

src/pages/LeaderboardPage.js

```

1 //The assign tasks are around 110th row!!!!
2 import { useEffect, useState } from 'react';
3 import { supabase } from '../supabaseClient';
4 import { useNavigate } from 'react-router-dom';
5 import React from 'react';
6
7
8 function DateRequestCard({ request, onAccept, onReject }) {
9   const { id, requester, status } = request;
10
11   return (
12     <li
13       style={{
14         display: 'flex',
15         alignItems: 'center',
16         justifyContent: 'space-between',
17         marginBottom: '12px',
18       }}
19     >
20     <div style={{ display: 'flex', alignItems: 'center' }}>
21       {requester?.profile_pic && (
22         <img
23           src={requester.profile_pic}
24           alt={` ${requester.username}'s avatar`}
25           style={{
26             width: '40px',
27             height: '40px',
28             borderRadius: '50%',
29             marginRight: '10px',
30             objectFit: 'cover',
31           }}
32         />
33       )}
34       <span>
35         <strong>{requester?.username || 'Unknown'}</strong> wants to go on a date
36         with you!{' '}
37         {status === 'pending' ? '' : ` (${status})`}
38       </span>
39     </div>
40     {status === 'pending' && (
41       <div>
42         <button onClick={() => onAccept(id, requester?.username)} style={{
43           marginRight: '5px' }}>
44            Accept
45         </button>
46         <button onClick={() => onReject(id)}> Reject</button>
47       </div>
48     )}
49   </li>
50 );
51 }
52
53 export default function LeaderboardPage() {
54   const [users, setUsers] = useState([]);

```

```
53 const [currentUser, setCurrentUser] = useState(null);
54 const [top3UserIds, setTop3UserIds] = useState([]);
55 const [incomingDateRequests, setIncomingDateRequests] = useState([]);
56 const [activeDate, setActiveDate] = useState(null);
57 const [loading, setLoading] = useState(true);
58 const [error, setError] = useState(null);
59 const navigate = useNavigate();
60 const [bannedUserIds, setBannedUserIds] = useState(new Set());
61 const [myTask, setMyTask] = useState('');
62 const [datingTask, setDatingTask] = useState(null);
63 const [activeReportId, setActiveReportId] = useState(null);
64 const [reportReasons, setReportReasons] = useState({});
65 const [askedUserIds, setAskedUserIds] = useState([]);
66
67
68
69 // after fetchCurrentUser, also fetch bans:
70 useEffect(() => {
71   if (currentUser?.id) {
72     fetchBans();
73   }
74 }, [currentUser]);
75
76 // existing ban handler
77 useEffect(() => {
78   fetchCurrentUser();
79 }, []);
80
81 useEffect(() => {
82   if (currentUser?.id) {
83     fetchLeaderboard();
84     fetchIncomingDateRequests();
85     fetchActiveDate();
86     fetchBans(); // also fetch bans here
87   }
88 }, [currentUser]);
89
90 async function assignRandomTaskToCouple(coupleId) {
91   const coupleRes = await supabase
92     .from('dating')
93     .select('user1_id, user2_id')
94     .eq('id', coupleId)
95     .single();
96
97   if (coupleRes.error) {
98     console.error('Error fetching couple:', coupleRes.error);
99     return;
100   }
101
102   const { user1_id, user2_id } = coupleRes.data;
103
104   // Fetch usernames of both users
105   const profileRes = await supabase
106     .from('profiles')
107     .select('id, username')
108     .in('id', [user1_id, user2_id]);
```

```
109
110   if (profileRes.error) {
111     console.error('Error fetching usernames:', profileRes.error);
112     return;
113   }
114
115   const usernames = profileRes.data.map(profile => profile.username);
116
117   // Tasks
118   const tasks = [
119     "Тасралтгүй 2 цаг video call. Ярих хугацаандаа зураг зурах. Тэгээд хамгийн  

120     гуигтай юм уу хамгийн хөгжилтэй уншсан уран зохиолуудаа ярих.",
121     "Эрэгтэй нь эмэгтэйд гэрээс очиж аваад зайсан толгой гараад хот би тэрэнд  

122     хайртай гээд орилох, орилж байгаагаа бичлэг болгоод групп дээр шэйр.",
123     "Хамтдаа кино үзчихээд юм авж идэнгээ хотын гудамжаар алхах.",
124     "Гэрээсээ хоол бэлдэж ирээд хамт уул руу гарж идэнгээ нэгэнийхээ зургийг дарж  

125     өгөх. (гоё хосийн зурагтай болох)",
126     "Өөрсдөө болзоогоо төлөвлөөд Youtube дээр влог болгож оруулах. Тэгээд групп дээр  

127     линкээ өг.",
128     "Хамт кино ямар ч хамаагүй байдлаар үзээд. Киноны хамгийн гоё хэсэг юу байсныг  

129     яриад тэр киногоо үнэлэх. Зураг эдр дарвал групп дээр шэйр",
130     "Нэгэндээ тоглодог 2 online, 1 offline тоглоомоо хуваалц. Тэгээд нэгийн сонгоод  

131     нэгнийхээ рекордыг эвд эсвэл нэгэнтэйгээ хамт тогло.",
132     "Дуртай 3 дуугаа нэгэнтэйгээ хуваалцаад, Нэг дуугаа сонгоод хамт дуулаад,  

133     тэрийгээ групп дээр шэйр.",
134     "30 минутын болзооны сорил. 5 минутанд өдөр тутамд юу хийдгээ, 10 минутанд  

135     хамгийн их мөрөөддөг зүйлээ, 15 минутанд чөлөөт цагаа хэрхэн өнгөрөөдөгөө.",
136     "Нэгэнтэйгээ 1 цаг ярингаа нэгэнийхээ хувцуудыг хараад ямар хувцас өмсөхийн  

137     заах. Тэгээд зурагаа даруулаад групп дээр шэйр."
138   ];
139
140   const randomTask = tasks[Math.floor(Math.random() * tasks.length)];
141
142   // Insert tasks for both users
143   const insertRes = await supabase.from('couple_task').insert([
144     {
145       assigner_id: null,
146       assignee_id: user1_id,
147       task_text: randomTask,
148       assigned_at: new Date(),
149       completed: false,
150       partnerRated: false,
151       partner_rating: null,
152     },
153     {
154       assigner_id: null,
155       assignee_id: user2_id,
156       task_text: randomTask,
157       assigned_at: new Date(),
158       completed: false,
159       partnerRated: false,
160       partner_rating: null,
161     },
162   ]);
163
164   if (insertRes.error) {
165     console.error('Error assigning couple task:', insertRes.error);
166     return;
167   }
```

```

158     }
159
160     // Add public notification
161     const message = `Шинэ хосийн даалгавар ${usernames[0]}, ${usernames[1]} хоёрт
    оногдлоо! 🎉 Даалгавар: "${randomTask}"`;
162
163     const notifRes = await supabase.from('notifications').insert({
164       user_id: null,
165       message,
166     });
167
168     if (notifRes.error) {
169       console.error('Error creating notification:', notifRes.error);
170     } else {
171       console.log(message);
172     }
173   }
174
175
176   const assignTask = async (assigneeId, taskText) => {
177     const { error } = await supabase.from('assigned_tasks').insert({
178       assigner_id: currentUser.id,
179       assignee_id: assigneeId,
180       task_text: taskText,
181     });
182
183     const { data } = await supabase
184       .from('profiles')
185       .select('username')
186       .eq('id', assigneeId)
187       .single();
188
189     if (error) {
190       console.error('Error assigning task:', error);
191       alert('Даалгавар өгөлт амжилтгүй.');
```

192 } else {
 193 // Also send a public notification
 194 await supabase.from('notifications').insert({
 195 user_id: null,
 196 message: `\${data.username}-д даалгавар ирлээ. Даалгавар нь "\${taskText}"`,
 197 });
 198
 199 alert('Даалгавар амжилттай өгөгдлөө!');

```

200     }
201   };
202
203   const handleSubmitReport = async (reportedId) => {
204     const reason = reportReasons[reportedId] || '';
205     if (!reason.trim()) return;
206
207     const { error } = await supabase.from('reports').insert({
208       reporter_id: currentUser.id,
209       reported_id: reportedId,
210       reason: reason.trim(),
211     });
212

```

```
213     if (error) {
214         console.error('Supabase insert error:', error);
215         alert('Failed to report');
216     }
217     else {
218         alert('Reported successfully');
219         setActiveReportId(null);
220         setReportReasons(prev => ({ ...prev, [reportedId]: '' }));
221     }
222     console.log('Current user:', currentUser);
223 };
224
225
226
227 useEffect(() => {
228     async function fetchMyTask() {
229         const { data, error } = await supabase
230             .from('assigned_tasks')
231             .select('task_text')
232             .eq('assignee_id', currentUser.id)
233             .order('assigned_at', { ascending: false })
234             .limit(1)
235             .single();
236
237         if (error && error.code !== 'PGRST116') {
238             console.error('Error fetching task:', error);
239         } else if (data) {
240             setMyTask(data.task_text);
241         }
242     }
243
244     if (currentUser?.id) {
245         fetchMyTask();
246     }
247 }, [currentUser?.id]);
248
249
250
251
252 async function fetchTop3UserIds() {
253     const { data, error } = await supabase
254         .from('profiles')
255         .select('id')
256         .order('christma_points', { ascending: false })
257         .limit(3);
258
259     if (!error && data) {
260         setTop3UserIds(data.map(user => user.id));
261         return data.map(user => user.id);
262     } else {
263         console.error('Failed to fetch top 3 users', error);
264         return [];
265     }
266 }
267
268 useEffect(() => {
```

```
269     fetchTop3UserIds();
270 }, []);
271
272 async function fetchBans() {
273     const top3Ids = await fetchTop3UserIds();
274
275     const { data: bans, error } = await supabase
276         .from('bans')
277         .select('banned_user_id')
278         .eq('banned_by_id', currentUser.id);
279
280     const { data: profiles } = await supabase
281         .from('profiles')
282         .select('is_banned, id')
283         .eq('id', bans.banned_user_id)
284         .eq('is_banned', true)
285
286     if (error) {
287         console.error('Error fetching bans:', error);
288         return;
289     }
290
291     const filteredBans = bans.filter(bans => !top3Ids.includes(bans.banned_user_id));
292
293     setBannedUserIds(new Set(filteredBans.map(b => b.banned_user_id)));
294 }
295
296
297 // ✅ Corrected ban handler with proper notification
298 async function handleBanUser(bannedUserId) {
299     // Fetch top 3 user IDs
300     const top3Ids = await fetchTop3UserIds();
301
302     // Check if currentUser is in top 3
303     const isCurrentUserTop3 = top3Ids.includes(currentUser.id);
304
305     if (isCurrentUserTop3) {
306         // Check if ANY of the top 3 users have already banned someone
307         const { data: existingBans, error } = await supabase
308             .from('bans')
309             .select('*')
310             .in('banned_by_id', top3Ids);
311
312         if (error) {
313             console.error('Error checking existing bans:', error);
314             return;
315         }
316
317         if (existingBans.length >= 1) {
318             alert('Эхний гурав дундаа ганц л хүнийг бандах боломжтой. Энэ хэрэглэгчийг  
бандахын тулд эхлээд бандуулсан байгаа хэрэглэгчийг бангаас нь гаргана уу.');
```

```
324 const dating = await isUserDating(bannedUserId);
325 if (dating) {
326   alert('Та болзож байгаа хэрэглэгчийг бандаж болохгүй.');
```

327 return;

328 }

329

330 // Fetch banned user's username for notification

331 const { data: bannedUserProfile, error: profileError } = await supabase

332 .from('profiles')

333 .select('username')

334 .eq('id', bannedUserId)

335 .single();

336

337 if (profileError) {

338 console.error('Error fetching banned user profile:', profileError);

339 alert('Failed to fetch banned user info.');

340 return;

341 }

342

343 // Proceed to ban user

344 const { error: insertError } = await supabase

345 .from('bans')

346 .insert([{ banned_user_id: bannedUserId, banned_by_id: currentUser.id }]);

347

348 if (insertError) {

349 console.error('Error banning user:', insertError);

350 alert('Failed to ban user.');

351 return;

352 }

353

354 // Insert notification AFTER successful ban

355 await supabase.from('notifications').insert({

356 user_id: null,

357 message: `Эхний гурав \${bannedUserProfile.username} ийг бандлаа! 🚫`,

358 });

359

360 alert('Та энэ хүнийг амжилттай бан длаа.');

361

362 await fetchBans(); // refresh banned list

363 fetchLeaderboard(); // refresh leaderboard if needed

364 }

365

366

367

368 // ✅ Corrected unban handler

369 async function handleUnbanUser(unbanUserId) {

370 await supabase

371 .from('bans')

372 .delete()

373 .eq('banned_user_id', unbanUserId)

374 .eq('banned_by_id', currentUser.id); // FIXED HERE

375

376 // Fetch banned user's username for notification

377 const { data: unBannedUserProfile, error: profileError } = await supabase

378 .from('profiles')

379 .select('username')

```

380     .eq('id', unbanUserId)
381     .single();
382
383
384     // Insert notification AFTER successful ban
385     await supabase.from('notifications').insert({
386       user_id: null, // or the user you want to notify if any
387       message: `${currentUser.username} нь ${unBannedUserProfile.username} ийг бангаас
388       rapралаа! 🟢`,
389     });
390
391     await fetchBans();
392   }
393
394   async function isUserDating(userId) {
395     const { data, error } = await supabase
396       .from('dating')
397       .select('id')
398       .or(`user1_id.eq.${userId},user2_id.eq.${userId}`)
399       .eq('status', 'accepted') // 🟢 Only accepted counts as dating
400       .limit(1)
401       .maybeSingle();
402
403     if (error && error.code !== 'PGRST116') {
404       console.error('Error checking dating status:', error);
405       return false;
406     }
407
408     return !!data; // true if currently dating
409   }
410
411
412
413   async function fetchCurrentUser() {
414     try {
415       setLoading(true);
416       setError(null);
417       const {
418         data: { user },
419       } = await supabase.auth.getUser();
420       if (!user) throw new Error('User not logged in');
421
422       const { data: profile } = await supabase
423         .from('profiles')
424         .select('id, username')
425         .eq('id', user.id)
426         .single();
427
428       setCurrentUser(profile);
429     } catch (err) {
430       setError(err);
431     } finally {
432       setLoading(false);
433     }
434   }

```



```
435
436 async function fetchLeaderboard() {
437   try {
438     setLoading(true);
439     const { data } = await supabase
440       .from('profiles')
441       .select('id, username, nickname, profile_pic, christmas_points, gender')
442       .order('christmas_points', { ascending: false })
443       .limit(500);
444
445     setUsers(data);
446     setTop3UserIds(data.slice(0, 3).map((u) => u.id));
447   } catch (err) {
448     setError(err);
449   } finally {
450     setLoading(false);
451   }
452 }
453
454 async function fetchIncomingDateRequests() {
455   if (!currentUser?.id) return;
456   try {
457     const { data: requests } = await supabase
458       .from('date_requests')
459       .select('id, requester_id, status, created_at')
460       .eq('requested_id', currentUser.id)
461       .order('created_at', { ascending: false });
462
463     const requesterIds = requests.map((r) => r.requester_id);
464     const { data: requestersProfiles } = await supabase
465       .from('profiles')
466       .select('id, username, profile_pic')
467       .in('id', requesterIds);
468
469     const combined = requests.map((r) => ({
470       ...r,
471       requester: requestersProfiles.find((p) => p.id === r.requester_id),
472     }));
473
474     setIncomingDateRequests(combined);
475   } catch (err) {
476     console.error('Error fetching date requests:', err);
477   }
478 }
479
480 async function fetchActiveDate() {
481   if (!currentUser?.id) {
482     setActiveDate(null);
483     return;
484   }
485   try {
486     const { data, error } = await supabase
487       .from('date_requests')
488       .select('*')
489       .or(
```

```
490     `and(requester_id.eq.${currentUser.id},status.eq.accepted),
and(requested_id.eq.${currentUser.id},status.eq.accepted)`
491   )
492   .maybeSingle(); // changed from .single() to .maybeSingle()
493
494   if (error) {
495     console.error('Error fetching active date:', error);
496     setActiveDate(null);
497     return;
498   }
499
500   setActiveDate(data || null);
501 } catch (err) {
502   console.error('Error fetching active date:', err);
503   setActiveDate(null);
504 }
505 }
506
507
508
509 // Assume you have currentUser and activeDate available
510 useEffect(() => {
511   if (!currentUser?.id || !activeDate?.id) {
512     setDatingTask(null);
513     return;
514   }
515
516   async function getPartnerId() {
517     // Fix .or() syntax: wrap conditions in parentheses
518     const { data, error } = await supabase
519       .from('dating')
520       .select('user1_id, user2_id')
521       .or(`user1_id.eq.${currentUser.id},user2_id.eq.${currentUser.id}`)
522       .single();
523
524     if (error && error.code !== 'PGRST116') {
525       console.error('Error fetching dating partner:', error);
526       return null;
527     }
528     if (!data) {
529       console.log('No dating row found');
530       return null;
531     }
532
533     const partnerId = data.user1_id === currentUser.id ? data.user2_id :
data.user1_id;
534     console.log('PartnerId found:', partnerId);
535     return partnerId;
536   }
537
538   async function fetchDatingTask() {
539     const partnerId = await getPartnerId();
540     if (!partnerId) {
541       setDatingTask(null);
542       return;
543     }
544   }
```

```
545     const { data, error } = await supabase
546       .from('couple_task')
547       .select('*')
548       .in('assignee_id', [currentUser.id, partnerId])
549       .order('assigned_at', { ascending: false })
550       .limit(1) // Add limit(1) for safety
551       .single();
552
553     if (error && error.code !== 'PGRST116') {
554       console.error('Error fetching dating task:', error);
555       setDatingTask(null);
556     } else if (data) {
557       console.log('Dating task found:', data);
558       setDatingTask(data.task_text);
559     } else {
560       setDatingTask(null);
561     }
562   }
563
564   fetchDatingTask();
565 }, [currentUser?.id, activeDate?.id]);
566
567
568
569
570
571 async function onLike(likedUserId) {
572   if (!currentUser || currentUser.id === likedUserId) return;
573
574   try {
575     // Step 1: Ban check
576     const { data: bannedUsers, error: bannedUsersError } = await supabase
577       .from('bans')
578       .select('banned_user_id')
579       .in('banned_user_id', [currentUser.id, likedUserId]);
580
581     if (bannedUsersError) throw bannedUsersError;
582
583     if (bannedUsers && bannedUsers.length > 0) {
584       if (bannedUsers.some(b => b.banned_user_id === currentUser.id)) {
585         alert('Та бандуулсан байгаа тул like явуулах боломжгүй.');
```

```
601
602     if (likeCheckError) throw likeCheckError;
603
604     if (existingLikes.length > 0) {
605         alert('Та энэ хүн рүү аль хэдийн like явуулсан байна! Маргааш дахин оролдоно
yy. ');
606         return;
607     }
608
609     // Step 3: Get liked user's profile
610     const { data: likedUser, error: userError } = await supabase
611         .from('profiles')
612         .select('username, christma_points, like_count')
613         .eq('id', likedUserId)
614         .single();
615
616     if (userError) throw userError;
617
618     // Step 4: Insert new like
619     const now = new Date();
620     const { error: insertError } = await supabase
621         .from('likes')
622         .insert({
623             liker_id: currentUser.id,
624             liked_id: likedUserId,
625             created_at: now.toISOString(),
626             like_day: today,
627         });
628
629     if (insertError) throw insertError;
630
631     // Step 5: Update liked user's Christma points and like_count
632     const updatedPoints = likedUser.christma_points + 2;
633     const updatedLikeCount = likedUser.like_count + 1;
634
635     const { error: updateError } = await supabase
636         .from('profiles')
637         .update({
638             christma_points: updatedPoints,
639             like_count: updatedLikeCount,
640         })
641         .eq('id', likedUserId);
642
643     if (updateError) throw updateError;
644
645     // Step 6: Create public notification
646     const { error: notifError } = await supabase.from('notifications').insert({
647         user_id: null,
648         message: `${currentUser.username} нь ${likedUser.username} рүү like явууллаа!
❤️ \`,
649     });
650
651     if (notifError) throw notifError;
652
653     alert('Та энэ хүн рүү like явууллаа! Энэ хүн +2 Christma оноо авлаа. ');
654     fetchLeaderboard();
655
```

```

656   } catch (err) {
657     console.error('Error liking user:', err);
658     alert('Like явуулахад асуудал гарсан байна. Дахин оролдоно уу.');
```

659 }

660 }

661

662

```

663   async function onAskDate(requestedId) {
664     if (!currentUser || currentUser.id === requestedId) return;
665
666     if (activeDate) {
667       alert('Та өөр хүнтэй болзож байгаа тул тэр болзоогоо дуусгачихаад шинийг
эхлүүлнэ үү.');
```

668 return;

669 }

670

```

671   try {
672     // 🚫 Ban check
673     const { data: bannedUsers, error: bannedUsersError } = await supabase
674       .from('bans')
675       .select('banned_user_id')
676       .in('banned_user_id', [currentUser.id, requestedId]);
677
678     if (bannedUsersError) throw bannedUsersError;
679
680     if (bannedUsers && bannedUsers.length > 0) {
681       if (bannedUsers.some(b => b.banned_user_id === currentUser.id)) {
682         alert('Та бан дуулсан байгаа тул болзооны санал явуулах хориотой.');
```

683 } else {

684 alert('Таны болзмоор байгаа хүн бан дуулсан байгаа тул банг аас гарахын
хүлээнэ үү.');

685 }

686 return;

687 }

688

```

689     // ❤️ Limit to one request per day
690     const { data: todayRequestsCount, error: todayError } = await supabase
691       .rpc('requests_sent_today', { user_id: currentUser.id });
692
693     if (todayError) throw todayError;
694
695     if (todayRequestsCount > 0) {
696       alert('Та өдөрт ганц л болзооны санал явуулах эрхтэй.');
```

697 return;

698 }

699

```

700     // 👤 Get both profiles
701     const { data: currentProfile, error: currentProfileError } = await supabase
702       .from('profiles')
703       .select('christma_points, gender, username')
704       .eq('id', currentUser.id)
705       .single();
706
707     if (currentProfileError) throw currentProfileError;
708
709     const { data: requestedProfile, error: requestedProfileError } = await supabase
710       .from('profiles')
```

```

711     .select('username, christma_points, gender')
712     .eq('id', requestedId)
713     .single();
714
715     if (requestedProfileError) throw requestedProfileError;
716
717     // ❌ Gender rule
718     if (currentProfile.gender === requestedProfile.gender) {
719         alert('Зөвхөн эсрэг хүйстэн рүүгээ болзооны санал явуулна уу.');
```

720 return;

721 }

722

723 // ❌ Christma points rule

724 if (currentProfile.christma_points < requestedProfile.christma_points) {

725 alert('Та зөвхөн өөрөөсөө БАГА оноотой хүн рүү болзооны санал явуулах эрхтэй')

726 ;

727 return;

728 }

729 // ✅ No duplicate request

730 const { data: existingRequest, error: existingRequestError } = await supabase

731 .from('date_requests')

732 .select('*')

733 .or(

734 `and(requester_id.eq.\${currentUser.id},requested_id.eq.\${requestedId}),`
 and(requester_id.eq.\${requestedId},requested_id.eq.\${currentUser.id})

735)

736 .maybeSingle();

737

738 if (existingRequestError) throw existingRequestError;

739

740 if (existingRequest?.status === 'pending') {

741 alert('Та аль хэдийн болзооны санал явуулсан байна.');

742 return;

743 }

744

745 if (existingRequest?.status === 'accepted') {

746 alert('Та энэ хүнтэй аль хэдийн болзож байна.');

747 return;

748 }

749

750 // 📁 Insert new request

751 await supabase

752 .from('date_requests')

753 .insert([{ requester_id: currentUser.id, requested_id: requestedId, status: 'pending' }]);

754

755 // 🎁 Update Christma points

756 const isRequesterTop3 = top3UserIds.includes(currentUser.id);

757 const pointsToAdd = isRequesterTop3 ? 10 : 5;

758 const updatedPoints = requestedProfile.christma_points + pointsToAdd;

759

760 await supabase

761 .from('profiles')

762 .update({ christma_points: updatedPoints })

763 .eq('id', requestedId);

764

```

765 // 📢 Notification
766 await supabase.from('notifications').insert({
767   user_id: null,
768   message: `${currentProfile.username} нь ${requestedProfile.username} рүү
Болзооны санал явуулаа! ❤️`,
769 });
770
771 alert(`Болзооны саналыг явуулсан! ${requestedProfile.username} нь
+${pointsToAdd} Christma оноо авлаа.`);
772
773 // ✅ Hide button for this user
774 setAskedUserIds(prev => [...prev, requestedId]);
775
776 // 🔄 Refresh UI
777 fetchLeaderboard();
778 fetchIncomingDateRequests();
779
780 } catch (err) {
781   console.error('Error sending date request:', err);
782   alert('Болзооны саналыг илгээх үед ямар нэгэн зүйл буруу боллоо.');
```

```

783 }
784 }
785
786
787 async function handleAccept(requestId, requesterId, requesterUsername) {
788   if (activeDate) {
789     alert('Та яг одоо болж байна. Энэ болзоогоо дуусгаад дараагийнхийг эхлүүлнэ
үү');
790     return;
791   }
792
793   try {
794     // 1. Mark the request as accepted
795     const { error: acceptError } = await supabase
796       .from('date_requests')
797       .update({ status: 'accepted' })
798       .eq('id', requestId);
799
800     if (acceptError) throw acceptError;
801
802     // 2. Fetch the date request to get requester_id
803     const { data: dateData, error: dateError } = await supabase
804       .from('date_requests')
805       .select('*')
806       .eq('id', requestId)
807       .single();
808
809     if (dateError) throw dateError;
810
811     // 3. Fetch date_count and christma_points for both users
812     const { data: profilesData, error: fetchError } = await supabase
813       .from('profiles')
814       .select('id, date_count, christma_points')
815       .in('id', [currentUser.id, dateData.requester_id]);
816
817     if (fetchError) throw fetchError;
818
```

```
819 const currentUserProfile = profilesData.find(p => p.id === currentUser.id);
820 const requesterProfile = profilesData.find(p => p.id === dateData.requester_id);
821
822 // Increment their date counts
823 const newCurrentUserCount = (currentUserProfile?.date_count || 0) + 1;
824 const newRequesterCount = (requesterProfile?.date_count || 0) + 1;
825
826 // Fetch the requester's username if not passed in
827 const { data: requesterProfileData, error: usernameError } = await supabase
828   .from('profiles')
829   .select('username')
830   .eq('id', dateData.requester_id)
831   .single();
832
833 if (usernameError) throw usernameError;
834
835 const requesterUsername = requesterProfileData.username;
836
837 // 4. Update currentUser date_count
838 const { error: updateCurrentUserError } = await supabase
839   .from('profiles')
840   .update({ date_count: newCurrentUserCount })
841   .eq('id', currentUser.id);
842
843 if (updateCurrentUserError) throw updateCurrentUserError;
844
845 // 5. Update requester date_count
846 const { error: updateRequesterError } = await supabase
847   .from('profiles')
848   .update({ date_count: newRequesterCount })
849   .eq('id', dateData.requester_id);
850
851 if (updateRequesterError) throw updateRequesterError;
852
853 console.log('✅ date_count incremented: currentUser=${newCurrentUserCount},
854 requester=${newRequesterCount}');
855
856 // 6. Insert into dating table
857 const { data: couple, error: datingError } = await supabase
858   .from('dating')
859   .insert([ { user1_id: dateData.requester_id, user2_id: currentUser.id } ])
860   .select()
861   .single();
862
863 if (datingError) throw datingError;
864
865 console.log('✅ Couple created with id:', couple.id);
866
867 // 6.5 Give 5 Christma points to requester
868 const updatedRequesterPoints = (requesterProfile?.christma_points || 0) + 5;
869
870 await supabase
871   .from('profiles')
872   .update({ christma_points: updatedRequesterPoints })
873   .eq('id', dateData.requester_id);
```



```
// 7. Optional notification (kept as-is)
await supabase.from('notifications').insert({
  user_id: null,
  message: `${currentUser.username} нь ${requesterUsername} ий болзооны саналыг зөвшөөрлөө! 💖`,
});

// 8. Assign a random task to the new couple
await assignRandomTaskToCouple(couple.id);

// 9. Refresh UI state
alert('Та болзооны саналыг хүлээж авлаа!');
fetchActiveDate();
fetchIncomingDateRequests();
fetchLeaderboard();

} catch (err) {
  console.error('Error accepting date request:', err);
  alert('Болзооны саналыг зөвшөөрөх үед ямар нэгэн юм буруу боллоо.');
```

```
}
}
```

```
async function handleReject(requestId) {
  try {
    const { data: requester } = await supabase
      .from('date_requests')
      .select('requester_id')
      .eq('id', requestId)
      .single();

    await supabase
      .from('date_requests')
      .update({ status: 'rejected' })
      .eq('id', requestId);

    const { data: name } = await supabase
      .from('profiles')
      .select('username')
      .eq('id', requester.requester_id)
      .single();

    await supabase.from('notifications').insert({
      user_id: null,
      message: `${currentUser.username} rejected ${name.username}'s date request.`,
    });

    alert('You rejected the date request.');
```

```
    fetchIncomingDateRequests();
  } catch (err) {
    console.error('Error rejecting request:', err);
  }
}
```

```
929 // ONLY ADDED THIS FUNCTION
930 async function endDate() {
931   if (!activeDate) return;
932
933   try {
934     // Fetch the date request record
935     const { data: dateRequest, error } = await supabase
936       .from('date_requests')
937       .select('id, created_at, requester_id, requested_id')
938       .eq('id', activeDate.id)
939       .single();
940
941     if (error) {
942       console.error('Error fetching date request:', error);
943       return;
944     }
945
946     const createdAtUTC = new Date(dateRequest.created_at).getTime(); // Supabase
timestamp (UTC)
947     const nowUTC = new Date().getTime(); // Current time (UTC)
948
949     const hoursDiff = (nowUTC - createdAtUTC) / (1000 * 60 * 60);
950     console.log('Created At:', new Date(createdAtUTC).toISOString());
951     console.log('Now:', new Date(nowUTC).toISOString());
952     console.log('Hour Difference:', hoursDiff);
953
954     if (hoursDiff < 24) {
955       const remainingMs = (24 - hoursDiff) * 60 * 60 * 1000;
956       const remainingHours = Math.floor(remainingMs / (1000 * 60 * 60));
957       const remainingMinutes = Math.floor((remainingMs % (1000 * 60 * 60)) / (1000 *
60));
958       alert(`Та багадаа 24 цаг болзох ёстой. ${remainingHours} цаг
${remainingMinutes} минутын дараа дахин үзнэ үү.`);
959       return;
960     }
961
962     // End the date
963     await supabase
964       .from('date_requests')
965       .update({ status: 'ended' })
966       .eq('id', activeDate.id);
967
968     // Get both usernames
969     const { data: requesterProfile } = await supabase
970       .from('profiles')
971       .select('username')
972       .eq('id', dateRequest.requester_id)
973       .single();
974
975     const { data: requestedProfile } = await supabase
976       .from('profiles')
977       .select('username')
978       .eq('id', dateRequest.requested_id)
979       .single();
980
981     // Notify everyone
982     await supabase.from('notifications').insert({
```

```

983     user_id: null,
984     message: `❤️ ${requesterProfile.username} ба ${requestedProfile.username}
болзоогоо дуусгалаа.` ,
985   });
986
987   alert('Та болзоогоо дуусгалаа.');
```

✔

```

988
989   setActiveDate(null);
990   fetchLeaderboard();
991   fetchIncomingDateRequests();
992 } catch (err) {
993   console.error('Error ending date:', err);
994 }
995 }
996
997 // define this inside the component
998 const handleViewProfile = (userId) => {
999   navigate(`/profile-view/${userId}`);
1000 }
1001
1002 function UserCard({
1003   user, rank, onLike, onAskDate, onViewProfile,
1004   currentUser, top3UserIds, onBan, onUnban, isBanned, isSelf, bannedUser,
1005   assignTask // pass this function from parent
1006 }) {
1007   const [showTaskInput, setShowTaskInput] = useState(false);
1008   const [taskInput, setTaskInput] = useState('');
1009
1010   const isTop3User = top3UserIds.includes(currentUser?.id);
1011   const isUserTop3Rank = rank < 3;
1012
1013   return (
1014     <li style={{
1015       listStyle: 'none',
1016       marginBottom: '20px',
1017       backgroundColor: isUserTop3Rank ? '#ffffae6' : 'transparent',
1018       borderRadius: '8px',
1019       padding: '10px',
1020       boxShadow: isUserTop3Rank ? '0 0 10px rgba(255, 215, 0, 0.5)' : 'none',
1021     }}>
1022     <div style={{ display: 'flex', alignItems: 'center' }}>
1023       {user.profile_pic && (
1024         <img
1025           src={user.profile_pic}
1026           alt={` ${user.nickname} || 'User' }'s avatar`}
1027           style={{
1028             width: '50px',
1029             height: '50px',
1030             borderRadius: '50%',
1031             marginRight: '10px',
1032             objectFit: 'cover',
1033           }}
1034         />
1035       )}
1036       <div style={{ flexGrow: 1 }}>
1037         <strong>#{rank + 1}</strong> {user.nickname} | {user.christma_points} ✨

```

```

1038     <br />
1039     Хүйс: {user.gender}
1040 </div>
1041
1042 <div>
1043     <button onClick={() => onLike(user.id)}>❤️ Like</button>
1044     {!askedUserIds.includes(user.id) && (
1045         <button onClick={() => onAskDate(user.id)}>📅 Болзоо</button>
1046     )}
1047     <button onClick={() => onViewProfile(user.id)}>👤 Profile үзэх</button>
1048
1049     {isTop3User && !isSelf && (
1050         isBanned
1051         ? <button onClick={() => onUnban(user.id)} style={{ backgroundColor: '
green', color: 'white' }}>
1052             ✅ Unban
1053         </button>
1054         : <button onClick={() => onBan(user.id)} style={{ backgroundColor: '
red', color: 'white' }}>
1055             ❌ Ban
1056         </button>
1057     )}
1058
1059     {isTop3User && isBanned && (
1060         <>
1061             <button onClick={() => setShowTaskInput(!showTaskInput)} style={{
backgroundColor: 'orange', color: 'white' }}>
1062                 📝 Даалгавар
1063             </button>
1064             {showTaskInput && (
1065                 <div>
1066                     <input
1067                         type="text"
1068                         placeholder="Даалгавараа оруулна уу"
1069                         value={taskInput}
1070                         onChange={(e) => setTaskInput(e.target.value)}
1071                     />
1072                     <button onClick={() => {
1073                         assignTask(user.id, taskInput); // send task
1074                         setShowTaskInput(false);
1075                         setTaskInput('');
1076                     }}>
1077                         Даалгавар илгээх
1078                     </button>
1079                 </div>
1080             )}
1081         </>
1082     )}
1083     <button onClick={() => setActiveReportId(user.id)}>
1084         🚩 Report
1085     </button>
1086
1087     {activeReportId === user.id && (
1088         <div style={{ marginTop: '10px' }}>
1089             <input
1090                 type="text"
1091                 placeholder="Report reason"

```

```

1092         value={reportReasons[user.id] || ''}
1093         onChange={(e) =>
1094             setReportReasons(prev => ({
1095                 ...prev,
1096                 [user.id]: e.target.value,
1097             }))
1098         }
1099         autoFocus
1100         style={{
1101             width: '100%',
1102             padding: '8px',
1103             border: '1px solid #ccc',
1104             borderRadius: '4px',
1105         }}
1106     />
1107     <br />
1108     <button onClick={() => handleSubmitReport(user.id)} style={{
marginTop: '5px' }}>
1109         Submit Report
1110     </button>
1111 </div>
1112     })
1113 </div>
1114 </div>
1115 </li>
1116 );
1117 }
1118
1119 async function handleVote(vote) {
1120     if (!activeDate || !datingTask) {
1121         alert('No active task or date. ');
1122         return;
1123     }
1124
1125     try {
1126         const points = vote === 'nice' ? 8 : 1;
1127
1128         // Step 1: Fetch current christmas_points
1129         const { data: profile, error: fetchError } = await supabase
1130             .from('profiles')
1131             .select('christmas_points')
1132             .eq('id', currentUser.id)
1133             .single();
1134
1135         if (fetchError) throw fetchError;
1136
1137         const newPoints = (profile.christmas_points || 0) + points;
1138
1139         // Step 2: Update Christmas points
1140         const { error: updateError } = await supabase
1141             .from('profiles')
1142             .update({ christmas_points: newPoints })
1143             .eq('id', currentUser.id);
1144
1145         if (updateError) throw updateError;
1146     }

```

```

1147
1148     alert(`You gave a "${vote}" vote and earned ${points} Christma points!`);
1149
1150     // Optionally refresh data or fetch a new task
1151
1152   } catch (err) {
1153     console.error('Error handling vote:', err);
1154     alert('Failed to record your vote.');
```

}

}

return (

onClicK={() => navigate('/profile')} style={{ marginBottom: 20 }}

BACK Profile pyy буцах

onClicK={() => navigate('/notifications')} style={{ marginBottom: 10 }}

Бүх мэдэгдэлүүд (Public Notifications)

activeDate && (

style={{ margin: '20px 0', textAlign: 'center' }}

onClicK={endDate} style={{ padding: '10px 20px', backgroundColor: '#ff4d4f', color: '#fff', border: 'none', borderRadius: '5px' }}

❤️ Болзоог дуусгах

)}

/* Single user task UI */

!activeDate && myTask && (

className="p-6 mb-6 bg-yellow-200 border-4 border-yellow-500 rounded-xl shadow-lg animate-pulse text-center"

className="text-2xl font-bold text-red-600 mb-2">📌 Таньд даалгавар ирсэн байна! 📌

className="text-xl text-gray-800">Таны даалгавар: <u>{myTask}</u>!!!</div>

)}

/* Couple task UI */

activeDate && datingTask && (

className="p-6 mb-6 bg-pink-100 border-4 border-pink-400 rounded-xl shadow-lg animate-pulse text-center"

className="text-2xl font-bold text-pink-600 mb-2">💖 Хосын даалгавар! 💖

className="text-2xl font-bold text-pink-600 mb-2">Ямар нэгэн юм болвол манай web асуудал хүлээхгүйг анхааруулья!</div>

className="text-xl text-gray-800">Та хоёрын хамтарсан даалгавар: <u>{datingTask}</u></div>

className="mt-4 text-lg text-gray-700">Хосынхоо даалгаврыг хэр сайн биелүүлснийг сонгоно уу 😊</div>

className="mt-4 flex justify-center space-x-4">

button

onClicK={() => handleVote('nice')}

localhost:49283/0d94a0e7-f380-49ea-8e52-02b533ddf974/

22/24

```

1195         className="px-6 py-2 bg-green-500 text-white font-bold rounded-lg
hover:bg-green-600 transition"
1196     >
1197         👍 Nice
1198     </button>
1199     <button
1200         onClick={() => handleVote('nah')}
1201         className="px-6 py-2 bg-red-500 text-white font-bold rounded-lg
hover:bg-red-600 transition"
1202     >
1203         👎 Nah
1204     </button>
1205 </div> */}
1206 </div>
1207 }}
1208
1209 {incomingDateRequests.length > 0 && (
1210     <div
1211         style={{
1212             marginTop: 30,
1213             padding: 15,
1214             border: '1px solid #ccc',
1215             borderRadius: 8,
1216         }}
1217     >
1218     <h3>📧 Чамд ирсэн болзох саналууд</h3>
1219     <ul style={{ padding: 0 }}>
1220         {incomingDateRequests.map((request) => (
1221             <DateRequestCard
1222                 key={request.id}
1223                 request={request}
1224                 onAccept={handleAccept}
1225                 onReject={handleReject}
1226             />
1227         ))}
1228     </ul>
1229 </div>
1230 }}
1231
1232 <h2>Ранк</h2>
1233
1234 {loading && <p>Loading...</p>}
1235 {error && <p style={{ color: 'red' }}>Error: {error.message}</p>}
1236
1237 {!loading && !error && (
1238     <>
1239         {users.map((user, index) => (
1240             <UserCard
1241                 key={user.id}
1242                 user={user}
1243                 rank={index}
1244                 onLike={onLike}
1245                 onAskDate={onAskDate}
1246                 onViewProfile={handleViewProfile}
1247                 currentUser={currentUser}
1248                 top3UserIds={top3UserIds}
1249                 onBan={handleBanUser}

```

```
1250         onUnban={handleUnbanUser}
1251         isBanned={bannedUserIds.has(user.id)}
1252         isSelf={user.id === currentUser.id || top3UserIds.includes(user.id)}
1253         assignTask={assignTask}
1254     />
1255   )}
1256 </>
1257   )}
1258   <hr></hr>
1259   <h3>Creator: Nazuke</h3>
1260 </div>
1261 );
1262 }
1263
```