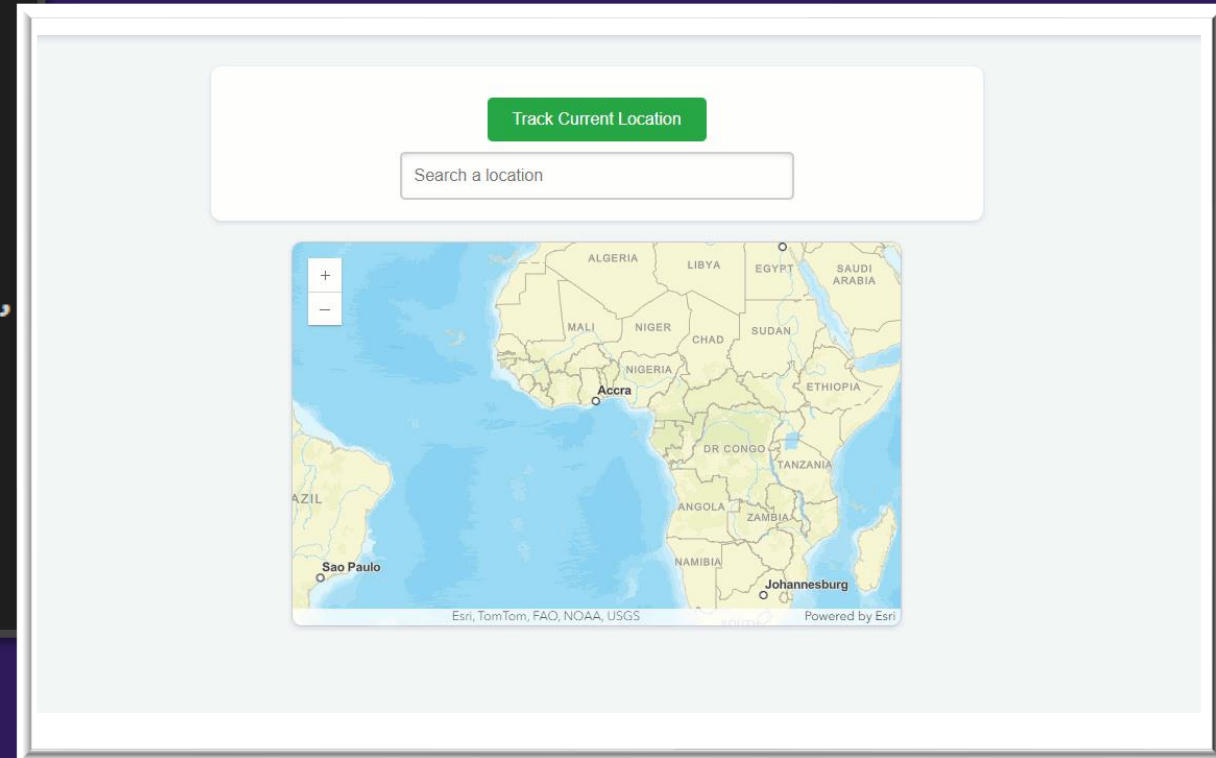
S

Efficient Rendering

```
const MapViewComponent = lazy(() => import("./components/MapViewComponent"));
const SimpleMapComponent = lazy(() => import("./components/SimpleMapComponent"));
const DemographicData = lazy(() => import("./components/DemographicData"));
<Suspense fallback={<div>Loading...</div>}>
  {state.location ? (
    <MapViewComponent location={state.location} landmarks={landmarks} />
  ) : (
    <SimpleMapComponent />
  )}
</Suspense>
```

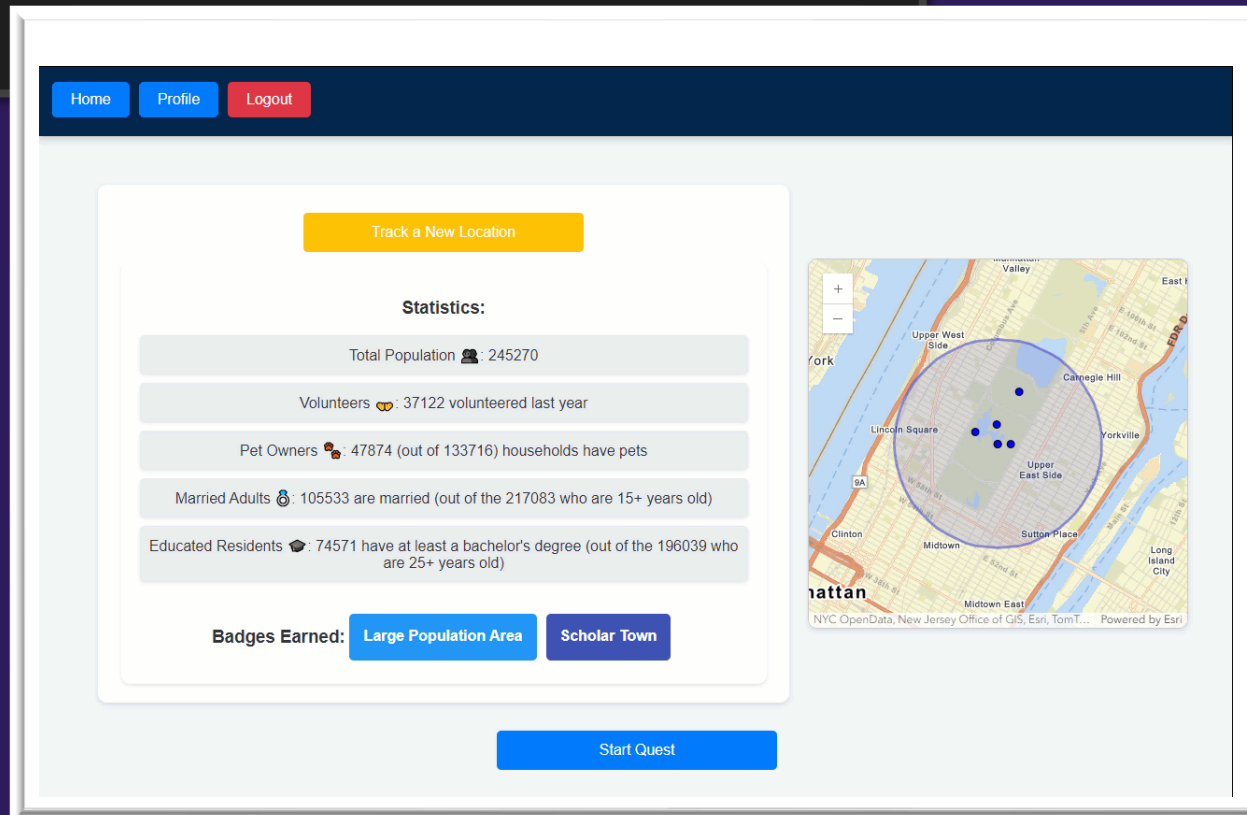
Efficient Rendering

```
useEffect(() => {  
  if (!location || !mapDiv.current) return;  
  
  const map = new Map({ basemap: 'streets' });  
  const view = new MapView({  
    container: mapDiv.current,  
    map: map,  
    center: [location.longitude, location.latitude],  
    zoom: 12,  
  });  
  
  return () => view.destroy();  
}, [location, landmarks]);
```



Efficient Rendering

```
{showLeaderboard && <Leaderboard />}
{selectedBadge && (
  <BadgeModal
    badge={selectedBadge}
    onClose={() => dispatch({ type: 'SET_SELECTED_BADGE', payload: null })}
  />
)}
```





G

Gamified UI
Components



A

Advanced State
Control



M

Memoization
Techniques ... to
evolve into
Modern
Optimization



E

Efficient
Rendering

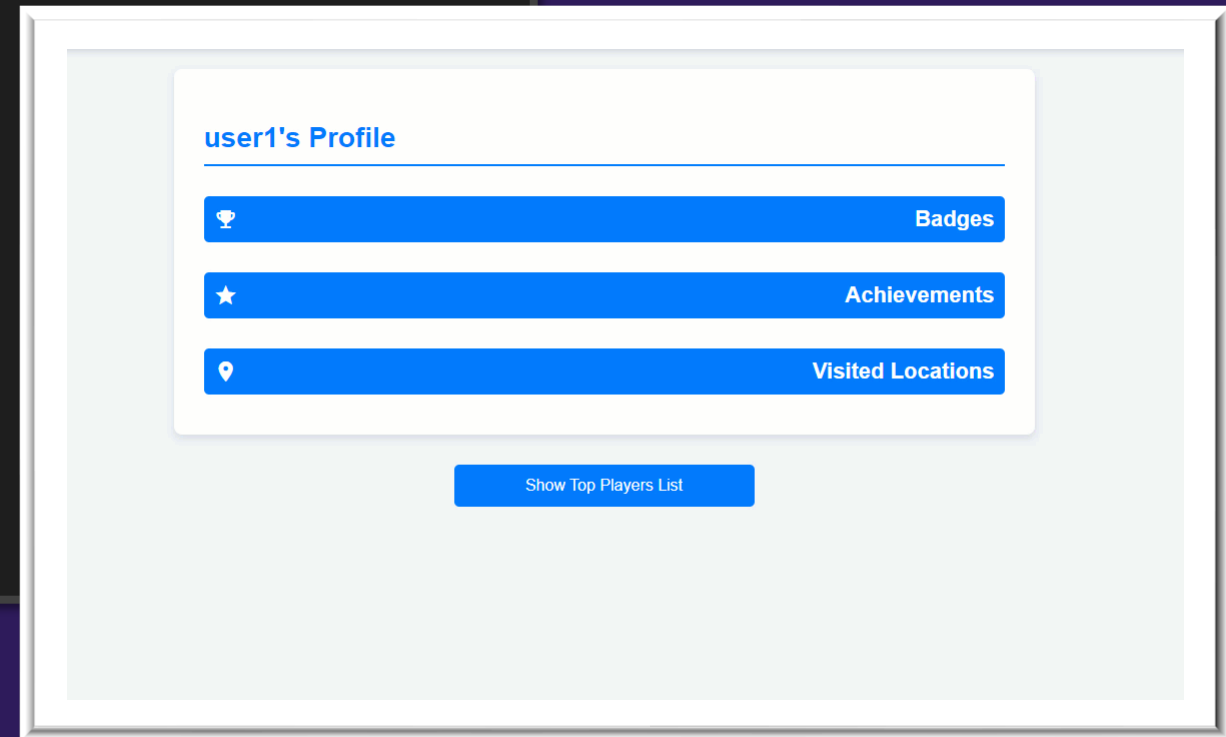


S

Social
Interaction

Social Interaction

```
const ShareButtons = ({ url, message }) => {  
  console.log("ShareButtons rendered with URL:", url, "and message:", message);  
  return (  
    <div className="share-buttons">  
      <TwitterShareButton url={url} title={message}>  
        <TwitterIcon size={32} round />  
      </TwitterShareButton>  
      <FacebookShareButton url={url} quote={message}>  
        <FacebookIcon size={32} round />  
      { /* other share buttons */ }  
    </div>  
  );  
};
```



Social Interaction

```
const PlayerList = ({ players }) => (  
  <div className="player-list">  
    <h3>Top Players</h3>  
    <ul>  
      {players.map(player => (  
        <li key={player.id}>  
          {player.name} - {player.score}  
        </li>  
      ))}  
    </ul>  
  </div>  
);
```

Show Top Players List

G

Streak Counters & XP Bars

A

Adaptive Learning System

M

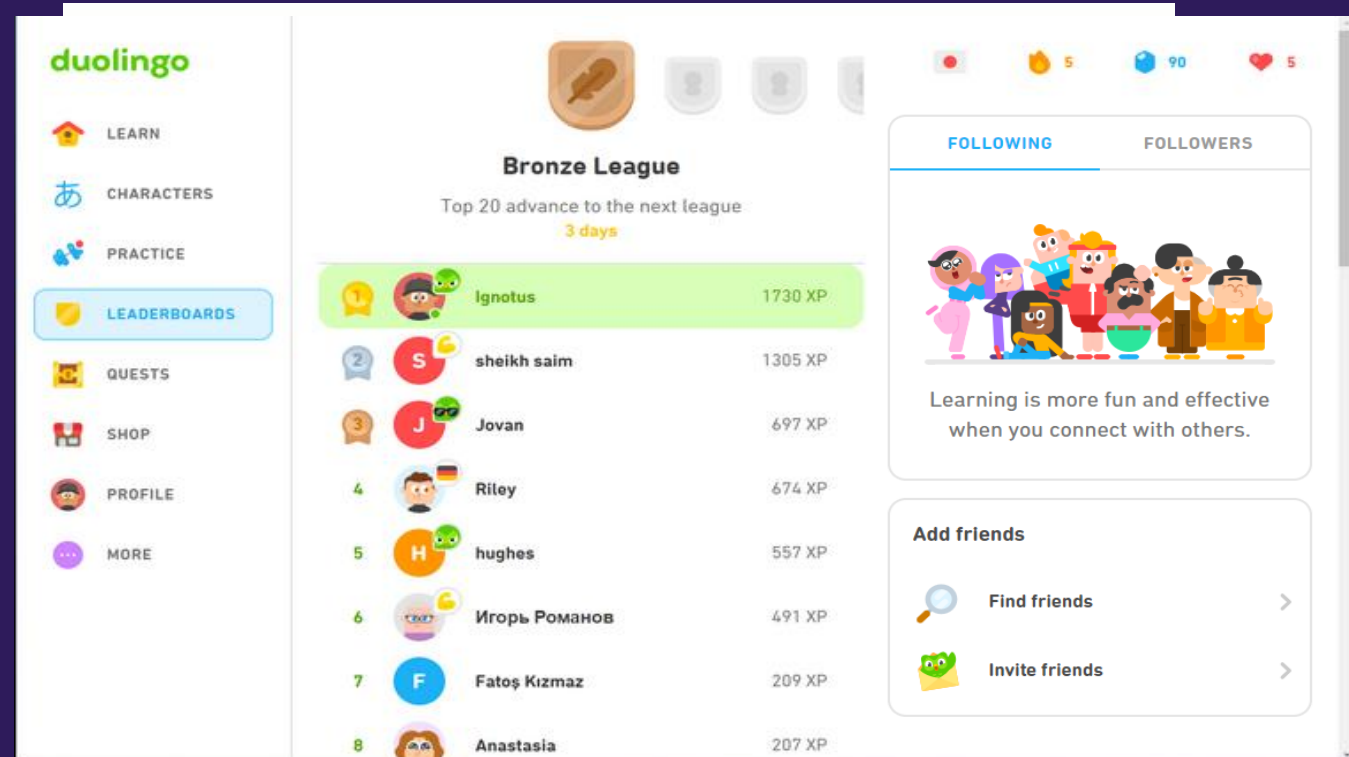
Optimized Lesson Rendering

E

Efficient Component Rendering

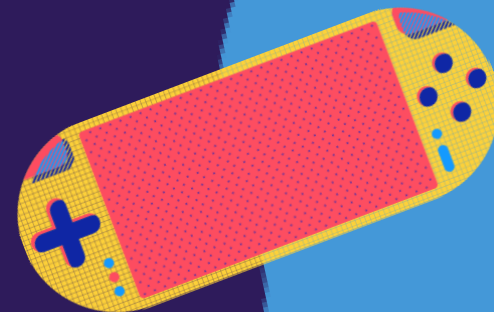
S

Leaderboards & Challenges



duolingo

GAMES



Thank you, React Rally!

Courtney Yatteau



@c_yatteau



courtneyyatteau



bit.ly/React-Rally-2024-Gamification