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={`${requester.username}'s avatar`}
25             style={{
26               width: '40px',
27               height: '40px',
28               borderRadius: '50%',
29               marginRight: '10px',
30               objectFit: 'cover',
31             }}
32           />
33         )}
34         <span>
35           <strong>{requester?.username || 'Unknown'}</strong> wants to go on a date
   with you!{' '}
36           {status === 'pending' ? '' : `(${status})`}
37         </span>
38       </div>
39       {status === 'pending' && (
40         <div>
41           <button onClick={() => onAccept(id, requester?.username)} style={{
   marginRight: '5px' }}>
42             ✅ Accept
43           </button>
44           <button onClick={() => onReject(id)}>❌ Reject</button>
45         </div>
46       )}
47     </li>
48   );
49 }
50
51 export default function LeaderboardPage() {
52   const [users, setUsers] = useState([]);
```

```
  const [currentUser, setCurrentUser] = useState(null);
  const [top3UserIds, setTop3UserIds] = useState([]);
  const [incomingDateRequests, setIncomingDateRequests] = useState([]);
  const [activeDate, setActiveDate] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const navigate = useNavigate();
  const [bannedUserIds, setBannedUserIds] = useState(new Set());
  const [myTask, setMyTask] = useState('');
  const [datingTask, setDatingTask] = useState(null);
  const [activeReportId, setActiveReportId] = useState(null);
  const [reportReasons, setReportReasons] = useState({});
  const [askedUserIds, setAskedUserIds] = useState([]);



  // after fetchCurrentUser, also fetch bans:
  useEffect(() => {
    if (currentUser?.id) {
      fetchBans();
    }
  }, [currentUser]);

  // existing ban handler
  useEffect(() => {
    fetchCurrentUser();
  }, []);

  useEffect(() => {
    if (currentUser?.id) {
      fetchLeaderboard();
      fetchIncomingDateRequests();
      fetchActiveDate();
      fetchBans();  // also fetch bans here
    }
  }, [currentUser]);

  async function assignRandomTaskToCouple(coupleId) {
    const coupleRes = await supabase
      .from('dating')
      .select('user1_id, user2_id')
      .eq('id', coupleId)
      .single();

    if (coupleRes.error) {
      console.error('Error fetching couple:', coupleRes.error);
      return;
    }

    const { user1_id, user2_id } = coupleRes.data;

    // Fetch usernames of both users
    const profileRes = await supabase
      .from('profiles')
      .select('id, username')
      .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 гуигтай юм уу хамгийн хөгжилтэй уншсан уран зохиолуудаа ярих.",
120       "Эрэгтэй нь эмэгтэйгээ гэрээс очиж аваад зайсан толгой гараад хот би тэрэнд хайртай гээд орилох, орилж байгаагаа бичлэг болгоод групп дээр шэйр.",
121       "Хамтдаа кино үзчихээд юм авж идэнгээ хотын гудамжаар алхах.",
122       "Гэрээсээ хоол бэлдэж ирээд хамт уул руу гарж идэнгээ нэгэнийхээ зургийг дарж өгөх. (гоё хосийн зурагтай болох)",
123       "Өөрсдөө болзоогоо төлөвлөөд Youtube дээр влог болгож оруулах. Тэгээд групп дээр линкээ өг.",
124       "Хамт кино ямар ч хамаагүй байдлаар үзээд. Киноны хамгийн гоё хэсэг юу байсныг яриад тэр киногоо үнэлэх. Зураг эдр дарвал групп дээр шэйр",
125       "Нэгэндээ тоглодог 2 online, 1 offline тоглоомоо хуваалц. Тэгээд нэгийн сонгоод нэгнийхээ рекордыг эвд эсвэл нэгэнтэйгээ хамт тогло.",
126       "Дуртай 3 дуугаа нэгэнтэйгээ хуваалцаад, Нэг дуугаа сонгоод хамт дуулаад, тэрийгээ групп дээр шэйр.",
127       "30 минутын болзооны сорил. 5 минутанд өдөр тутамд юу хийдгээ, 10 минутанд хамгийн их мөрөөддөг зүйлээ, 15 минутанд чөлөөт цагаа хэрхэн өнгөрөөдөгөө.",
128       "Нэгэнтэйгээ 1 цаг яринга нэгэнийхээ хувцуудыг хараад ямар хувцас өмсөхийн заах. Тэгээд зурагаа даруулаад групп дээр шэйр."
129     ];
130
131     const randomTask = tasks[Math.floor(Math.random() * tasks.length)];
132
133     // Insert tasks for both users
134     const insertRes = await supabase.from('couple_task').insert([
135       {
136         assigner_id: null,
137         assignee_id: user1_id,
138         task_text: randomTask,
139         assigned_at: new Date(),
140         completed: false,
141         partner_rated: false,
142         partner_rating: null,
143       },
144       {
145         assigner_id: null,
146         assignee_id: user2_id,
147         task_text: randomTask,
148         assigned_at: new Date(),
149         completed: false,
150         partner_rated: false,
151         partner_rating: null,
152       },
153     ]);
154
155     if (insertRes.error) {
156       console.error('Error assigning couple task:', insertRes.error);
157       return;
```

```
    }

    // Add public notification
    const message = `Шинэ хосийн даалгавар ${usernames[0]}, ${usernames[1]} хоёрт оногдлоо! 💑 Даалгавар: "${randomTask}"`;

    const notifRes = await supabase.from('notifications').insert({
      user_id: null,
      message,
    });

    if (notifRes.error) {
      console.error('Error creating notification:', notifRes.error);
    } else {
      console.log(message);
    }
  }


  const assignTask = async (assigneeId, taskText) => {
    const { error } = await supabase.from('assigned_tasks').insert({
      assigner_id: currentUser.id,
      assignee_id: assigneeId,
      task_text: taskText,
    });

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

    if (error) {
      console.error('Error assigning task:', error);
      alert('Даалгавар өгөлт амжилтгүй.');
    } else {
      // Also send a public notification
      await supabase.from('notifications').insert({
        user_id: null,
        message: `${data.username}-д даалгавар ирлээ. Даалгавар нь "${taskText}". 🧠`,
      });

      alert('Даалгавар амжилттай өгөгдлөө!');
    }
  };

  const handleSubmitReport = async (reportedId) => {
    const reason = reportReasons[reportedId] || '';
    if (!reason.trim()) return;

    const { error } = await supabase.from('reports').insert({
      reporter_id: currentUser.id,
      reported_id: reportedId,
      reason: reason.trim(),
    });

```

```
    if (error) {
      console.error('Supabase insert error:', error);
      alert('Failed to report');
    }
    else {
      alert('Reported successfully');
      setActiveReportId(null);
      setReportReasons(prev => ({ ...prev, [reportedId]: '' }));
    }
    console.log('Current user:', currentUser);
  };



  useEffect(() => {
    async function fetchMyTask() {
      const { data, error } = await supabase
        .from('assigned_tasks')
        .select('task_text')
        .eq('assignee_id', currentUser.id)
        .order('assigned_at', { ascending: false })
        .limit(1)
        .single();

      if (error && error.code !== 'PGRST116') {
        console.error('Error fetching task:', error);
      } else if (data) {
        setMyTask(data.task_text);
      }
    }

    if (currentUser?.id) {
      fetchMyTask();
    }
  }, [currentUser?.id]);




  async function fetchTop3UserIds() {
    const { data, error } = await supabase
      .from('profiles')
      .select('id')
      .order('christma_points', { ascending: false })
      .limit(3);

    if (!error && data) {
      setTop3UserIds(data.map(user => user.id));
      return data.map(user => user.id);
    } else {
      console.error('Failed to fetch top 3 users', error);
      return [];
    }
  }

  useEffect(() => {
```

```
    fetchTop3UserIds();
  }, []);

  async function fetchBans() {
    const top3Ids = await fetchTop3UserIds();

    const { data: bans, error } = await supabase
      .from('bans')
      .select('banned_user_id')
      .eq('banned_by_id', currentUser.id);

    const { date: profiles} = await supabase
      .from('profiles')
      .select('is_banned, id')
      .eq('id', bans.banned_user_id)
      .eq('is_banned', true)

    if (error) {
      console.error('Error fetching bans:', error);
      return;
    }

    const filteredBans = bans.filter(bans => !top3Ids.includes(bans.banned_user_id));

    setBannedUserIds(new Set(filteredBans.map(b => b.banned_user_id)));
  }


  // ✅ Corrected ban handler with proper notification
  async function handleBanUser(bannedUserId) {
    // Fetch top 3 user IDs
    const top3Ids = await fetchTop3UserIds();

    // Check if currentUser is in top 3
    const isCurrentUserTop3 = top3Ids.includes(currentUser.id);

    if (isCurrentUserTop3) {
      // Check if ANY of the top 3 users have already banned someone
      const { data: existingBans, error } = await supabase
        .from('bans')
        .select('*')
        .in('banned_by_id', top3Ids);

      if (error) {
        console.error('Error checking existing bans:', error);
        return;
      }

      if (existingBans.length >= 1) {
        alert('Эхний гурав дундаа ганц л хүнийг бандах боломжтой. Энэ хэрэглэгчийг бандахын тулд эхлээд бандуулсан байгаа хэрэглэгчийг бангаас нь гаргана уу.');
        return;
      }
    }

    // Check if banned user is currently dating
```

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

    // Fetch banned user's username for notification
    const { data: bannedUserProfile, error: profileError } = await supabase
      .from('profiles')
      .select('username')
      .eq('id', bannedUserId)
      .single();

    if (profileError) {
      console.error('Error fetching banned user profile:', profileError);
      alert('Failed to fetch banned user info.');
      return;
    }

    // Proceed to ban user
    const { error: insertError } = await supabase
      .from('bans')
      .insert([{ banned_user_id: bannedUserId, banned_by_id: currentUser.id }]);

    if (insertError) {
      console.error('Error banning user:', insertError);
      alert('Failed to ban user.');
      return;
    }

    // Insert notification AFTER successful ban
    await supabase.from('notifications').insert({
      user_id: null,
      message: `Эхний гурав ${bannedUserProfile.username} ийг бандлаа! 🚫`,
    });

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

    await fetchBans();       // refresh banned list
    fetchLeaderboard();      // refresh leaderboard if needed
  }



  // ✅ Corrected unban handler
  async function handleUnbanUser(unbanUserId) {
    await supabase
      .from('bans')
      .delete()
      .eq('banned_user_id', unbanUserId)
      .eq('banned_by_id', currentUser.id); // FIXED HERE

    // Fetch banned user's username for notification
    const { data: unBannedUserProfile, error: profileError } = await supabase
      .from('profiles')
      .select('username')
```

```
      .eq('id', unbanUserId)
      .single();


    // Insert notification AFTER successful ban
    await supabase.from('notifications').insert({
      user_id: null, // or the user you want to notify if any
      message: `${currentUser.username} нь ${unBannedUserProfile.username} ийг бангаас гаргалаа! ✅`,
    });

    await fetchBans();
  }


  async function isUserDating(userId) {
    const { data, error } = await supabase
      .from('dating')
      .select('id')
      .or(`user1_id.eq.${userId},user2_id.eq.${userId}`)
      .eq('status', 'accepted') // ✅ Only accepted counts as dating
      .limit(1)
      .maybeSingle();

    if (error && error.code !== 'PGRST116') {
      console.error('Error checking dating status:', error);
      return false;
    }

    return !!data; // true if currently dating
  }



  async function fetchCurrentUser() {
    try {
      setLoading(true);
      setError(null);
      const {
        data: { user },
      } = await supabase.auth.getUser();
      if (!user) throw new Error('User not logged in');

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

      setCurrentUser(profile);
    } catch (err) {
      setError(err);
    } finally {
      setLoading(false);
    }
  }
```

```
435
436   async function fetchLeaderboard() {
437     try {
438       setLoading(true);
439       const { data } = await supabase
440         .from('profiles')
441         .select('id, username, profile_pic, christma_points, gender')
442         .order('christma_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(
```

```
          `and(requester_id.eq.${currentUser.id},status.eq.accepted),
and(requested_id.eq.${currentUser.id},status.eq.accepted)`
        )
        .maybeSingle(); // changed from .single() to .maybeSingle()

      if (error) {
        console.error('Error fetching active date:', error);
        setActiveDate(null);
        return;
      }

      setActiveDate(data || null);
    } catch (err) {
      console.error('Error fetching active date:', err);
      setActiveDate(null);
    }
  }



  // Assume you have currentUser and activeDate available
  useEffect(() => {
    if (!currentUser?.id || !activeDate?.id) {
      setDatingTask(null);
      return;
    }

    async function getPartnerId() {
      // Fix .or() syntax: wrap conditions in parentheses
      const { data, error } = await supabase
      .from('dating')
      .select('user1_id, user2_id')
      .or(`user1_id.eq.${currentUser.id},user2_id.eq.${currentUser.id}`)
      .single();

      if (error && error.code !== 'PGRST116') {
        console.error('Error fetching dating partner:', error);
        return null;
      }
      if (!data) {
        console.log('No dating row found');
        return null;
      }

      const partnerId = data.user1_id === currentUser.id ? data.user2_id :
data.user1_id;
      console.log('PartnerId found:', partnerId);
      return partnerId;
    }

    async function fetchDatingTask() {
      const partnerId = await getPartnerId();
      if (!partnerId) {
        setDatingTask(null);
        return;
      }

```

```
      const { data, error } = await supabase
        .from('couple_task')
        .select('*')
        .in('assignee_id', [currentUser.id, partnerId])
        .order('assigned_at', { ascending: false })
        .limit(1)               // Add limit(1) for safety
        .single();

      if (error && error.code !== 'PGRST116') {
        console.error('Error fetching dating task:', error);
        setDatingTask(null);
      } else if (data) {
        console.log('Dating task found:', data);
        setDatingTask(data.task_text);
      } else {
        setDatingTask(null);
      }
    }

    fetchDatingTask();
  }, [currentUser?.id, activeDate?.id]);




  async function onLike(likedUserId) {
    if (!currentUser || currentUser.id === likedUserId) return;

    try {
      // Step 1: Ban check
      const { data: bannedUsers, error: bannedUsersError } = await supabase
        .from('bans')
        .select('banned_user_id')
        .in('banned_user_id', [currentUser.id, likedUserId]);

      if (bannedUsersError) throw bannedUsersError;

      if (bannedUsers && bannedUsers.length > 0) {
        if (bannedUsers.some(b => b.banned_user_id === currentUser.id)) {
          alert('Та бандуулсан байгаа тул like явуулах боломжгүй.');
        } else {
          alert('Энэ хэрэглэгч бандуулсан байгаа тул та like явуулж болохгүй.');
        }
        return;
      }

      // Step 2: Check if liked today
      const today = new Date().toISOString().split('T')[0]; // YYYY-MM-DD

      const { data: existingLikes, error: likeCheckError } = await supabase
        .from('likes')
        .select('id')
        .eq('liker_id', currentUser.id)
        .eq('liked_id', likedUserId)
        .eq('like_day', today);
```

```
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
    уу.');
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')
```

```
        .select('username, christma_points, gender')
        .eq('id', requestedId)
        .single();

      if (requestedProfileError) throw requestedProfileError;

      // ❌ Gender rule
      if (currentProfile.gender === requestedProfile.gender) {
        alert('Зөвхөн эсрэг хүйстэн рүүгээ болзооны санал явуулна уу.');
        return;
      }

      // ❌ Christma points rule
      if (currentProfile.christma_points < requestedProfile.christma_points) {
        alert('Та зөвхөн өөрөөсөө БАГА оноотой хүн рүү болзооны санал явуулах эрхтэй')
;
        return;
      }

      // ✅ No duplicate request
      const { data: existingRequest, error: existingRequestError } = await supabase
        .from('date_requests')
        .select('*')
        .or(
          `and(requester_id.eq.${currentUser.id},requested_id.eq.${requestedId}),
and(requester_id.eq.${requestedId},requested_id.eq.${currentUser.id})`
        )
        .maybeSingle();

      if (existingRequestError) throw existingRequestError;

      if (existingRequest?.status === 'pending') {
        alert('Та аль хэдийн болзооны санал явуулсан байна.');
        return;
      }

      if (existingRequest?.status === 'accepted') {
        alert('Та энэ хүнтэй аль хэдийн болзож байна.');
        return;
      }

      // 📬 Insert new request
      await supabase
        .from('date_requests')
        .insert([{ requester_id: currentUser.id, requested_id: requestedId, status: '
pending' }]);

      // 🎁 Update Christma points
      const isRequesterTop3 = top3UserIds.includes(currentUser.id);
      const pointsToAdd = isRequesterTop3 ? 10 : 5;
      const updatedPoints = requestedProfile.christma_points + pointsToAdd;

      await supabase
        .from('profiles')
        .update({ christma_points: updatedPoints })
        .eq('id', requestedId);
```

```
765       // 🔔 Notification
766       await supabase.from('notifications').insert({
767         user_id: null,
768         message: `${currentProfile.username} нь ${requestedProfile.username} рүү болзооны санал явууллаа! 💌`,
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 болзож 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=${newCurrentUserCount},
    requester=${newRequesterCount}`);
854
855       // 6. Insert into dating table
856       const { data: couple, error: datingError } = await supabase
857         .from('dating')
858         .insert([{ user1_id: dateData.requester_id, user2_id: currentUser.id }])
859         .select()
860         .single();
861
862       if (datingError) throw datingError;
863
864       console.log('✅ Couple created with id:', couple.id);
865
866       // 6.5 Give 5 Christma points to requester
867       const updatedRequesterPoints = (requesterProfile?.christma_points || 0) + 5;
868
869       await supabase
870         .from('profiles')
871         .update({ christma_points: updatedRequesterPoints })
872         .eq('id', dateData.requester_id);
873
```

```
      // 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({
```

```
        user_id: null,
        message: `💔 ${requesterProfile.username} ба ${requestedProfile.username} болзоогоо дуусгалаа.`,
      });

      alert('Та болзоогоо дуусгалаа.');

      setActiveDate(null);
      fetchLeaderboard();
      fetchIncomingDateRequests();
    } catch (err) {
      console.error('Error ending date:', err);
    }
  }

  // ✅ define this inside the component
  const handleViewProfile = (userId) => {
    navigate(`/profile-view/${userId}`);
  }

  function UserCard({
    user, rank, onLike, onAskDate, onViewProfile,
    currentUser, top3UserIds, onBan, onUnban, isBanned, isSelf, bannedUser,
    assignTask // pass this function from parent
  }) {
    const [showTaskInput, setShowTaskInput] = useState(false);
    const [taskInput, setTaskInput] = useState('');

    const isTop3User = top3UserIds.includes(currentUser?.id);
    const isUserTop3Rank = rank < 3;

    return (
      <li style={{
        listStyle: 'none',
        marginBottom: '20px',
        backgroundColor: isUserTop3Rank ? '#fffae6' : 'transparent',
        borderRadius: '8px',
        padding: '10px',
        boxShadow: isUserTop3Rank ? '0 0 10px rgba(255, 215, 0, 0.5)' : 'none',
      }}>
        <div style={{ display: 'flex', alignItems: 'center' }}>
          {user.profile_pic && (
            <img
              src={user.profile_pic}
              alt={`${user.nickname || 'User'}'s avatar`}
              style={{
                width: '50px',
                height: '50px',
                borderRadius: '50%',
                marginRight: '10px',
                objectFit: 'cover',
              }}
            />
          )}
          <div style={{ flexGrow: 1 }}>
            <strong>#{rank + 1}</strong> {user.nickname} | {user.christma_points}✨
```

```
            <br />
            Хүйс: {user.gender}
          </div>

          <div>
            <button onClick={() => onLike(user.id)}>❤️ Like</button>
            {!askedUserIds.includes(user.id) && (
              <button onClick={() => onAskDate(user.id)}>💌 Болзоо</button>
            )}
            <button onClick={() => onViewProfile(user.id)}>👀 Profile үзэх</button>

            {isTop3User && !isSelf && (
              isBanned
                ? <button onClick={() => onUnban(user.id)} style={{ backgroundColor: '
green', color: 'white' }}>
                    ✅ Unban
                  </button>
                : <button onClick={() => onBan(user.id)} style={{ backgroundColor: '
red', color: 'white' }}>
                    🚫 Ban
                  </button>
            )}

            {isTop3User && isBanned && (
              <>
                <button onClick={() => setShowTaskInput(!showTaskInput)} style={{
backgroundColor: 'orange', color: 'white' }}>
                  📝 Даалгавар
                </button>
                {showTaskInput && (
                  <div>
                    <input
                      type="text"
                      placeholder="Даалгавараа оруулна уу"
                      value={taskInput}
                      onChange={(e) => setTaskInput(e.target.value)}
                    />
                    <button onClick={() => {
                      assignTask(user.id, taskInput); // send task
                      setShowTaskInput(false);
                      setTaskInput('');
                    }}>
                      Даалгавар илгээх
                    </button>
                  </div>
                )}
              </>
            )}
            <button onClick={() => setActiveReportId(user.id)}>
              🚩 Report
            </button>

            {activeReportId === user.id && (
              <div style={{ marginTop: '10px' }}>
                <input
                  type="text"
                  placeholder="Report reason"
```

```
                  value={reportReasons[user.id] || ''}
                  onChange={(e) =>
                    setReportReasons(prev => ({
                      ...prev,
                      [user.id]: e.target.value,
                    }))
                  }
                  autoFocus
                  style={{
                    width: '100%',
                    padding: '8px',
                    border: '1px solid #ccc',
                    borderRadius: '4px',
                  }}
                />
                <br />
                <button onClick={() => handleSubmitReport(user.id)} style={{ marginTop: '5px' }}>
                  Submit Report
                </button>
              </div>
            )}
          </div>
        </div>
      </li>
    );
  }

  async function handleVote(vote) {
    if (!activeDate || !datingTask) {
      alert('No active task or date.');
      return;
    }

    try {
      const points = vote === 'nice' ? 8 : 1;

      // Step 1: Fetch current christma_points
      const { data: profile, error: fetchError } = await supabase
        .from('profiles')
        .select('christma_points')
        .eq('id', currentUser.id)
        .single();

      if (fetchError) throw fetchError;

      const newPoints = (profile.christma_points || 0) + points;

      // Step 2: Update Christma points
      const { error: updateError } = await supabase
        .from('profiles')
        .update({ christma_points: newPoints })
        .eq('id', currentUser.id);

      if (updateError) throw updateError;

```

```
      alert(`You gave a "${vote}" vote and earned ${points} Christma points!`);

      // Optionally refresh data or fetch a new task

    } catch (err) {
      console.error('Error handling vote:', err);
      alert('Failed to record your vote.');
    }
  }

  return (
    <div style={{ maxWidth: 600, margin: 'auto', padding: 20 }}>
      <button onClick={() => navigate('/profile')} style={{ marginBottom: 20 }}>
        🔙 Profile руу буцах
      </button>
      <button onClick={() => navigate('/notifications')} style={{ marginBottom: 10 }}>
        📣 Бүх мэдэгдэлүүд (Public Notifications)
      </button>

      {activeDate && (
      <div style={{ margin: '20px 0', textAlign: 'center' }}>
        <button onClick={endDate} style={{ padding: '10px 20px', backgroundColor: '#ff4d4f', color: '#fff', border: 'none', borderRadius: '5px' }}>
          💔 Болзоог дуусгах
        </button>
      </div>
    )}

      {/* Single user task UI */}
      {!activeDate && myTask && (
        <div className="p-6 mb-6 bg-yellow-200 border-4 border-yellow-500 rounded-xl shadow-lg animate-pulse text-center">
          <h2 className="text-2xl font-bold text-red-600 mb-2">🚨 Таньд даалгавар ирсэн байна! 🚨</h2>
          <h3 className="text-xl text-gray-800">Таны даалгавар: <u>{myTask}</u>!!!</h3>
        </div>
      )}

      {/* Couple task UI */}
      {activeDate && datingTask &&(
        <div className="p-6 mb-6 bg-pink-100 border-4 border-pink-400 rounded-xl shadow-lg animate-pulse text-center">
          <h2 className="text-2xl font-bold text-pink-600 mb-2">💞 Хосын даалгавар! 💞</h2>
          <h3 className="text-2xl font-bold text-pink-600 mb-2">Ямар нэгэн юм болвол манай web асуудал хүлээхгүйг анхааруулья!</h3>
          <h3 className="text-xl text-gray-800">Та хоёрын хамтарсан даалгавар: <u>{datingTask}</u></h3>

          {/* <h3 className="mt-4 text-lg text-gray-700">Хосынхоо даалгаврыг хэр сайн биелүүлснийг сонгоно уу 🙂</h3>

          <div className="mt-4 flex justify-center space-x-4">
            <button
              onClick={() => handleVote('nice')}
```

```
              className="px-6 py-2 bg-green-500 text-white font-bold rounded-lg hover:bg-green-600 transition"
            >
              👍 Nice
            </button>
            <button
              onClick={() => handleVote('nah')}
              className="px-6 py-2 bg-red-500 text-white font-bold rounded-lg hover:bg-red-600 transition"
            >
              👎 Nah
            </button>
          </div> */}
        </div>
      )}

      {incomingDateRequests.length > 0 && (
          <div
            style={{
            marginTop: 30,
            padding: 15,
            border: '1px solid #ccc',
            borderRadius: 8,
          }}
        >
          <h3>📩 Чамд ирсэн болзох саналууд</h3>
          <ul style={{ padding: 0 }}>
            {incomingDateRequests.map((request) => (
              <DateRequestCard
                key={request.id}
                request={request}
                onAccept={handleAccept}
                onReject={handleReject}
              />
            ))}
          </ul>
        </div>
      )}

      <h2>Ранк</h2>

      {loading && <p>Loading...</p>}
      {error && <p style={{ color: 'red' }}>Error: {error.message}</p>}

      {!loading && !error && (
        <>
            {users.map((user, index) => (
              <UserCard
                key={user.id}
                user={user}
                rank={index}
                onLike={onLike}
                onAskDate={onAskDate}
                onViewProfile={handleViewProfile}
                currentUser={currentUser}
                top3UserIds={top3UserIds}
                onBan={handleBanUser}
```

```
                onUnban={handleUnbanUser}
                isBanned={bannedUserIds.has(user.id)}
                isSelf={user.id === currentUser.id || top3UserIds.includes(user.id)}
                assignTask={assignTask}
              />
            ))}
        </>
      )}
      <hr></hr>
      <h3>Creator: Nazuke</h3>
    </div>
  );
}

```