



Dhirubhai Ambani **University** Technology

Formerly DA-IICT

Software Engineering

Group 27

Project Title : Connect Sphere

Unit Testing

Professor: Saurabh Tiwari

Introduction:

This document details the unit testing effort for the Connect-Sphere application, utilizing the Vitest framework.

Overall Unit Testing:

All files

98.12% Statements 33131/4228294.55% Branches 7637/807791.53% Functions 831/88698.12% Lines 33131/42282

Press n or j to go to the next uncovered block, b, p or k for the previous block.

Filter:

File		Statements		Branches		Functions		Lines	
src		100%	28/28	100%	3/3	0%	0/1	100%	28/28
src/app/api/auth/callback		100%	68/68	96.15%	25/26	100%	5/5	100%	68/68
src/app/api/auth/callback/route.test		100%	338/338	100%	25/25	100%	3/3	100%	338/338
src/app/api/messages/[messageId]		98.82%	252/255	95.83%	69/72	100%	11/11	98.82%	252/255
src/app/api/messages/blocks		100%	108/108	100%	37/37	100%	4/4	100%	108/108
src/app/api/messages/blocks/route.test		100%	249/249	100%	32/32	100%	3/3	100%	249/249
src/app/api/messages/conversations		97.64%	1076/1102	91.24%	271/297	97.22%	35/36	97.64%	1076/1102
src/app/api/messages/conversations/route.test		100%	265/265	100%	24/24	100%	4/4	100%	265/265
src/app/api/moderation/bans		100%	914/914	99.58%	240/241	100%	10/10	100%	914/914
src/app/api/moderation/bans/route.test		99.47%	378/380	93.44%	57/61	100%	7/7	99.47%	378/380
src/app/api/moderation/reports		98.69%	227/230	95.58%	65/68	100%	9/9	98.69%	227/230
src/app/api/moderation/reports/route.test		98.63%	216/219	95.12%	39/41	100%	4/4	98.63%	216/219
src/app/api/payments/create-order		99.4%	335/337	97.05%	66/68	100%	5/5	99.4%	335/337
src/app/api/payments/verify		100%	348/348	98.11%	52/53	100%	4/4	100%	348/348
src/app/api/reports		100%	305/305	100%	68/68	100%	7/7	100%	305/305
src/app/api/reports/route.test		100%	137/137	100%	15/15	100%	2/2	100%	137/137
src/app/communities		100%	1348/1348	100%	219/219	100%	7/7	100%	1348/1348
src/components		98.89%	539/545	90.42%	85/94	78.57%	11/14	98.89%	539/545
src/components/analytics		100%	160/160	100%	23/23	100%	8/8	100%	160/160
src/components/auth		95.07%	637/670	86.58%	71/82	92%	23/25	95.07%	637/670
src/components/communities		99.42%	3607/3628	96.09%	713/742	94.78%	109/115	99.42%	3607/3628

src/components/communities/communities-list.test	<div></div>	100%	133/133	100%	16/16	100%	1/1	100%	133/133
src/components/communities/community-actions.test	<div></div>	100%	86/86	100%	12/12	100%	0/0	100%	86/86
src/components/communities/community-card.test	<div></div>	100%	80/80	100%	10/10	100%	0/0	100%	80/80
src/components/communities/community-comments-section.test	<div></div>	100%	164/164	100%	21/21	100%	0/0	100%	164/164
src/components/communities/community-members-list.test	<div></div>	100%	141/141	100%	10/10	100%	0/0	100%	141/141
src/components/communities/community-members-manager.test	<div></div>	100%	164/164	100%	11/11	100%	0/0	100%	164/164
src/components/communities/community-post-card.test	<div></div>	100%	164/164	100%	11/11	100%	0/0	100%	164/164
src/components/communities/create-community-form.test	<div></div>	100%	84/84	86.66%	13/15	100%	2/2	100%	84/84
src/components/communities/create-community-post.test	<div></div>	98.12%	209/213	96.87%	31/32	100%	3/3	98.12%	209/213
src/components/communities/payment-modal.test	<div></div>	99.7%	1016/1019	100%	132/132	100%	37/37	99.7%	1016/1019
src/components/feed	<div></div>	97.19%	9054/9315	94.76%	1975/2084	87.21%	116/133	97.19%	9054/9315
src/components/messages	<div></div>	94.83%	2624/2767	79.56%	475/597	79.45%	58/73	94.83%	2624/2767
src/components/messages/new-chat-dialog.test	<div></div>	100%	406/406	100%	40/40	100%	5/5	100%	406/406
src/components/messages/report-user-dialog.test	<div></div>	100%	316/316	100%	24/24	100%	2/2	100%	316/316
src/components/moderation	<div></div>	97.61%	983/1007	83.69%	195/233	81.25%	26/32	97.61%	983/1007
src/components/notifications	<div></div>	98.68%	527/534	92.65%	164/177	100%	22/22	98.68%	527/534
src/components/notifications/collab-invites.test	<div></div>	99.47%	191/192	92.59%	25/27	100%	1/1	99.47%	191/192
src/components/notifications/notification-bell.test	<div></div>	100%	212/212	100%	39/39	100%	0/0	100%	212/212
src/components/notifications/notification-item.test	<div></div>	98.95%	190/192	100%	35/35	75%	3/4	98.95%	190/192
src/components/notifications/notification-list.test	<div></div>	99.7%	340/341	97.36%	74/76	100%	1/1	99.7%	340/341
src/components/privacy	<div></div>	100%	837/837	100%	129/129	100%	15/15	100%	837/837
src/components/profile	<div></div>	99.19%	4709/4747	97.79%	711/727	96%	48/50	99.19%	4709/4747
src/components/search	<div></div>	100%	203/203	97.22%	70/72	100%	10/10	100%	203/203
src/components/search/search-bar.test	<div></div>	99.56%	682/685	98.87%	88/89	80%	4/5	99.56%	682/685
src/components/settings	<div></div>	100%	240/240	91.93%	57/62	90%	9/10	100%	240/240
src/components/settings/archived-posts-list-ui.test	<div></div>	100%	244/244	100%	45/45	100%	5/5	100%	244/244

src/components/settings/profile-form.test	<div></div>	100%	167/167	100%	23/23	100%	4/4	100%	167/167
src/components/settings/verification-status-card.test	<div></div>	100%	168/168	100%	7/7	100%	0/0	100%	168/168
src/components/stories	<div></div>	100%	124/124	93.1%	27/29	100%	4/4	100%	124/124
src/components/stories/StoryUploadForm.test	<div></div>	100%	63/63	100%	11/11	100%	0/0	100%	63/63
src/components/ui	<div></div>	96%	1202/1252	97.19%	104/107	100%	12/12	96%	1202/1252
src/components/ui/dropdown-menu.test	<div></div>	100%	55/55	100%	7/7	100%	0/0	100%	55/55
src/components/ui/separator.test	<div></div>	100%	31/31	100%	5/5	100%	0/0	100%	31/31
src/components/ui/sheet.test	<div></div>	100%	41/41	100%	6/6	100%	0/0	100%	41/41
src/components/ui/switch.test	<div></div>	100%	25/25	100%	5/5	100%	0/0	100%	25/25
src/components/ui/toast.test	<div></div>	100%	77/77	100%	10/10	100%	0/0	100%	77/77
src/components/ui/tooltip.test	<div></div>	100%	33/33	100%	5/5	100%	0/0	100%	33/33
src/hooks	<div></div>	63.04%	203/322	100%	54/54	93.33%	14/15	63.04%	203/322
src/hooks/use-mobile.test	<div></div>	100%	119/119	100%	26/26	100%	2/2	100%	119/119
src/hooks/use-realtime.test	<div></div>	100%	404/404	100%	55/55	100%	1/1	100%	404/404
src/hooks/use-toast.test	<div></div>	100%	347/347	100%	86/86	100%	9/9	100%	347/347
src/lib	<div></div>	97.85%	1639/1675	93.76%	406/433	82%	82/100	97.85%	1639/1675
src/lib/post-data.test	<div></div>	100%	224/224	100%	38/38	100%	4/4	100%	224/224
src/lib/supabase	<div></div>	100%	502/502	96.2%	76/79	100%	12/12	100%	502/502
src/lib/supabase/admin.test	<div></div>	100%	51/51	100%	11/11	100%	0/0	100%	51/51
src/lib/verification.test	<div></div>	100%	402/402	100%	63/63	100%	28/28	100%	402/402

App Folder:

1. API :

I. Auth: API[AUTH]

```
✓ src/app/api/auth/callback/route.test/GET.test.ts (19)
✓ GET /callback (19)
  ✓ redirects to next page when authentication succeeds
  ✓ creates profile if none exists
  ✓ redirects to login when code is missing
  ✓ redirects to login when exchangeCodeForSession fails
  ✓ uses metadata username and sanitizes it
  ✓ falls back to 'user-xxxx' username when metadata username is missing
  ✓ uses user metadata.full_name as display_name
  ✓ logs profile creation error but still redirects
  ✓ defaults redirect to /feed when next param is missing
  ✓ cookies.get returns the underlying cookie value
  ✓ cookies.set forwards the correct name, value and options
  ✓ cookies.remove forwards name and options to cookiesStore.delete
  ✓ sanitize() removes invalid chars, trims and lowercases
  ✓ uses idSuffix fallback when sanitize(readUsername) produces empty username
  ✓ uses fallback username when sanitize(readUsername) returns empty
  ✓ correctly chooses displayName based on metadata fallback order
  ✓ uses fallback idSuffix = 'user' when user.id is missing
  ✓ falls back to displayName = 'New User' when all metadata and emailLocal are missing
  ✓ falls back to 'user' when user.id is null or empty

Test Files 1 passed (1)
Tests 19 passed (19)
Start at 11:45:52
Duration 3.85s (transform 118ms, setup 224ms, collect 167ms, tests 50ms, environment 1.05s, prepare 1.08s)

HINT Report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from v8
-----
File      % Stats % Branch % Funcs % Lines Uncovered Line #s
-----
All files 100    98.03    100    100
callback 100    96.15    100    100
route.ts  100    96.15    100    100  54
callback/route.test 100    100    100    100
GET.test.ts 100    100    100    100
-----
```

II. Message : Messgae[ID]

```
✓ src/app/api/messages/messageId/route.test.ts (16)
✓ PATCH /api/messages/messageId (16)
  ✓ 401 Unauthorized
  ✓ 400 messageId is required (NaN)
  ✓ 400 Invalid JSON payload
  ✓ 400 Unsupported action
  ✓ 400 Message not found
  ✓ 500 Unable to delete message (metadata load error)
  ✓ 500 Unable to delete message (participant verify error)
  ✓ 403 Not participant of conversation
  ✓ 403 delete-for-me only on messages
  ✓ Success delete-for-me updates deleted by when not already deleted
  ✓ No-op delete-for-me if already deleted by contains user
  ✓ 500 delete-for-me update error
  ✓ 403 delete-for-everyone only sender allowed
  ✓ Success delete-for-everyone sets flag and updates deleted_by
  ✓ No-op delete-for-everyone when already flagged
  ✓ 500 delete-for-everyone update error

Test Files 1 passed (1)
Tests 16 passed (16)
Start at 11:47:29
Duration 3.79s (transform 124ms, setup 188ms, collect 182ms, tests 18ms, environment 1.21s, prepare 792ms)

HINT Report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from v8
-----
File      % Stats % Branch % Funcs % Lines Uncovered Line #s
-----
All files 98.82    95.83    100    98.82
route.test.ts 100    100    100    100
route.ts    97.69    91.61    100    97.69 171-173
-----
```

(i). Blocks :

```
✓ src/app/api/messages/blocks/route.test/DELETE.test.ts (7)
✓ DELETE /api/messages/blocks (7)
  ✓ should return 401 if user is not authenticated
  ✓ should return 400 if targetUserId is not provided
  ✓ should return 400 if user tries to unblock themselves
  ✓ should return 404 if block entry is not found
  ✓ should return 500 if there is an error looking up block entry
  ✓ should return 500 when delete operation fails
  ✓ should return success if block entry is deleted successfully
✓ src/app/api/messages/blocks/route.test/POST.test.ts (6)
✓ POST /api/messages/blocks (6)
  ✓ should block a user successfully
  ✓ should return 401 if user is unauthorized
  ✓ should return 400 if JSON payload is invalid
  ✓ should return 400 if targetUserId is missing
  ✓ should return 400 if user tries to block themselves
  ✓ should return 404 if target user does not exist
  ✓ should return 500 if there is an error during upsert
  ✓ should return 500 if target lookup fails

Test Files 2 passed (2)
Tests 15 passed (15)
Start at 11:48:21
Duration 4.07s (transform 150ms, setup 371ms, collect 323ms, tests 33ms, environment 2.24s, prepare 2.23s)

HINT Report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from v8
-----
File      % Stats % Branch % Funcs % Lines Uncovered Line #s
-----
All files 100    100    100    100
blocks    100    100    100    100
route.ts  100    100    100    100
blocks/route.test 100    100    100    100
DELETE.test.ts 100    100    100    100
POST.test.ts 100    100    100    100
-----
```

(ii). Conversation :

```
✓ src/app/api/messages/conversations/route.delete.test.ts (7)
  ✓ DELETE /api/messages/conversations (?)
    ✓ 401 Unauthorized
    ✓ 400 conversationId is required (missing)
    ✓ 400 conversationId is required (NaN)
    ✓ 500 Unable to delete conversation (participant query error)
    ✓ 404 Conversation not found
    ✓ Success delete reveal conversation
    ✓ 500 Could not delete conversation (delete error)
✓ src/app/api/messages/conversations/route.filterHidden.test.ts (3)
  ✓ filterHidden helper (3)
    ✓ returns all visible when none hidden
    ✓ keeps hidden when last message older than hidden_at
    ✓ reveals hidden when last message newer than hidden_at
✓ src/app/api/messages/conversations/route.get.test.ts (3)
  ✓ GET /api/messages/conversations (stable integration) (3)
    ✓ 401 Unauthorized
    ✓ 500 participant load error (conversationParticipants select)
    ✓ generic 500 error path (non-RequestError)
✓ src/app/api/messages/conversations/route.helpers.test.ts (16)
  ✓ getMessageTimestamp (2)
    ✓ returns 0 for null/undefined/empty
    ✓ parses created_at when present
  ✓ filterHidden (4)
    ✓ reveals when last message is newer than hidden_at
    ✓ does not reveal when last message time equals hidden_at (strict >)
    ✓ keeps hidden when no last message
    ✓ shows when not hidden
  ✓ mergeParticipants (3)
    ✓ dedupes profiles across base, fallbacks, and extras
    ✓ handles undefined base map gracefully
    ✓ ignores falsy profiles during merge
  ✓ selectOtherParticipant (2)
    ✓ returns the non-self participant
    ✓ returns undefined when only self is present
  ✓ composeSummaries (5)
    ✓ sorts by last_message created_at descending, then by created_at
    ✓ skips conversations where other participant is a moderator and current is not
    ✓ marks blocked flags correctly
    ✓ skips conversations without an other participant (only self present)
    ✓ uses fallbackProfiles and conversationFallbacks to build participants when base missing
✓ src/app/api/messages/conversations/route.patch.test.ts (18)
  ✓ PATCH /api/messages/conversations (18)
    ✓ 401 Unauthorized
    ✓ 400 Invalid JSON payload
    ✓ 400 conversationId is required (NaN)
    ✓ 400 action is required
    ✓ 400 Unsupported action
    ✓ 500 Unable to update conversation (participant query error)
    ✓ 404 Conversation not found
    ✓ Success hide sets hidden_at timestamp
    ✓ Success unhide clears hidden_at
    ✓ 500 Could not update conversation (update error)
✓ src/app/api/messages/conversations/route.post.test.ts (18)
  ✓ POST /api/messages/conversations (18)
    ✓ 401 Unauthorized
    ✓ 400 Invalid JSON payload
    ✓ 400 targetUserId is required
    ✓ 400 Cannot start conversation with yourself
    ✓ 404 Target moderator hidden when current not moderator
    ✓ 403 You have blocked this user
    ✓ 403 You are blocked by this user
    ✓ 404 Conversation not found after summaries
    ✓ Success returns conversation
    ✓ 500 Could not open conversation (generic)
✓ src/app/api/messages/conversations/route.test.tsx (31)
  ✓ conversations route helpers (4)
    ✓ getMessageTimestamp returns 0 for null/undefined and timestamp for valid message
    ✓ mergeParticipants merges base/fallbacks/extras and dedupes by id
    ✓ selectOtherParticipant returns participant not equal to current user
    ✓ composeSummaries filters out moderators for non-moderator user and sets block flags, sorts by last message date
  ✓ buildConversationSummaries integration (2)
    ✓ returns [] early when conversationIds empty or invalid
    ✓ calls fetchConversationBox, fetchParticipantMap and fetchLastMessageMap and composes results
  ✓ fetchBlockedSets (2)
    ✓ returns sets of blocked and blockedby on success
    ✓ throws RequestError on db failure
  ✓ getProfileThrow (3)
    ✓ returns profile on success
    ✓ throws 404 RequestError when profile not found
    ✓ throws RequestError when 400 error
  ✓ findExistingConversationId/createConversation/ensureConversation/upsertHiddenAt (6)
    ✓ findExistingConversationId returns null when none exist
    ✓ findExistingConversationId returns an id when overlap found
    ✓ createConversation inserts conversation and participants and returns id
    ✓ createConversation throws if participant insert fails
    ✓ ensureConversation uses existing id if found otherwise creates new one
    ✓ upsertHiddenAt updates and throws on error
  ✓ HTTP handlers: POST, GET, PATCH, DELETE (14)
    ✓ POST returns 401 when unauthorized
    ✓ POST handles invalid json
    ✓ POST rejects missing targetUserId and self-target
    ✓ POST returns 404 when target is moderator and current user not moderator
    ✓ POST returns 403 when blocked or blockedby
    ✓ POST creates or returns conversation successfully
    ✓ GET returns 401 when unauthorized
    ✓ GET returns visible conversations and reveals hidden
    ✓ PATCH returns 401 when unauthorized
    ✓ PATCH returns error when missing body params or not a participant
    ✓ PATCH hide/unhide success
    ✓ DELETE returns 401 when unauthorized
    ✓ DELETE handles missing conversationId and participant checks
    ✓ DELETE succeeds when participant exists and records removed
  ✓ DELETE /api/messages/conversations/route.test/DELETE.test.ts (5)
    ✓ DELETE /api/messages/conversations (5)
      ✓ should return 401 if user is not authenticated
      ✓ should return 400 if conversationId is not provided
      ✓ should return 404 if conversation is not found
      ✓ should return 500 if conversation lookup throws error
      ✓ should return success if conversation is deleted
  ✓ src/app/api/messages/conversations/route.test/GET.test.ts (6)
    ✓ GET /api/messages/conversations (6)
      ✓ should return conversations list without hidden reveal update [skipped]
      ✓ should return 500 if conversations fetch fails for current user
      ✓ should return 200 with empty list if no conversations
      ✓ should return 401 when user is not authenticated
      ✓ should return 500 if profile lookup fails
      ✓ should return 500 if block set lookup fails
  ✓ src/app/api/messages/conversations/route.test/POST.test.ts (4)
    ✓ POST /api/messages/conversations (4)
      ✓ returns 401 if user is not authenticated
      ✓ returns 400 if targetUserId is missing
      ✓ returns 400 if targetUserId equals user.id
      ✓ returns 404 if target is moderator but user is not

Test Files 10 passed (18)
Tests 94 passed | 1 skipped (95)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	94.59	91.9	88.63	94.59	
app/api/messages/conversations	97.64	91.24	97.22	97.64	
route.delete.test.ts	100	100	100	100	
route.filterHidden.test.ts	100	100	100	100	
route.get.test.ts	100	100	100	100	
route.helpers.test.ts	100	100	100	100	
route.patch.test.ts	100	100	100	100	
route.post.test.ts	98.06	100	75	98.06	30-32
route.ts	95.76	86.66	100	95.76	80-81,95-96,113,143-144,292-294,344-345,361-362,379-380,405-406,499-500,542-544
app/api/messages/conversations/route.test	100	100	100	100	
DELETE.test.ts	100	100	100	100	
GET.test.ts	100	100	100	100	
POST.test.ts	100	100	100	100	

III. Moderation :

(i). Bans :

```

✓ src/app/api/moderation/bans/route.test.ts (100) 300ms
✓ moderation/bans route.ts - comprehensive mutation killing tests (100)
  ✓ getModeratorContext() (5)
    ✓ returns 401 when auth error exists
    ✓ returns 401 when user is null (no error)
    ✓ returns 403 when profile is null
    ✓ returns 403 when is_moderator is false
    ✓ returns 403 when is_moderator is undefined
  ✓ GET() (21)
    ✓ returns active bans (default status) with profiles
    ✓ filters expired bans when status=active
    ✓ does not filter bans when status=all
    ✓ returns setupRequired when bans query has PGST100 error
    ✓ returns setupRequired when bans query has PGST101 error
    ✓ returns setupRequired when bans query has PGST102 error
    ✓ returns setupRequired when error message includes user_bans
    ✓ returns setupRequired when error details includes user_bans
    ✓ returns setupRequired when error message includes profiles
    ✓ returns 500 for non-schema id errors on bans query
    ✓ returns setupRequired when profiles query has schema error
    ✓ returns setupRequired when profiles error message includes profiles
    ✓ returns setupRequired when profiles error details includes profiles
    ✓ continues without profiles when profiles query has non-schema error
    ✓ skips profiles query when no user_ids
    ✓ handles null data from bans query
    ✓ deduplicates user_ids for profiles query
    ✓ filters out bans with falsy user_id
    ✓ returns 500 on unexpected exception
    ✓ handles error.message being undefined
    ✓ handles error.details being non-string
  ✓ POST() (17)
    ✓ creates ban successfully with all fields
    ✓ creates permanent ban when expiresAt is null
    ✓ creates ban with empty reason (null)
    ✓ creates ban with whitespace-only reason (treated as null)
    ✓ returns 400 when userId is missing
    ✓ returns 400 when userId is not a string
    ✓ returns 400 when userId is empty string
    ✓ returns 400 when expiresAt is invalid date
    ✓ returns 400 when expiresAt is invalid format
    ✓ returns 500 when insert fails
    ✓ returns 500 when RPC refresh fails
    ✓ handles request.json() throwing
    ✓ returns 500 on unexpected exception
    ✓ returns 401 when unauthorized
    ✓ returns 403 when not moderator
    ✓ handles reason being non-string
    ✓ handles expiresAt being non-string (ignored)
  ✓ PATCH() (16)
    ✓ lifts ban successfully with user_id returned
    ✓ lifts ban successfully without liftReason (uses default)
    ✓ lifts ban with empty liftReason (uses default)
    ✓ lifts ban with whitespace-only liftReason (uses default)
    ✓ returns success when updateBan is null (ban not found or already lifted)
    ✓ returns success when updateBan has no user_id
    ✓ returns 400 when banId is missing
    ✓ returns 400 when banId is not a string
    ✓ returns 400 when banId is empty string
    ✓ returns 500 when update fails
    ✓ returns 500 when RPC refresh fails
    ✓ handles request.json() throwing
    ✓ returns 500 on unexpected exception
    ✓ returns 401 when unauthorized
    ✓ returns 403 when not moderator
    ✓ handles liftReason being non-string (uses default)
  ✓ returns liftReason being non-string (uses default)
  ✓ Edge cases - mutation killing (41)
    ✓ keeps ban with expires_at exactly equal to now (edge case)
    ✓ POST: userId as object is rejected
    ✓ POST: reason with only newlines is treated as null
    ✓ POST: expiresAt as number is ignored
    ✓ POST: expiresAt as empty string returns success (empty string is falsy)
    ✓ PATCH: banId as object is rejected
    ✓ PATCH: liftReason with tabs only uses default
    ✓ PATCH: updateBan with user_id as empty string skips RPC
    ✓ GET: error message with USER_BANS (uppercase) triggers setupRequired
    ✓ GET: profiles error with PGST101 triggers setupRequired
    ✓ GET: profiles error with PGST102 triggers setupRequired
    ✓ GET: handles duplicate user_ids correctly
    ✓ GET: ban with user_id not in profiles returns null profile
    ✓ GET: status=active applies is() filter
    ✓ GET: status=all does not apply is() filter for lifted_at
    ✓ POST: body is null (from catch) returns 400
    ✓ PATCH: body is undefined returns 400
    ✓ GET: active filter with expires_at far in future keeps ban
    ✓ GET: active filter with expires_at far in past removes ban
    ✓ GET: calls from() with exact table name 'user_bans'
    ✓ GET: calls from() with exact table name 'profiles' for profile lookup
    ✓ GET: calls select() with exact column list for bans
    ✓ GET: calls select() with exact column list for profiles
    ✓ GET: calls order() with exact perms
    ✓ GET: calls limit() with exact value 200
    ✓ GET: calls in() with exact column name 'id' for profiles
    ✓ POST: calls from() with 'user_bans' for update (supersede)
    ✓ POST: calls update() with supersede fields
    ✓ POST: calls eq() with 'user_id' and target
    ✓ POST: calls is() with 'lifted_at' and null
    ✓ POST: calls insert() with correct fields
    ✓ POST: calls rpc() with 'refresh_profile_ban_state' and target user
    ✓ PATCH: calls from() with 'user_bans' for update
    ✓ PATCH: calls update() with lift fields
    ✓ PATCH: uses default lift reason when not provided
    ✓ PATCH: calls eq() with 'id' and banId
    ✓ PATCH: calls select() with 'user_id'
    ✓ PATCH: calls rpc() with 'refresh_profile_ban_state' and user from updateBan
    ✓ getModeratorContext: calls from() with 'profiles'
    ✓ getModeratorContext: calls select() with 'is_moderator'
    ✓ getModeratorContext: calls eq() with 'id' and user_id

```

```

✓ src/app/api/moderation/bans/route.test/GET.test.ts (4)
  ✓ GET() moderation bans (simple mocks) (4)
    ✓ should return active bans with profiles
    ✓ should return 401 if user is not authenticated
    ✓ should return 403 if user is not a moderator
    ✓ should return 404 when ban is not found
    ✓ should return 409 when ban is already lifted
  ✓ PATCH /api/moderation/bans (lift ban) (5)
    ✓ should lift a ban successfully
    ✓ should return 400 when banId is missing
    ✓ should return 401 when user is unauthorized
    ✓ should return 403 if user is not a moderator
    ✓ should return 404 when ban is not found (update returned no row)
    ✓ should return 409 when ban is already lifted
    ✓ should return 200 success even when ban update returns null-without-error
    ✓ should return success when RPC fails because updatedBan has no user_id
    ✓ should return 400 when request.json throws (banId missing)
✓ src/app/api/moderation/bans/route.test/POST.test.ts (7)
  ✓ POST() moderation bans (7)
    ✓ should successfully create a ban
    ✓ should return 401 if user is not authenticated
    ✓ should return 403 if user is not a moderator
    ✓ should return 400 if userId is missing
    ✓ should return 400 if expiresAt is invalid
    ✓ should return 500 if there is an error creating the ban
    ✓ should return 500 if refreshing ban state fails

```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	99.54	97.57	100	99.54	
bans	100	99.58	100	100	
route.test.ts	100	99.25	100	100	42
route.ts	100	100	100	100	
bans/route.test	99.47	93.44	100	99.47	
GET.test.ts	100	100	100	100	
PATCH.test.ts	98.37	88	100	98.37	174,195
POST.test.ts	100	94.44	100	100	39

(ii). Reports :

```

✓ src/app/api/moderation/reports/route.payload.test.ts (2)
  ✓ Moderation Reports payloads (2)
    ✓ under_review: sets resolved_at/resolved_by to null and trims resolution_note
    ✓ dismissed: sets resolved_at/resolved_by and allows null resolution_note
✓ src/app/api/moderation/reports/route.test.ts (11)
  ✓ Moderation Reports API (11)
    ✓ GET /api/moderation/reports (5)
      ✓ 401 unauthorized when not logged in
      ✓ 403 forbidden when user not moderator
      ✓ filters by valid status and returns reports
      ✓ 500 when reports query fails
      ✓ ignores invalid status filter and returns all
    ✓ PATCH /api/moderation/reports (6)
      ✓ 401 unauthorized when not logged in
      ✓ 403 forbidden when user not moderator
      ✓ 400 reportId is required
      ✓ 400 Invalid status
      ✓ success updates report with resolved fields for terminal statuses
      ✓ 500 update error returns failure

```

```

✓ src/app/api/moderation/reports/route.test/GET.test.ts (4)
  ✓ GET() moderation reports (4)
    ✓ should return reports when authorized moderator
    ✓ should return 401 when user is not authenticated
    ✓ should return 403 when user is not a moderator
    ✓ should return 500 when loading reports fails
  ✓ PATCH /api/moderation/reports/route.test/PATCH.test.ts (6)
    ✓ PATCH() moderation reports (6)
      ✓ should update report successfully
      ✓ should return 401 when user is unauthorized
      ✓ should return 403 when user is not a moderator
      ✓ should return 400 when reportId is missing
      ✓ should return 400 when status is invalid
      ✓ should return 500 when update fails
Test Files  8 passed (8)
Tests      143 passed (143)

```

reports	98.69	95.58	100	98.69	
route.payload.test.ts	100	88.88	100	100	24
route.test.ts	100	100	100	100	
route.ts	96.62	94.73	100	96.62	122-124
reports/route.test	98.63	95.12	100	98.63	
GET.test.ts	98.05	94.73	100	98.05	76,86
PATCH.test.ts	99.13	95.45	100	99.13	78

IV. Payment :

(i). Create-order

```
✓ src/app/api/payments/create-order/route.test.ts (17)
✓ POST /api/payments/create-order (100% White Box Coverage) (17)
✓ should return 500 if Razorpay credentials are not configured
✓ should only initialize Razorpay once (normalization check)
✓ should return 401 Unauthorized if auth fails
✓ should return 400 if communityId or amount is missing
✓ should return 400 if amount is not a positive number
✓ should return 404 if community not found in DB
✓ should return 400 if user is already an active member
✓ should successfully create order and return ID
✓ should throw error if Razorpay returns an invalid/empty order object
✓ should truncate receipt to <= 40 characters
✓ should round amount to integer paise (e.g., 99.999 -> 10000)
✓ should include notes: community_id, user_id, community_name
✓ should return 500 with descriptive error on razorpay API failure
✓ should return 500 for 401 status code (invalid credentials)
✓ should re-throw and be caught by general catch for unhandled Razorpay errors
✓ should return 500 for unhandled exceptions (e.g., JSON parsing failure)
✓ should include error stack in response if NODE_ENV is development
```

(ii). Verify

```
✓ src/app/api/payments/verify/route.test.ts (22)
✓ POST /api/payments/verify (22)
✓ 401 if user not logged in
✓ 400 if required fields missing
✓ 400 if razorpay_order_id or razorpay_payment_id missing
✓ 400 invalid payment signature
✓ 400 invalid signature with extra fields noise
✓ 400 when razorpay_order_id is not a string
✓ 404 if community does not exist
✓ existing member -> update membership
✓ existing member update error -> 500
✓ new member -> inserts membership
✓ new member insert error -> 500
✓ catch block -> 500
✓ existing member update returns null data -> treated as success
✓ new member insert succeeds but select().single() returns null -> still success
✓ calls from() with 'communities' table
✓ calls from() with 'community_members' table
✓ calls select() with '*' for lookups
✓ crypto.createHmac called with 'sha256' algorithm
✓ hmac.update called with correct text format 'order_id|payment_id'
✓ hmac.digest called with 'hex' encoding
✓ update() called with correct status fields for existing member
✓ insert() called with correct fields for new member

Test Files 2 passed (2)
Tests 39 passed (39)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	99.7	97.52	100	99.7	
create-order	99.4	97.05	100	99.4	
route.test.ts	100	100	100	100	
route.ts	98.21	94.87	100	98.21	90-91
verify	100	98.11	100	100	
route.test.ts	100	100	100	100	
route.ts	100	95.83	100	100	100

V. Reports :

```
✓ src/app/api/reports/route.test.ts (20)
✓ POST /api/reports (20)
✓ 401 if auth fails
✓ 400 invalid JSON payload
✓ 400 missing reportedUserId
✓ 400 cannot report yourself
✓ 400 invalid category
✓ 500 Unable to verify reported user (profile error)
✓ 404 Reported user not found
✓ 500 Could not submit report (insert error)
✓ 200 success with trimmed description and normalized evidence
✓ description blank becomes null
✓ 500 generic unexpected error (non-RequestError)
✓ calls from() with exact table name "profiles" for lookup
✓ calls from() with exact table name "user_reports" for insert
✓ calls select() with "id" for profile lookup
✓ insert payload has correct structure
✓ evidenceIds with non-string items are filtered
✓ evidenceIds undefined becomes empty array
✓ evidenceIds non-array becomes empty array
✓ description empty string becomes null
✓ 401 when user is null with no error

✓ src/app/api/reports/route.test.POST.test.ts (8)
✓ POST /api/reports (8)
✓ should successfully submit a report
✓ should return 401 if user is unauthorized
✓ should return 400 if JSON payload is invalid
✓ should return 400 if reportedUserId is missing
✓ should return 400 if user tries to report themselves
✓ should return 400 if report category is invalid
✓ should return 404 if reported user is not found
✓ should return 500 if there is an error inserting the report
```



```
Test Files 2 passed (2)
Tests      28 passed (28)
Start at   11:57:16
Duration   3.61s (transform 172ms, setup 422ms, collect 481ms, tests 41ms, environment 10ms)
```

HTML Report is generated

You can run `npx vite preview --outDir html` to see the test results.

% Coverage report from v8

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
reports	100	100	100	100	
route.test.ts	100	100	100	100	
route.ts	100	100	100	100	
reports/route.test	100	100	100	100	
POST.test.ts	100	100	100	100	

2. Communities :

I. CreateCommunity

```
✓ src/app/communities/actions.test.ts (94)
✓ createCommunity (15)
  ✓ returns error if user not logged in
  ✓ requires verified creator
  ✓ rejects empty name
  ✓ invalid slug results in error
  ✓ rejects paid community with no price
  ✓ insertion failure returns DB error
  ✓ successful creation with free community
  ✓ successful creation with paid community
  ✓ returns error when slug already exists
  ✓ rejects paid community with zero price
  ✓ rejects paid community with negative price
  ✓ queries profiles table for verification
  ✓ queries communities table for slug check
  ✓ inserts into communities table
  ✓ inserts into community_members table for owner
```

II. JoinCommunity

```
✓ joinCommunity (14)
✓ requires login
✓ community not found with error
✓ community not found when null data
✓ already active member
✓ pending membership returns error
✓ rejoin if previously left
✓ rejoin update failure returns error
✓ insert active member for free community
✓ insert pending member for paid community
✓ insert error
✓ queries communities table first
✓ queries community_members table for membership check
✓ inserts into community_members table
✓ community error with non-null data still returns error
```

III. leaveCommunity

```
✓ leaveCommunity (9)
✓ requires login
✓ owner cannot leave
✓ update failure
✓ successful leave returns left: true
✓ successful leave with no slug still revalidates /communities
✓ community not found returns success (graceful handling)
✓ queries communities table for owner check
✓ updates community_members table
✓ updates with status: left
```

IV. removeCommunityMember

```
✓ removeCommunityMember (27)
  ✓ requires login
  ✓ community not found with error
  ✓ community not found with null data
  ✓ non-owner non-co-owner cannot remove
  ✓ actor membership error returns error
  ✓ actor membership not found returns error
  ✓ actor with inactive status cannot remove
  ✓ actor with pending status cannot remove
  ✓ cannot remove self
  ✓ target not found with error
  ✓ target not found with null data
  ✓ target with non-active status returns member not found
  ✓ target with pending status returns member not found
  ✓ cannot remove community owner
  ✓ co-owner cannot remove another co-owner
  ✓ co-owner can remove member
  ✓ co-owner can remove admin
  ✓ admin update fails
  ✓ successful removal returns removed: true
  ✓ queries communities table first
  ✓ queries community_members for target lookup (as owner)
  ✓ queries community_members for actor lookup (as non-owner)
  ✓ admin updates community_members table
  ✓ update payload includes status, left and role: member
  ✓ owner can always remove (even without actor membership)
  ✓ moderator cannot remove members
  ✓ admin role cannot remove members
```

V. DeleteCommunity

```
✓ deleteCommunity (0)
  ✓ requires login
  ✓ community not found with error
  ✓ community not found with null data
  ✓ must be owner
  ✓ delete error
  ✓ successful delete returns deleted: true
  ✓ queries communities table for ownership check
  ✓ deletes from communities table
  ✓ error with non-null data still returns community not found
```

VI. UpdateMemberRole

```
✓ updateMemberRole (20)
  ✓ requires login
  ✓ community not found with error
  ✓ community not found with null data
  ✓ only owner allowed
  ✓ cannot modify self
  ✓ target member not found with error
  ✓ target member not found with null data
  ✓ cannot promote owner
  ✓ must be active member
  ✓ pending member cannot be promoted
  ✓ update fails
  ✓ successful role change returns updated: true
  ✓ demote co-owner to member
  ✓ queries communities table for ownership
  ✓ queries community_members for target
  ✓ admin updates community_members table
  ✓ update payload includes role
  ✓ update payload with member role
  ✓ error with non-null data still returns community not found
  ✓ member error with non-null data still returns member not found
```

```
Test Files 1 passed (1)
Tests 94 passed (94)
Start at 11:59:12
Duration 3.09s (transform 146ms, setup 149ms, collect 139ms, tests 84ms, environment 745ms, prepare 802ms)
```

HTML Report is generated

You can run `npx vite preview --outDir html` to see the test results.

% Coverage report from v8

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
actions.test.ts	100	100	100	100	
actions.ts	100	100	100	100	

Components Folder:

1. Analytics

```
✓ src/components/analytics/engagement-chart.test.tsx (1)
  ✓ EngagementChart (1)
    ✓ renders the base chart structure with all data series
✓ src/components/analytics/page.test.tsx (1)
  ✓ AnalyticsPage (1)
    ✓ renders inside the Appshell and wires up child components
✓ src/components/analytics/stats-card.test.tsx (2)
  ✓ StatsCard (2)
    ✓ renders title, value, and default upward trend styling
    ✓ applies downward styling when trendDirection is down

Test Files 3 passed (3)
Tests 4 passed (4)
Start at 12:04:03
Duration 4.86s (transform 156ms, setup 617ms, collect 4.22s, tests 143ms, environment 2.66s, prepare 1.88s)

🔗 Report is generated
You can run npx vite preview --outDir html to see the test results.

% Coverage report from vitest
-----
File                                % Stmts % Branch % Funcs % Lines Uncovered Line #s
-----
All Files                          92.24    100     100    100
components/analytics
engagement-chart.test.tsx          100     100    100    100
engagement-chart.tsx               100     100    100    100
page.test.tsx                      100     100    100    100
page.tsx                           100     100    100    100
stats-card.test.tsx                100     100    100    100
stats-card.tsx                     100     100    100    100
components/ui
card.test.tsx                      78.94    100    100    78.94 51-56,71-76
card.tsx                           78.94    100    100    78.94
lib                                 60      100     50     60
utils.ts                           60      100     50     60 15-20
-----
```

2. Auth

```
✓ src/components/auth/auth-forms.test.tsx (9) 4253ms
  ✓ LoginForm (3) 1233ms
    ✓ shows validation errors on empty submit
    ✓ shows toast on failed login 61ms
    ✓ redirects on successful login 548ms
  ✓ SignupForm (2) 1816ms
    ✓ shows toast on failed signup 905ms
    ✓ redirects on successful signup 911ms
  ✓ ForgotPasswordForm (2) 421ms
    ✓ shows toast on failure
    ✓ success toast show
  ✓ UpdatePasswordForm (2) 782ms
    ✓ shows toast on update fail 404ms
    ✓ success redirect 378ms
✓ src/components/auth/social-buttons.test.tsx (2)
  ✓ SocialButtons (2)
    ✓ shows the correct CTA copy for signup mode
    ✓ calls Supabase OAuth with the Next callback path

Test Files 2 passed (2)
Tests 11 passed (11)
Start at 12:05:10
Duration 8.67s (transform 201ms, setup 325ms, collect 1.68s, tests 4.40s, environment 1.66s, prepare 1.25s)

🔗 Report is generated
You can run npx vite preview --outDir html to see the test results.

% Coverage report from vitest
-----
File                                % Stmts % Branch % Funcs % Lines Uncovered Line #s
-----
All Files                          89.82    86.66    83.87    89.82
components/auth
auth-forms.test.tsx               99.46    100     100    99.46 50
forgot-password-form.test.tsx      91.25    72.72    100    91.25 46,78-83
login-form.test.tsx               94.11    61.53    71.42    94.11 91-93,111-115
signup-form.test.tsx              90.4     83.33    100    90.4 52-58,85-89
social-buttons.test.tsx           100     100     100    100
social-buttons.tsx                100     100     100    100
update-password-form.test.tsx      93.61    88.88    100    93.61 74-79
components/ui
alert.tsx                         86.36    100     100    86.36 39-44
button.tsx                        100     50     100    100 45
dialog.tsx                        46.34    100     0    46.34 23-29,57-66,71-80,88-96,103-108
form.tsx                          89.1     86.66    100    89.1 52-53,132-142
input.tsx                         100     100     100    100
label.tsx                         100     100     100    100
lib                                 60      100     50     60
utils.ts                           60      100     50     60 15-20
-----
```

3. Communication

I. Community List

```
✓ src/components/communities/communities-list.mutation.test.tsx (12) 1699ms
  ✓ CommunitiesList - Mutation Killing Suite (12) 1699ms
    ✓ shows correct empty state when no communities are found 355ms
    ✓ shows search-specific empty state
    ✓ renders communities in the 'All' tab
    ✓ filters communities by free membership
    ✓ filters communities by paid membership
    ✓ 'My' tab shows only communities the user has joined
    ✓ shows empty message in 'My' tab when no membership match
    ✓ search filters communities correctly
    ✓ handles successful join of a free community
    ✓ handles join error
    ✓ handles successful leave
    ✓ handles leave error

✓ src/components/communities/communities-list.test.tsx/CommunitiesList.test.tsx (9) 910ms
  ✓ CommunitiesList (Mock Architecture Version) (9) 910ms
    ✓ Happy Path Tests (3) 459ms
      ✓ renders all communities 330ms
      ✓ joins a community successfully
      ✓ leaves a community successfully
    ✓ Edge Case Tests (6) 454ms
      ✓ handles empty community list
      ✓ handles search query filtering
      ✓ handles joinCommunity error
      ✓ shows 'no communities found' when list is empty
      ✓ renders paid community label and join button
      ✓ handles leaveCommunity error
```

II. Community Actions

```
✓ src/components/communities/community-actions.mutation.test.tsx (17) 844ms
✓ CommunityActions mutation killers (17) 844ms
  ✓ returns null when user is owner
  ✓ shows Leave button when user is member (not owner)
  ✓ shows Join button for non-member free community
  ✓ shows Join (Paid) button for paid community
  ✓ opens payment modal on paid community join click
  ✓ calls joinCommunity and shows success toast for free community
  ✓ shows error toast when joinCommunity fails
  ✓ calls leaveCommunity and shows success toast
  ✓ shows error toast when leaveCommunity fails
  ✓ calls router.refresh on payment success callback
  ✓ closes payment modal via close button
  ✓ does not render PaymentModal for free community
  ✓ does not show payment modal for paid community without price
  ✓ renders correct Leave button text and icon
  ✓ renders correct Join button text for free community
  ✓ renders correct Join button text for paid community
  ✓ handles paid community join click when price is set

✓ src/components/communities/community-actions.test/CommunityActions.test.tsx (6) 415ms
✓ CommunityActions (6) 415ms
  ✓ Happy Paths (3)
    ✓ renders Join button for non-member in free community
    ✓ renders Leave button for member
    ✓ opens payment modal for paid community
  ✓ Edge Cases (3)
    ✓ handles join community error
    ✓ handles leave community error
    ✓ renders nothing for owners
```

III. Community comment section

```
✓ src/components/communities/community-comments-section.mutation.test.tsx (15) 377ms
✓ CommunityCommentsSection - Mutation Killing Tests (15) 376ms
  ✓ fetches and displays comments
  ✓ handles fetch error by clearing comments
  ✓ shows loading state initially
  ✓ posts a comment successfully
  ✓ handles API insert validation error
  ✓ handles unexpected thrown error during submit
  ✓ toggles reply mode
  ✓ shows loading text while submitting
  ✓ handles null parents array by treating it as empty list
  ✓ does not submit when comment is empty
  ✓ uses fallback error message when e.message is missing
  ✓ toggles reply Off when clicking Reply again
  ✓ renders fallback avatar initial for current user when display_name missing
  ✓ renders fallback avatar initial when author has no display_name
  ✓ fires onCountChange with comment count

✓ src/components/communities/community-comments-section.test/CommunityCommentsSection.test.tsx (8) 332ms
✓ CommunityCommentsSection - Full Mutation Coverage (8) 330ms
  ✓ renders loading + empty state
  ✓ fetches and displays comment with metadata
  ✓ allows posting a new comment
  ✓ does not post empty comment
  ✓ handles fetch error safely
  ✓ handles posting error with toast
  ✓ allows replying to a comment and shows reply UI
  ✓ renders nested replies correctly
```

IV. Community Member List

```
✓ src/components/communities/community-members-list.mutation.test.tsx (18) 451ms
✓ CommunityMembersList - Mutation Killing Suite (18) 448ms
  ✓ renders fallback initial + fallback name when display_name and username missing
  ✓ renders ONLY display_name when username is missing
  ✓ renders initial from display_name
  ✓ renders username inside span wrapper
  ✓ renders owner badge
  ✓ renders co-owner badge
  ✓ renders no badge for regular member
  ✓ falls back to user_id when username missing
  ✓ shows 'Hi more members' when more than 20 members exist
  ✓ promotes member + co-owner
  ✓ demotes co-owner + member
  ✓ shows toast when updateMemberRole returns error
  ✓ disables toggle button in pending state
  ✓ hides role toggle when management is disabled
  ✓ helper functions (4)
    ✓ covers admin & moderator
    ✓ getRoleIcon returns Shield for admin/moderator
    ✓ default branches return UserCheck icon + outline badge
    ✓ returns Crown icon for owner and co-owner

✓ src/components/communities/community-members-list.test/CommunityMembersList.test.tsx (4) 361ms
✓ CommunityMembersList() (4)
  ✓ Happy Paths (2)
    ✓ should render the list of community members correctly
    ✓ should call updateMemberRole when role is changed
  ✓ Edge Cases (2)
    ✓ should display +1 more when more than 20 members
    ✓ should show toast on error from updateMemberRole
```

VI. Community Member Manager

```
✓ src/components/communities/community-members-manager.mutation.test.tsx (47) 954ms
✓ CommunityMembersManager - Mutation Killing Suite (19) 539ms
  ✓ renders empty state when no members exist
  ✓ renders fallback name & avatar initials correctly
  ✓ owner can remove a regular member
  ✓ co-owner cannot remove another co-owner
  ✓ owner cannot remove themselves
  ✓ owner can toggle co-owner role
  ✓ toggle co-owner shows error on failure
  ✓ removal error shows toast
  ✓ displays Updating... state when updating role
  ✓ enters removing pending state when starting removal
  ✓ does NOT allow removal when viewerRole is null
  ✓ does NOT allow removing yourself
  ✓ renders outline badge for regular member
  ✓ does NOT allow co-owner to toggle roles
  ✓ does NOT allow toggling owner role
  ✓ does NOT allow user to toggle their own role
  ✓ disables toggle button while updating role
  ✓ does not render username when user.username is missing
  ✓ does not show remove button for inactive members

✓ src/components/communities/community-members-manager.test/CommunityMembersManager.test.tsx (4) 361ms
✓ CommunityMembersManager (4)
  ✓ allows owners to remove members but not themselves
  ✓ blocks owners from removing other owners
  ✓ allows co-owners to remove regular members only
  ✓ returns false when viewer has no role
```

```

✓ CommunityMembersManager - RHL Mutation Killing Suite (24) 412ms
  ✓ renders empty state with no members
  ✓ renders 'Unknown user' when no display name or username present
  ✓ renders username fallback when display_name missing
  ✓ renders 'Owner' badge for owner
  ✓ renders 'Co-owner' badge for co-owner
  ✓ renders no badge for regular member
  ✓ owner CAN toggle member = co-owner
  ✓ co-owner CANNOT toggle roles
  ✓ viewerRole=null CANNOT toggle roles
  ✓ viewerRole='member' CANNOT toggle roles
  ✓ owner CANNOT toggle owner
  ✓ owner CANNOT toggle own role
  ✓ shows toast on role update error
  ✓ owner CAN remove member
  ✓ viewerRole=null cannot remove
  ✓ viewerRole='member' cannot remove
  ✓ co-owner cannot remove co-owner
  ✓ co-owner CANNOT remove owner
  ✓ owner CANNOT remove owner
  ✓ owner CANNOT remove themselves
  ✓ cannot remove member if status != active
  ✓ shows toast on removal error
  ✓ shows Updating... while updating role
  ✓ shows Removing... during removal pending state

```

```

✓ src/components/communities/community-members-manager.test/CommunityMembersManager.test.tsx (7) 797ms
  ✓ CommunityMembersManager() (7) 793ms
    ✓ Happy Path Tests (3) 717ms
      ✓ should render members correctly
      ✓ should allow owner to toggle co-owner role
      ✓ should allow owner to remove a member 394ms
    ✓ Edge Case Tests (4)
      ✓ should not allow co-owner to remove another co-owner
      ✓ should handle empty members list
      ✓ renders correct fallback name when user has no display_name
      ✓ should not allow owner to remove themselves

```

VII. Community Post Card

```

✓ src/components/communities/community-post-card/mutation.test.tsx (17) 864ms
✓ CommunityPostCard - Mutation Killing Suite (17) 861ms
  ✓ renders formatted time when created_at is present
  ✓ renders fallback avatar initial
  ✓ renders correct initial from display_name
  ✓ displays premium lockout when user is not premium
  ✓ does NOT display lockout when user IS premium
  ✓ renders post text when canViewContent
  ✓ renders image media
  ✓ renders video media
  ✓ renders hashtags
  ✓ prevents like when user not logged in
  ✓ likes the post successfully
  ✓ unlikes the post successfully
  ✓ handles like error
  ✓ toggles comments visible
  ✓ disables comments when premium locked
  ✓ does not drop like count below zero when unliking
  ✓ updates comment count when child reports new value

```

```

✓ src/components/communities/community-post-card.test/CommunityPostCard.test.tsx (5) 489ms
✓ CommunityPostCard() (5) 482ms
  ✓ Happy Paths (5) 423ms
    ✓ renders core content
    ✓ increments like count when clicked
  ✓ Edge Cases (3)
    ✓ show premium message for non-premium users
    ✓ shows fallback avatar initial when display_name is missing
    ✓ shows toast if user is not logged in when liking

```

VIII. Community Card

```

✓ src/components/communities/community-card.test/CommunityCard.test.tsx (6)
✓ CommunityCard (Mock Architecture Version) (6)
  ✓ Happy Paths (3)
    ✓ renders community details
    ✓ calls onJoin for free communities
    ✓ calls onLeave for members
  ✓ Edge Cases (3)
    ✓ disables join button for inactive communities
    ✓ opens payment modal for paid communities
    ✓ refreshes page after payment success

```

IX. Create Community Form

```

✓ src/components/communities/create-community-form/mutation.test.tsx (9) 2264ms
✓ CreateCommunityForm - Mutation Killing Tests (9) 2262ms
  ✓ renders the VERIFIED gating UI when user is not verified 418ms
  ✓ renders the price field when membership type is Paid 302ms
  ✓ submits successfully with valid data 359ms
  ✓ handles API validation failure
  ✓ handles unexpected thrown error
  ✓ shows loading state during submit
  ✓ requires price when switching to paid membership
  ✓ keeps submit button disabled for unverified users
  ✓ navigates to the new community slug after success

  ✓ should show toast on error from updateMembership

✓ src/components/communities/create-community-form.test/CommunityForm.test.tsx (4) 816ms
✓ CreateCommunityForm (Mock Architecture Version) (4) 816ms
  ✓ renders form fields 355ms
  ✓ submits successfully
  ✓ shows error when name is too short
  ✓ handles API error

```

X. Create Community Post

```

✓ src/components/communities/create-community-post/mutation.test.tsx (10) 828ms
✓ CreateCommunityPost - Mutation Killing Suite (10) 826ms
  ✓ blocks posting when canPost = false
  ✓ shows error when submitting with no text or media
  ✓ submits successfully with text only
  ✓ uploads media files and posts
  ✓ handles backend insert validation error
  ✓ catches unexpected thrown errors
  ✓ shows loading state while posting
  ✓ requires authentication before posting
  ✓ blocks members without owner privileges
  ✓ falls back to default error message when exception lacks message

  ✓ should show toast on error from updateMembership

✓ src/components/communities/create-community-post.test/CreateCommunityPost.test.tsx (8) 1703ms
✓ CreateCommunityPost - Full Working Suite (8) 1708ms
  ✓ renders UI correctly 383ms
  ✓ blocks posting when canPost = false
  ✓ requires text or media
  ✓ handles not-logged-in user
  ✓ denies posting for non-owner/co-owner
  ✓ submits text-only posts
  ✓ handles upload error 771ms
  ✓ resets form after success

```

XI. Payment Modal

```
✓ src/components/communities/payment-modal.mutation.test.tsx (20) 1661ms
  ✓ PaymentModal mutation killers (20) 1658ms
    ✓ does not render content when closed
    ✓ renders dialog content when open
    ✓ shows error toast when Razorpay key is missing
    ✓ shows error toast when create-order API fails
    ✓ shows error toast when response is invalid
    ✓ opens Razorpay checkout on valid order
    ✓ calls onSuccess and onClose after successful verify
    ✓ shows error toast when verify fails
    ✓ Cancel button calls onClose
    ✓ shows error toast when razorpayloaded is false
    ✓ displays community name in dialog description
    ✓ displays formatted amount in dialog
    ✓ disables Pay button when loading
    ✓ shows verification error toast when verify throws exception
    ✓ calls setLoading(false) when modal dismissed
    ✓ uses default error message when API error is missing
    ✓ uses default currency INR when not provided
    ✓ sends correct payload to create-order API
    ✓ uses default error message when verify error is missing
    ✓ shows community membership label in dialog
```

```
✓ src/components/communities/payment-modal.test/paymentModal.test.tsx (35) 2615ms
  ✓ PaymentModal() (35) 2613ms
    ✓ renders PaymentModal correctly when open
    ✓ calls onClose when Cancel is clicked
    ✓ initiates Razorpay when Pay Now is clicked
    ✓ shows configuration error toast when Razorpay key is missing
    ✓ shows toast when Razorpay throws error
    ✓ shows toast when Razorpay is not loaded
    ✓ shows payment error when creating order fails
    ✓ shows Payment Failed toast when verification fails
    ✓ completes payment flow on success
    ✓ calls create-order API with exact endpoint and correct payload
    ✓ calls verify API with exact endpoint and correct payload
    ✓ passes correct options object to Razorpay constructor
    ✓ shows error when orderId is missing from create-order response
    ✓ shows error when amount is missing from create-order response
    ✓ uses INR as default currency when not provided
    ✓ handles verification network error
    ✓ calls onSuccess callback when Razorpay modal is dismissed
    ✓ uses default error message when server error is undefined
    ✓ uses default error message when verification error is undefined
    ✓ uses default error message when verification throws error without message
    ✓ uses default error message when payment throws error without message
    ✓ manages loading state during payment flow
    ✓ runs useEffect when isOpen changes from false to true
    ✓ checks for window object in useEffect
    ✓ updates razorpayloaded state when isOpen changes
    ✓ sets loading to false after Razorpay opens successfully
    ✓ sets loading to false when Razorpay modal is dismissed
    ✓ sets loading to false when verification fails
    ✓ sets loading to false when verification throws error
    ✓ sets loading to false when payment throws error
    ✓ logs to console.error when payment fails
    ✓ logs to console.error when verification fails
    ✓ enables Pay Now button when Script loads
    ✓ shows error toast when Script fails to load
    ✓ shows Processing... and disables button during payment
```

```
Test Files 19 passed (19)
Tests 257 passed (257)
Start at 12:06:13
Duration 22.70s (transform 1.18s, setup 4.98s, collect 55.12s, tests 18.53s, environment 24.19s, prepare 6.85s)
```

HTML Report is generated
You can run `npm vite preview --outDir html` to see the test results.

% Coverage report from v8

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	98.86	96.71	86.62	98.86	
app/communities	0.32	100	0	0.32	
actions.ts	0.32	100	0	0.32	7-806
components/communities	99.42	96.69	94.78	99.42	
communities-list.mutation.test.tsx	100	100	100	100	
communities-list.tsx	100	97.50	100	100	63
community-actions.mutation.test.tsx	100	100	100	100	
community-actions.tsx	100	100	100	100	
community-card.tsx	100	95.65	80	100	97
community-comments-section.mutation.test.tsx	99.34	96.55	100	99.34	56-76
community-comments-section.tsx	100	97.91	85.71	100	84
community-members-list.mutation.test.tsx	100	100	100	100	
community-members-list.tsx	100	92	100	100	71-72
community-members-manager.mutation.test.tsx	100	100	100	100	
community-members-manager.tsx	98.24	88.88	100	98.24	61,69-70
community-post-card.mutation.test.tsx	100	100	100	100	
community-post-card.tsx	97.33	95.52	100	97.33	75-78
create-community-form.mutation.test.tsx	100	100	100	100	
create-community-form.tsx	100	81.33	90	100	50,87-89,110
create-community-post.mutation.test.tsx	98.61	96.96	100	98.61	28,53-54
create-community-post.tsx	99.46	87.72	62.5	99.46	177
payment-modal.mutation.test.tsx	99.41	100	100	99.41	121
payment-modal.tsx	96.17	96.96	100	96.17	57-63
components/communities/communities-list.test	100	100	100	100	
CommunitiesList.test.tsx	100	100	100	100	
components/communities/community-actions.test	100	100	100	100	
CommunityActions.test.tsx	100	100	100	100	
components/communities/community-card.test	100	100	100	100	
CommunityCard.test.tsx	100	100	100	100	
components/communities/community-comments-section.test	100	100	100	100	
CommunityCommentsSection.test.tsx	100	100	100	100	
components/communities/community-members-list.test	100	100	100	100	
CommunityMembersList.test.tsx	100	100	100	100	
components/communities/community-members-manager.test	100	100	100	100	
CommunityMemberManager.test.tsx	100	100	100	100	
components/communities/community-post-card.test	100	100	100	100	
CommunityPostCard.test.tsx	100	100	100	100	
components/communities/create-community-form.test	100	86.66	100	100	
CreateCommunityForm.test.tsx	100	86.66	100	100	39,47
components/communities/create-community-post.test	98.12	96.87	100	98.12	
CreateCommunityPost.test.tsx	98.12	96.87	100	98.12	68-71

components/communities/payment-modal.test	99.7	100	100	99.7	866,899,921
PaymentModal.test.tsx	99.7	100	100	99.7	
components/ui	95.85	94.11	100	95.85	
alert-dialog.tsx	93.75	100	100	93.75	92-97
avatar.tsx	100	100	100	100	
badge.tsx	100	100	100	100	
button.tsx	100	50	100	100	45
card.tsx	89.47	100	100	89.47	51-56
dialog.tsx	100	100	100	100	
form.tsx	98.01	87.5	100	98.01	52-53
input.tsx	100	100	100	100	
label.tsx	100	100	100	100	
select.tsx	89.09	100	100	89.09	106-111,141-146
tabs.tsx	100	100	100	100	
textarea.tsx	100	100	100	100	

4. Feed:

I. Comment Section:

src/components/feed/comments-section.test.tsx (32) 2193ms	
CommentsSection (32) 2193ms	
Initial Rendering (4)	
renders the component	
shows loading state initially	
shows empty state when no comments	
displays comment count	
Fetching Comments (5)	
fetches and displays parent comments	
fetches and displays replies	
fetches like status for comments	
handles fetch error and logs to console	
sets comments to empty array on error	
Posting Comments (9) 1200ms	
posts a new comment	
increments comment count after posting	
clears textarea after posting	
does not submit empty or whitespace-only comment	
does not submit comment without authentication	
handles comment submission error with object containing message	
handles comment submission error with non-object error	
disables post button when textarea is empty	
shows 'Posting...' during submission	
Reply Functionality (5) 300ms	
shows reply input when clicking Reply	
cancels reply mode	
toggles reply mode off when clicking Reply again	
posts reply with parent_id	
does not show Reply button for replies	
Like Functionality (5)	
likes a comment	
requires authentication to like	
handles like toggle error	
displays like count correctly	
shows singular 'Like' for count of 1	
Comment Count Callback (1)	
calls onCommentCountChange when comments are loaded	
Auth States (3)	
does not show comment input without auth	
shows comment input with auth	
disables like button without auth	

II. Create Post Modal

src/components/feed/create-post-modal.test.tsx (71) 12605ms	
CreatePostModal (71) 12605ms	
Rendering (4)	
renders when open	
does not render when closed	
renders textarea placeholder	
renders all action buttons	
Text Input (3) 1120ms	
updates text when typing 300ms	
extracts hashtags from text 524ms	
displays hashtag badges	
Media Upload (5)	
renders file input	
accepts multiple files	
accepts images and videos	
displays image preview after selection	
can remove uploaded media	
Location (1)	
updates location input	
Visibility Setting (3)	
defaults to public visibility	
can change visibility to followers	
can change visibility to private	
Collaborator Search (4) 548ms	
searches profiles when typing	
displays search results	
adds collaborator when clicked	
does not search with query less than 2 characters	
Post Creation (9) 2135ms	
is disabled when no content	
is enabled when text is entered	
creates post with text	
shows success toast on post creation	
calls onPostCreated callback	
closes modal after post creation	
includes hashtags in post 435ms	
includes location in post 408ms	
sets is-private based on visibility	
Error Handling (2) 618ms	
shows error toast when post creation fails 321ms	
does not close modal on error	
Loading State (1)	
disables post button while loading	

- ✓ CreatePostModal - Additional Coverage (16) 4918ms
 - ✓ Video Upload (3)
 - ✓ handles video file upload
 - ✓ handles mixed image and video uploads
 - ✓ filters out non-image/video files
 - ✓ Media Upload to Storage (3) 1105ms
 - ✓ uploads media to storage bucket 425ms
 - ✓ includes media URI in post data 325ms
 - ✓ handles storage upload error 355ms
 - ✓ Collaboration Invites (4) 2250ms
 - ✓ sends collaboration invites when collaborators are selected 767ms
 - ✓ handles collaboration invite error gracefully 581ms
 - ✓ creates multiple collaboration invites 756ms
 - ✓ removes collaborator from selection
 - ✓ Dialog Close Handler (2) 320ms
 - ✓ prevents closing when loading 314ms
 - ✓ allows closing when not loading
 - ✓ Form Reset After Post (1) 453ms
 - ✓ resets all form fields after successful post 453ms
 - ✓ Authentication Check (1)
 - ✓ throws error when user is not authenticated
 - ✓ Edge Cases (2) 405ms
 - ✓ handles empty media files array
 - ✓ trims whitespace from text
- ✓ CreatePostModal - Final Coverage (13) 2814ms
 - ✓ Storage Bucket Error (1)
 - ✓ handles getBucketError error
 - ✓ Collaborator Display (3) 325ms
 - ✓ displays username when display_name is present
 - ✓ displays unknown when display_name is null
 - ✓ displays username when no display_name exists
 - ✓ Collaborator Results Dropdown (3) 725ms
 - ✓ hides results when query is empty
 - ✓ hides results after selecting a collaborator
 - ✓ does not show dropdown when collaborator results are empty 411ms
 - ✓ getInitials Helper (1)
 - ✓ handles null name by returning ?
 - ✓ handles empty string name by returning ?
 - ✓ uppercases the first character
 - ✓ Collaborator Badge Username Display (2)
 - ✓ displays username in badge when both username and display_name exist
 - ✓ displays display_name in badge when no username
 - ✓ Excluded Collaborators in Search (1)
 - ✓ filters out already selected collaborators from search results

- ✓ Uncovered functions (18) 347ms
 - ✓ triggers file input click on image button
 - ✓ triggers file input click on video button
 - ✓ handles smile emoji button click
 - ✓ processes FileReader onload callback
 - ✓ handles dialog onChange
 - ✓ covers remove media button click
 - ✓ covers remove collaborator X button
 - ✓ covers searchProfiles with query less than 2 chars
 - ✓ covers handleMediaSelect with no files
 - ✓ covers all arrow functions in JSX

III. Create post:

- ✓ src/components/feed/create-post-test.tsx (24) 947ms
 - ✓ CreatePost (24) 944ms
 - ✓ Rendering (6)
 - ✓ renders the create post card
 - ✓ displays user avatar
 - ✓ displays avatar fallback when no avatar URL
 - ✓ displays U fallback when no user data
 - ✓ renders Photo/Video button
 - ✓ renders Thread button
 - ✓ User Fetching (3)
 - ✓ fetches user on mount
 - ✓ handles user fetch error gracefully
 - ✓ updates avatar when user data loads
 - ✓ Modal Opening (4)
 - ✓ opens modal with post tab when clicking main button
 - ✓ opens modal with post tab when clicking Photo/Video button
 - ✓ opens modal with thread tab when clicking thread button
 - ✓ modal is closed by default
 - ✓ Modal Closing (2)
 - ✓ closes modal when close button is clicked
 - ✓ can reopen modal after closing
 - ✓ Post Creation (2)
 - ✓ refreshes page when post is created
 - ✓ logs message when post is created
 - ✓ Button Variants and Styling (3)
 - ✓ main button has correct variant
 - ✓ Photo/Video button has ghost variant
 - ✓ thread button has ghost variant
 - ✓ Tab Switching (2)
 - ✓ switches to thread tab correctly
 - ✓ maintains correct tab when reopening
 - ✓ Edge Cases (2)
 - ✓ handles user with no email
 - ✓ handles undefined user_metadata

IV. Engagement Action

```
✓ src/components/feed/engagement-actions.mutation.test.tsx (3) 376ms
✓ EngagementActions mutation killers (3) 374ms
  ✓ shows login-required toast when unauthenticated like attempted
  ✓ increments then decrements like without going negative
  ✓ early returns and shows already shared toast when share exists
✓ src/components/feed/engagement-actions.test.tsx (40) 1095ms
✓ EngagementActions (40) 1093ms
  ✓ renders all action buttons
  ✓ displays correct like count
  ✓ displays correct comment count
  ✓ displays correct share count
  ✓ displays correct save/bookmark count
  ✓ displays singular 'Like' when count is 1
  ✓ displays singular 'Comment' when count is 1
  ✓ displays singular 'Share' when count is 1
  ✓ displays singular 'Save' when count is 1
  ✓ shows login toast when liking without auth
  ✓ shows login toast when bookmarking without auth
  ✓ shows login toast when sharing without auth
  ✓ checks user engagement on mount
  ✓ indicates when post is already liked
  ✓ indicates when post is bookmarked
  ✓ likes a post and updates count
  ✓ unlikes a post and decreases count
  ✓ bookmarks a post and updates count
  ✓ shares a post and updates count
  ✓ prevents duplicate shares with existing share toast
  ✓ handles like error gracefully
  ✓ handles bookmark error gracefully
  ✓ handles share error gracefully
  ✓ calls onCommentClick when comment button is clicked
  ✓ calls onEngagementChange after liking
  ✓ calls onEngagementChange after bookmarking
  ✓ calls onEngagementChange after sharing
  ✓ fetches and displays post counts on mount
  ✓ handles missing engagement check when no currentUserid
  ✓ disables buttons while loading
  ✓ handles default counts when post has no counts
  ✓ copies post URL to clipboard on share
  ✓ handles fetch error in checkUserEngagement
  ✓ handles fetch error in fetchCounts
  ✓ handles share fetch error gracefully
  ✓ handles checkUserEngagement error
  ✓ handles like error with non-object error
  ✓ handles bookmark error with non-object error
  ✓ handles share error with non-object error
  ✓ removes bookmark when already bookmarked
```

V. Feed List:

```
✓ src/components/feed/feed-list.test.tsx (24) 477ms
✓ feedList (24) 476ms
  ✓ Initial Rendering (4)
    ✓ renders initial posts
    ✓ passes currentUserid to Postcard components
    ✓ displays posts in correct order
    ✓ renders empty state when no posts
  ✓ Infinite Scroll Setup (3)
    ✓ creates IntersectionObserver on mount
    ✓ observes the last post element
    ✓ attaches ref to last post only
  ✓ Loading More Posts (4)
    ✓ fetches more posts when last post intersects
    ✓ displays loading skeleton while fetching
    ✓ appends new posts to existing list
    ✓ calculates correct range for page 2
  ✓ Deduplication (2)
    ✓ filters out duplicate posts
    ✓ only adds unique posts from new data
  ✓ End of Feed (4)
    ✓ sets hasMore to false when less than PAGE_SIZE posts returned
    ✓ displays end of feed message
    ✓ does not show end message when posts array is empty
    ✓ stops fetching when hasMore is false
  ✓ Error Handling (2)
    ✓ logs error when fetch fails
    ✓ does not crash when fetch returns error
  ✓ Loading State (2)
    ✓ prevents multiple simultaneous loads
    ✓ hides loading skeleton after fetch completes
  ✓ creates new observer when posts change
  ✓ Edge Cases (2)
    ✓ handles empty author data
    ✓ skips fetch on initial render (page 0)
```

VI. Follow Button:

```
✓ src/components/feed/follow-button.test.tsx (19) 208ms
✓ Followbutton (19) 208ms
  ✓ does not render a button for the profile owner
  ✓ renders follow when not following
  ✓ shows login toast when clicked while not logged in
  ✓ follows a public profile (inserts into followers)
  ✓ sends a follow request for private profiles (inserts into follow_requests)
  ✓ unfollows an already-followed user (deletes from followers)
  ✓ cancels a pending follow request (deletes from follow_requests)
  ✓ calls onFollowChange after the setTimeout delay 524ms
  ✓ handles errors when checking follow status (checkFollowStatus catch)
  ✓ shows error toast when follow insert fails
  ✓ does not render button when currentUserid is the same as targetUserId
  ✓ disables button and shows loading indicator when isLoading is true
  ✓ does not insert duplicate follow request if one already exists
  ✓ shows login toast when clicked while not logged in
  ✓ shows error toast if unfollow delete fails
  ✓ uses correct button variant based on follow status
  ✓ shows error toast with error message on follow insert failure
  ✓ sets followStatus to pending and calls onFollowChange after follow request sent 492ms
  ✓ resets followStatus to none and calls onFollowChange after unfollow or cancel 492ms
```

VII. Follow Request:

```
✓ src/components/feed/follow-requests.test.tsx (17) 839ms
✓ FollowRequests (17) 836ms
  ✓ renders loading state initially
  ✓ renders empty state when no requests
  ✓ renders follow requests list
  ✓ accepts a single request
  ✓ declines a single request
  ✓ accepts all requests
  ✓ handles fetch error
  ✓ does not render Accept All button when no requests
  ✓ does not render Accept All button for single request
  ✓ shows error toast if delete fails during accept
  ✓ shows error toast if insert fails during accept
  ✓ shows error toast if insert fails during accept all
  ✓ shows error toast if delete fails during accept all
  ✓ displays singular 'request' text for single pending request
  ✓ displays plural 'requests' text for multiple pending requests
  ✓ calls onRequestUpdate callback after accepting a request
  ✓ calls onRequestUpdate callback after accepting all requests
```

VIII. Friend Suggestion:

```
✓ src/components/feed/friend-suggestions.test.tsx (10) 525ms
  ✓ FriendSuggestions (10) 522ms
    ✓ shows loading skeletons initially
    ✓ shows nothing if no suggestions
    ✓ displays user suggestions if present
    ✓ displays username in fallback if display_name missing
    ✓ handles suggestion fetch error gracefully
    ✓ displays 'U' fallback when both display_name and username are missing
    ✓ displays 'U' fallback when username is empty string
    ✓ uppercases the fallback initial
    ✓ renders profile link correctly
    ✓ does not fetch suggestions or render when currentUserId is empty
```

IX. Post Card:

```
✓ src/components/feed/post-card.mutation.test.tsx (10) 390ms
  ✓ PostCard - mutation killers (10) 388ms
    ✓ renders post with author name and time ago
    ✓ falls back to 'recently' when date parsing fails
    ✓ renders collaborators popover when collaborators present
    ✓ shows comments section when engagement action clicked
    ✓ displays media image when media_url is present
    ✓ returns null after delete
    ✓ handles archive error and shows toast
    ✓ handles delete error and shows toast
    ✓ skips actions when not owner
    ✓ renders archived post state
  ✓ src/components/feed/post-card.test.tsx (12) 983ms
    ✓ PostCard (12) 980ms
      ✓ Basic Rendering (5)
        ✓ renders post with text content
        ✓ renders author information
        ✓ renders time ago
        ✓ renders avatar with correct image
        ✓ renders avatar fallback when no image
      ✓ Author Display (2)
        ✓ displays 'Unknown User' when author has no display_name
        ✓ links to author profile
      ✓ Time Display (2)
        ✓ handles invalid date gracefully
        ✓ handles null created_at
      ✓ Media Display (1)
        ✓ renders image media
        ✓ renders video media
        ✓ does not render media section when no media
      ✓ Collaborators (7)
        ✓ does not show collaborators icon when no collaborators
        ✓ shows collaborators icon when collaborators exist
        ✓ filters out non-accepted collaborators
        ✓ displays collaborator names in popover
        ✓ handles collaborators without display_name
        ✓ handles null collaborators array
        ✓ handles empty accepted collaborators
      ✓ Comments Toggle (4)
        ✓ does not show comments section initially
        ✓ shows comments section when engagement action clicked
        ✓ hides comments section when clicked again
        ✓ passes correct props to CommentsSection
      ✓ User Authentication (2)
        ✓ fetches current user on mount
        ✓ handles no authenticated user
      ✓ getInitials Helper (4)
        ✓ returns first character uppercase
        ✓ returns '?' for null name
        ✓ returns '?' for undefined name
        ✓ returns '?' for empty string
```

```
✓ More Options Button (1)
  ✓ renders more options button
✓ PostCard - handlers and collaborators (2)
  ✓ fetches and displays collaborators
  ✓ handles null or empty collaborators gracefully
```

X. Stories Reaction:

```
✓ src/components/feed/stories-reactions.test.tsx (19) 712ms
  ✓ StoriesReactions (19) 710ms
    ✓ Initial Rendering (2)
      ✓ renders all reaction buttons
      ✓ does not fetch reactions without currentUserId
    ✓ Fetching User Reaction (2)
      ✓ fetches user reaction on mount
      ✓ handles no existing user reaction
    ✓ Fetching Reaction Counts (3)
      ✓ displays reaction counts
      ✓ shows singular 'reaction' for count of 1
      ✓ handles error fetching reaction counts
    ✓ Adding Reactions (2)
      ✓ adds a new reaction
      ✓ shows user reaction badge after reacting
      ✓ shows login prompt when not authenticated
    ✓ Removing Reactions (2)
      ✓ removes existing reaction when clicked again
      ✓ decrements count when removing reaction
    ✓ Changing Reactions (2)
      ✓ changes from one reaction to another
      ✓ updates counts when changing reactions
    ✓ Error Handling (2)
      ✓ handles delete error
      ✓ handles upsert error
    ✓ Callback Functionality (1)
      ✓ calls onReactionChange after adding reaction
    ✓ Loading State (1)
      ✓ disables buttons while loading
    ✓ Data Returned from Upsert (1)
      ✓ uses data from upsert response when available
```

XI. Stories:

✓ src/components/feed/stories.test.tsx (48) 1038ms					
✓ Stories (48) 1036ms					
✓ Initial Rendering (3)					
✓ renders stories component					
✓ fetches current user on mount					
✓ fetches stories after getting user					
✓ Story Display (5)					
✓ shows upload button when user has no stories					
✓ shows user's own story when exists					
✓ renders followed users' stories					
✓ shows loading spinner while fetching					
✓ shows 'no stories' when empty and user is logged in					
✓ Avatar Rendering (4)					
✓ renders avatar image when available					
✓ renders avatar fallback when no image					
✓ handles owner story with no avatar					
✓ handles story with undefined profile					
✓ Story Click and Modal (7)					
✓ opens modal when clicking on story					
✓ displays story author in modal					
✓ displays story media in modal					
✓ renders video in modal for video stories					
✓ displays story text in modal					
✓ does not render text section when story has no text					
✓ renders reactions component in modal					
✓ Story Deletion (6)					
✓ shows delete button for own story					
✓ does not show delete button for others' stories					
✓ deletes story successfully					
✓ handles database deletion error					
✓ handles storage deletion error gracefully					
✓ prevents deleting others' stories					
✓ Story Upload (1)					
✓ calls fetchStories after upload					
✓ Error Handling (2)					
✓ handles fetch stories error					
✓ handles no current user					
✓ Story Border Colors (2)					
✓ shows primary border for active stories					
✓ shows muted border for expired stories					
✓ Author Info Edge Cases (2)					
✓ handles story with null author avatar					
✓ handles story with null author display name					
✓ Story Sorting (1)					
✓ displays owner's story first					
✓ Edge Cases - Full Coverage (15) 305ms					
✓ does not fetch stories when user ID is undefined					
✓ handles fetchStories returning null/undefined					
✓ handles story with no author object					
✓ closes modal when onOpenChange is called with falsy value					
✓ handles profile with null display name					
✓ refreshes stories after reaction change					
✓ extracts file path correctly from storage URL					
✓ handles story with undefined expires_at					
✓ handles empty stories array from fetch					
✓ handles stories with mixed ownership					
✓ sets loading to false after fetch completes					
✓ handles deletion when selectedStory is null					
✓ handles deletion attempt on non-owned story					
✓ handles storage URL without stories bucket					
✓ prevents deleting someone else's story and shows error toast					
Files: 14 passed (14)					
Tests: 93 passed (93)					
File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
all files	93.57	94.75	80.82	93.57	
components/feed	97.10	94.76	87.21	97.10	
components-section.test.tsx	98.52	98.95	100	98.52	1113-1121,1438-1446
comments-section.tsx	96.82	77.63	100	96.82	172-177,183-187
create-post-modal.test.tsx	99.85	99.47	100	99.85	1907-1908
create-post-modal.tsx	97.24	96.13	95	97.24	110-125
create-post-mutation.test.tsx	100	100	100	100	
create-post-test.tsx	100	100	100	100	
create-post.tsx	100	100	100	100	
engagement-actions.mutation.test.tsx	100	100	65.51	100	
engagement-actions.test.tsx	92.13	89.71	100	92.13893-939-940,942-943,1002-1003,1005-1006,1049-1050,1076-1077,1109-1110,1112-1113,1151-1152,1185-1186,1188-1189,1233-1234,1236-1237,1282-1283,1285-1286,1330-1331,1333-1334
engagement-actions.tsx	96.68	96.53	100	96.68	150-150
feed-list.test.tsx	97.44	97.05	85.71	97.44	39-40,58-66
feed-list.tsx	100	100	100	100	
follow-button.test.tsx	100	98.55	100	100	58
follow-button.tsx	98.25	89.83	100	98.25	74-76,78-80
follow-requests.test.tsx	98.25	91.42	100	98.25	120,162,221,305,350,405,463,569,624
follow-requests.tsx	100	92.3	100	100	47,91,168
friend-suggestions.test.tsx	100	100	100	100	
friend-suggestions.tsx	100	94.11	100	100	91
post-card.mutation.test.tsx	85.08	86.2	100	85.08	38-44,105-108,117-119,129-131
post-card.test.tsx	100	100	100	100	
post-card.tsx	66.66	87.73	57.14	66.66	91-115,118-141,144-168
stories-reactions.test.tsx	99.55	100	100	99.55	336-338
stories-reactions.tsx	100	94.23	100	100	40,105,125
stories.test.tsx	100	100	100	100	
stories.tsx	93.89	91.93	83.33	93.89	42-44,71-75,91-93,207-208
components/ui	72.02	92.85	50	72.02	
avatar.tsx	100	100	100	100	
badge.tsx	100	100	100	100	
button.tsx	100	50	100	100	45
card.tsx	100	100	100	100	
dropdown-menu.tsx	35.55	100	0	35.55	27-30,47-55,83-92,99-115,123-138,147-156,163-168,172-181
popover.tsx	100	100	100	100	

5. Messages

I. Chat Layout

```
src/components/messages/chat-layout.test.tsx (64) 6126ms
✓ ChatLayout (64) 6124ms
  ✓ renders conversations from the API and opens the detail view 371ms
  ✓ shows fallback UI and toast when conversations fail to load
  ✓ displays a blocked alert when the conversation is blocked by the current user
  ✓ sends a message via Supabase when messaging is allowed 516ms
  ✓ rolls back delete UI when the API fails
  ✓ renders message attachments and hides broken images
  ✓ shows deleted message indicators for removed messages
  ✓ shows a preview for selected images and allows removing it
  ✓ shows a 'blocked you' alert when the other user blocked the current user
  ✓ shows an error toast when a non-image file is selected
  ✓ shows an error toast when image file exceeds 10MB
  ✓ allows inserting emoji into the message input
  ✓ allows clearing chat for current user
  ✓ deletes the conversation for everyone
  ✓ blocks the other user 310ms
  ✓ unblocks a previously blocked user
  ✓ navigates to the user profile on 'View profile'
  ✓ deletes a message for everyone
  ✓ renders date separators for messages
  ✓ initializes isChatVisible=true when URL contains userID or conversationId
  ✓ scrolls to bottom after messages appear
  ✓ handles fetchConversations error path
  ✓ clears userID and conversationId from URL
  ✓ updates conversation flags
  ✓ shows toast when opening conversation fails
  ✓ handles fetchMessages error
  ✓ creates preview for valid image
  ✓ uploads image successfully
  ✓ closes emoji picker on emoji click
  ✓ sends message when only image is selected
  ✓ auto selects conversation by conversationId param
  ✓ fetches conversations when realtime adds one
  ✓ ignores onMessageInserted when different conversation
  ✓ ignores invalid onMessageUpdated payload
  ✓ renders 'You deleted this message' when deleted by includes current user
  ✓ applies muted time label on deleted for everyone messages
  ✓ disables send when no message and no image
  ✓ does nothing when sending with empty message and no image
  ✓ stops sending if image upload fails
  ✓ shows error toast when database insert fails
  ✓ handles unexpected send exception and shows toast
  ✓ throws when submitting report with no current participant
  ✓ throws when report API returns error
  ✓ submits report successfully
  ✓ returns empty preview when no last_message
  ✓ returns 'You deleted this message' when deleted by contains current user
  ✓ opens conversation and closes new chat dialog onChatStarted
  ✓ displays unread count badge when >0
```

```
✓ treats message as deleted for user when deleted by includes them
✓ renders italic deleted message fallback
✓ disables send button when sending or empty
  ✓ updates conversation last_message when realtime update matches
  ✓ updates a single message inside messages list when realtime update arrives
  ✓ selects conversation when URL matches existing conversation
  ✓ adds new message on realtime INSERT
  ✓ updates selectedConversation.last_message when realtime update matches its last_message
  ✓ blocks sending when conversation is blocked by me
  ✓ shows not delivered toast when other user blocked me
  ✓ does nothing on empty send
  ✓ captures failureMessage on report exception
  ✓ truncates preview text longer than 72 chars
  ✓ skips test-only scroll lock effect when NODE_ENV != test
  ✓ renders italic deleted message when no content and no image
  ✓ disables send button when isSending or empty
```

II. NewChatDialog:

```
src/components/messages/new-chat-dialog.test.tsx (17) 1146ms
✓ NewChatDialog() UPDATED MOCK ARCHITECTURE (17) 1143ms
  ✓ renders dialog when isOpen is true
  ✓ performs search and displays results
  ✓ calls onChatStarted when user is clicked
  ✓ displays "No users found." when no results
  ✓ handles search errors gracefully
  ✓ does not search when term length is exactly 1
  ✓ searches when term length is exactly 2
  ✓ clears results when search term becomes too short
  ✓ filters moderators when current user is not moderator
  ✓ shows moderators when current user is moderator
  ✓ handles user with null display name
  ✓ does not search when user is not logged in
  ✓ handles null profile gracefully
  ✓ logs error to console when search fails
  ✓ updates search input value on change
  ✓ starts with loading set to false initially
  ✓ shows "No users found" only when length > 1, not >= 1
```

III. Report User Dialog:

```
src/components/messages/report-user-dialog.test.tsx (5) 1170ms
✓ ReportUserDialog() UPDATED ARCHITECTURE (5) 1168ms
  ✓ renders the dialog with initial values
  ✓ submits the form with valid data
  ✓ shows an error when category is missing
  ✓ shows an error when description is less than 10 characters 326ms
  ✓ handles onSubmit rejection gracefully
src/components/messages/report-user-dialog.test.tsx (8) 1774ms
✓ ReportUserDialog() (8) 1772ms
  ✓ renders dialog with initial values
  ✓ renders all category options 128ms
  ✓ submits form with valid data 311ms
  ✓ shows error when category missing
  ✓ shows error when description is too short
  ✓ accepts description with exactly 10 characters
  ✓ filters out empty evidence URLs
  ✓ handles onSubmit rejection gracefully
Test Files 4 passed (4)
Tests 94 passed (94)
```

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	95.87	82.28	80.89	95.87	
components/messages	94.83	79.56	79.45	94.83	
chat-layout.test.tsx	99.82	99.29	99	99.82	144-145
chat-layout.tsx	89.92	70.57	76.66	89.92	...4,508-510,522-524,531,535-537,545-547,554,678-681,723-734,744-745,755-756,760-768,773-786,828,850-851,855-856,860-861,873,888-889,893-894,900,944-945,1434,1508-1510
new-chat-dialog.tsx	100	96.55	100	100	55
report-user-dialog.tsx	98.17	92.1	75	98.17	100-102
components/messages/new-chat-dialog.test	100	100	100	100	
NewChatDialog.test.tsx	100	100	100	100	
components/messages/report-user-dialog.test	100	100	100	100	
ReportUserDialog.test.tsx	100	100	100	100	
async.test.tsx	100	100	100	100	
components/ui	97.58	94.73	100	97.58	
alert-dialog.tsx	100	100	100	100	
alert.tsx	100	100	100	100	
avatar.tsx	100	100	100	100	
badge.tsx	100	100	100	100	
button.tsx	100	66.66	100	100	45
dialog.tsx	100	100	100	100	
input.tsx	100	100	100	100	
label.tsx	100	100	100	100	
popover.tsx	100	100	100	100	
scroll-area.tsx	95.96	75	100	95.96	38

6. Moderation

I. Moderator Dashboard:

✓ src/components/moderation/moderator-dashboard.mutation.test.tsx (0) 504ms					
✓ ModeratorDashboard - mutation killers (0) 502ms					
✓ shows loading state then renders reports					
✓ shows error toast when reports fail					
✓ shows error toast when bans fail					
filters reports by status					
selects a report and shows detail panel					
renders ban list and handles lift ban					
✓ shows empty state when no reports match filter					
✓ shows setuprequired state for bans when flagged					
✓ src/components/moderation/moderator-dashboard.test.tsx (22) 4234ms					
✓ ModeratorDashboard (22) 4232ms					
✓ dismisses a report when the moderator saves the status 459ms					
✓ issues a ban for the selected report and hides the form afterward					
✓ shows an error when bans fail to load and retries on demand					
✓ hides the ban form when the user is already suspended					
✓ shows toast and leaves reports empty when loading reports fails					
filters reports based on selected status					
✓ saves report status with override and resolution note 685ms					
✓ shows error toast when updating report status fails					
✓ shows error when custom ban date is selected but empty 387ms					
✓ shows error when issuing ban fails					
✓ shows toast when lifting ban fails					
✓ shows empty state when no report is selected					
renders evidence URLs when provided					
✓ hides ban form after issuing ban due to justbanned/serId					
✓ resets report status override, note and ban errors when switching reports 351ms					
✓ clears justBanned/serId after bans update					
✓ shows all reports when filter = all					
✓ renders report details when a report is selected					
✓ renders evidence URLs when present					
✓ renders banner when reportee already banned					
✓ submits custom date when banPreset = custom 613ms					
✓ shows empty-state message when no report is selected					
Test Files 2 passed (2)					
Tests 30 passed (30)					

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	96.3	86.4	80	96.3	
components/moderation	97.61	83.69	81.25	97.61	
moderator-dashboard.mutation.test.tsx	98.11	92.1	77.77	98.11	58,130
moderator-dashboard.test.tsx	100	94.68	77.77	100	112,133-139
moderator-dashboard.tsx	95.89	76.97	83.71	95.89	199-200,208-209,220-222,286-287,435,458-460
components/ui	93.5	93.75	100	93.5	
badge.tsx	100	100	100	100	
button.tsx	100	50	100	100	45
card.tsx	89.47	100	100	89.47	71-76
input.tsx	100	100	100	100	
label.tsx	100	100	100	100	
select.tsx	89.09	100	100	89.09	106-111,141-146
textarea.tsx	100	100	100	100	
lib	60	100	50	60	
utils.ts	60	100	50	60	15-20

7. Notification

I. Collab Invite

✓ src/components/notifications/collab-invites.mutation.test.tsx (9) 547ms					
✓ CollabInvites - mutation killers (0) 565ms					
✓ shows loading then empty message when no invites					
✓ returns early and shows nothing if user is null					
✓ renders invite cards with inviter name and post snippet					
✓ clicking Accept updates post collaborators and marks invite accepted					
✓ clicking Decline marks invite declined and removes from list					
✓ shows error toast when accept fails due to post fetch error					
✓ shows error toast when decline update fails					
✓ handles inviter with null display name and falls back to username					
✓ handles post with null text and falls back to post id					
✓ src/components/notifications/collab-invites.test/CollabInvites.test.tsx (6) 429ms					
✓ CollabInvites() UPDATED MOCK ARCHITECTURE (0) 426ms					
✓ renders loading state initially					
✓ shows 'no invites' message when none exist					
✓ renders invites and handles ACCEPT successfully					
✓ handles DECLINE successfully					
✓ handles ACCEPT error correctly					
✓ handles DECLINE error correctly					

II. Notification Bell

```
✓ src/components/notifications/notification-bell.test.tsx (21) 1177ms
✓ NotificationBell() UPDATED ARCHITECTURE (21) 1174ms
✓ renders bell with no unread notifications
✓ displays correct unread count
✓ shows 99+ if count > 99
✓ subscribes to notifications on mount
✓ skips fetching if user is missing
✓ handles real-time notifications correctly
✓ refreshes unread count on notifications:refresh event
✓ unsubscribes on unmount
✓ removes event listener on unmount
✓ displays exactly 99+ for unread count of 99
✓ displays exactly 99+ for unread count of 100
✓ does not render badge when unread is 0
✓ renders badge when unread > 0
✓ uses toast callback in handleRealtimeNotification
✓ uses default title when notification has no title
✓ uses undefined description when notification has no body
✓ checks globalThis and window existence for SSR safety
✓ verifies optional chaining for removeEventListener
✓ skips event listener setup in SSR environment
✓ returns early from SSR guard when conditions fail
✓ handles window being undefined during cleanup
```

```
✓ src/components/notifications/notification-bell.mutation.test.tsx (3)
✓ NotificationBell - mutation killers (3)
✓ renders zero badge when no unread
✓ shows badge after refresh and opens popover
✓ shows toast when realtime notification arrives
```

III. Notification Item

```
✓ src/components/notifications/notification-item.test.tsx (20) 740ms
✓ NotificationItem Component - Updated Architecture (20) 737ms
✓ renders correct notification details
✓ marks notification as read
✓ accepts collaboration invite
✓ handles error when marking as read fails
✓ handles RPC error on invite acceptance
✓ declines collaboration invite
✓ shows declined status after declining invite
✓ shows accepted status after accepting invite
✓ disables buttons while loading
✓ shows body text when body is provided
✓ does not show body when body is null
✓ updates isRead state when item is read changes
✓ displays avatar fallback when avatar_url is null
✓ handles actor with null display_name
✓ handles actor with null username and display_name
✓ marks notification as read when already read
✓ handles error message in mark as read
✓ logs error to console on RPC error
✓ shows collab buttons only when status is pending
✓ handles missing metadata invite_id gracefully
```

IV. Notification List

```
✓ src/components/notifications/notification-list.mutation.test.tsx (3) 406ms
✓ NotificationList - mutation killers (3) 402ms
✓ renders empty state when no notifications
✓ renders items and mark-all-read succeeds
✓ handles mark-all-read failure without crashing
```

```
✓ src/components/notifications/notification-list.test.tsx (24) 1449ms
✓ NotificationList() Updated Test Architecture (24) 1445ms
✓ renders 5 skeletons while loading
✓ shows "no notifications" message when empty list is returned
✓ renders fetched notifications
✓ marks all notifications as read when button clicked
✓ handles realtime notifications via subscribeNotifications()
✓ handles missing user gracefully
✓ dispatches notifications:refresh when mark all as read
✓ calls unsubscribe on unmount
✓ adds realtime notification when initial fetch is null
✓ marks items visually as read after Mark all
✓ verifies all items have is_read: true after mark all
✓ preserves exact order when realtime notification is prepended
✓ verifies handleRealtime is called in useEffect with correct dependencies
✓ handles empty prev array when realtime notification arrives
✓ runs in client environment with valid globalThis.window
✓ verifies notifyThreadRefresh checks both globalThis and window existence
✓ spreads notification correctly when adding to empty prev
✓ maintains handleRealtime callback stability across renders
✓ transforms all items to is_read: true after mark all
✓ returns empty object mutation detection for map result
✓ detects LogicalOperator mutation in SSR guard
✓ verifies is_read property is set to true (not false) after mark all
✓ verifies mark all preserves all notification properties (not empty object)
✓ SSR guard requires both globalThis.underlined check AND window existence

Test Files 7 passed (7)
Tests 86 passed (86)
```

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	88.67	95.1	68.18	88.67	
components/notifications	98.68	92.05	100	98.68	
collab-invites.mutation.test.tsx	99.19	96.47	100	99.19	59
collab-invites.tsx	96.42	81.57	100	96.42	31-32,37-38
notification-bell.mutation.test.tsx	100	100	100	100	
notification-bell.tsx	96.22	94.73	100	96.22	46-47
notification-item.tsx	100	93.11	100	100	31,39,67
notification-list.mutation.test.tsx	100	100	100	100	
notification-list.tsx	100	95.23	100	100	55
components/notifications/collab-invites.test	99.47	92.59	100	99.47	
collab-invites.test.tsx	99.47	92.59	100	99.47	72
components/notifications/notification-bell.test	100	100	100	100	
NotificationBell.test.tsx	100	100	100	100	
components/notifications/notification-item.test	98.95	100	75	98.95	
NotificationItem.test.tsx	98.95	100	75	98.95	36-37
components/notifications/notification-list.test	99.7	97.36	100	99.7	
NotificationList.test.tsx	99.7	97.36	100	99.7	213
components/ui	91.66	90	100	91.66	
avatar.tsx	100	100	100	100	
button.tsx	100	50	100	100	45
card.tsx	78.94	100	100	78.94	51-56,71-76
skeleton.tsx	100	100	100	100	

8. Privacy

```
src/components/privacy/privacy-settings.test.tsx (28) 1077ms
✓ PrivacySettings (28) 1075ms
  ✓ Loading State (2)
    ✓ displays loading message while fetching settings
    ✓ fetches privacy settings on mount
  ✓ Rendering After Load (2)
    ✓ renders privacy settings card
    ✓ displays all setting sections
  ✓ Profile Visibility Setting (4)
    ✓ displays current profile visibility
    ✓ updates profile visibility to private
    ✓ sets allow_follow_requests to false when visibility is public
    ✓ sets allow_follow_requests to true when visibility is followers
  ✓ Post Visibility Setting (2)
    ✓ updates post visibility
    ✓ displays correct description for post visibility
  ✓ Switch Settings (4)
    ✓ toggles show_online_status
    ✓ toggles show_followers
    ✓ toggles show_following
    ✓ toggles allow_tagging
  ✓ Direct Messages Setting (2)
    ✓ updates allow_direct_messages
    ✓ can set direct messages to none
  ✓ Mentions Setting (1)
    ✓ updates allow_mentions
  ✓ Success Toast (2)
    ✓ shows success toast after updating setting
    ✓ calls onSettingsUpdate callback after save
  ✓ Error Handling (1)
    ✓ shows error toast when fetch fails
    ✓ does not show error for POSTitis (no rows found)
    ✓ shows error toast when update fails
  ✓ Loading and Saving States (1)
    ✓ disables controls while saving
  ✓ Visibility Icons and Descriptions (3)
    ✓ displays correct icon for public visibility
    ✓ displays correct description for private visibility
    ✓ displays correct description for followers visibility
  ✓ Edge Cases (2)
    ✓ handles undefined userId
    ✓ includes updated_at timestamp in upsert

Test Files 1 passed (1)
Tests 28 passed (28)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	88.22	100	68.18	88.22	
components/privacy	100	100	100	100	
privacy-settings.test.tsx	100	100	100	100	
privacy-settings.tsx	100	100	100	100	
hooks	12.97	100	0	12.97	
use-toast.ts	12.97	100	0	12.97	30-33,62-75,78-130,137-192,196-199

9. Profile

I. Follow List

```
src/components/profile/follow-list.test.tsx (41) 1688ms
✓ FollowList (41) 1685ms
  ✓ renders loading skeleton initially
  ✓ renders empty state when no users found
  ✓ renders followers list correctly
  ✓ renders following list correctly
  ✓ does not render FollowButton for current user
  ✓ renders FollowButton for other users
  ✓ handles fetch error gracefully
  ✓ avoids duplicate users when pagination loads
  ✓ sets correct query parameters for followers
  ✓ sets correct query parameters for following
  ✓ renders avatar with fallback when avatar url is not provided
  ✓ uses username as fallback when display_name is not provided
  ✓ handles empty display_name and username gracefully
  ✓ handles private profile correctly
  ✓ calls order with ascending=false for newest first
  ✓ applies range pagination correctly (page 0)
  ✓ renders 'No following found' for following type with no users
  ✓ handles undefined currentUserId
  ✓ displays user avatar image when provided
  ✓ stops loading more when less than PAGE_SIZE items returned
  ✓ continues loading more when PAGE_SIZE items returned
  ✓ handles error in finally block correctly
  ✓ filters out existing users when appending new page
  ✓ displays correct count text format for multiple users
  ✓ renders last user element for Intersection observer
  ✓ resets pagination when userId changes
  ✓ renders isLoadingMore skeleton at bottom when paginating
  ✓ shows 'You've reached the end' message when all pages loaded and users > PAGE_SIZE
  ✓ does not show 'You've reached the end' when users <= PAGE_SIZE
  ✓ lastUserElement returns early when isLoading is true
  ✓ lastUserElement returns early when isLoadingMore is true
  ✓ IntersectionObserver.disconnect is called when ref changes
  ✓ IntersectionObserver triggers setPage when isIntersecting and hasNext
  ✓ handles data as null when error is present
  ✓ handles both error and data being null
  ✓ correctly sets isLoading for page 0 and isLoadingMore for page > 0
  ✓ filters duplicate users correctly with Set comparison
  ✓ handles type change from followers to following
  ✓ observer.current.observe is called with the correct node
  ✓ correctly maps and renders all users in list
  ✓ lastUserElement is only attached to last element in map
```

II. Post Click Handler

```
src/components/profile/PostClickHandler.test.tsx (27) 321ms
✓ PostClickHandler (27) 319ms
  ✓ Rendering (5)
    ✓ renders children content
    ✓ applies cursor-pointer class
    ✓ applies grid layout styles when post has media
    ✓ does not apply grid layout styles when post has no media
    ✓ does not apply grid layout styles when media is undefined
  ✓ Navigation - Posts Context (3)
    ✓ navigates to profile posts page on click
    ✓ uses default context (posts) when context is not specified
    ✓ disables scroll on navigation
  ✓ Navigation - Threads Context (1)
    ✓ navigates to profile posts page for threads context
  ✓ Navigation - Saved Context (2)
    ✓ navigates to dedicated post page for saved context
    ✓ uses different route format for saved posts
  ✓ Error Handling (4)
    ✓ does not navigate when post has no id
    ✓ does not navigate when post has no author
    ✓ does not navigate when author has no username
    ✓ logs error message when navigation fails validation
  ✓ Click Handling (2)
    ✓ handles multiple clicks
    ✓ calls router.push with correct parameters each time
  ✓ Different Post Structures (3)
    ✓ handles post with empty string id
    ✓ handles post with special characters in username
    ✓ handles post with UUID-format id
  ✓ Children Rendering (2)
    ✓ renders complex children elements
    ✓ renders nested components
  ✓ Context Variations (1)
    ✓ handles all three context values correctly
  ✓ Styling Edge Cases (3)
    ✓ applies grid styles for posts with single media item
    ✓ applies grid styles for posts with multiple media items
    ✓ does not apply grid styles for empty media array
  ✓ Partial Post Type (1)
    ✓ handles minimal post object
```

III. Profile Content Tab

```
src/components/profile/profile-content-tabs.test.tsx (38) 696ms
✓ ProfileContentTabs (38) 649ms
  ✓ Rendering (3)
    ✓ renders all four tabs
    ✓ renders tab labels
    ✓ displays posts tab content by default
  ✓ Posts Tab (9)
    ✓ displays posts in a grid
    ✓ renders post images when media is present
    ✓ renders placeholder when post has no media
    ✓ displays empty state when no posts
    ✓ shows 'View all posts' button when hasMorePosts is true
    ✓ does not show 'View all posts' button when hasMorePosts is false
    ✓ renders multiple posts
    ✓ handles posts with null text
    ✓ handles posts with missing media array
  ✓ Threads Tab (18)
    ✓ displays threads with author information
    ✓ displays thread metadata
    ✓ displays formatted timestamp
    ✓ renders author avatar
    ✓ displays empty state when no threads
    ✓ renders multiple threads
    ✓ handles threads with null counts
    ✓ handles threads without created_at
    ✓ renders avatar fallback when no avatar URL
    ✓ handles threads with missing author
  ✓ Saved Tab (4)
    ✓ renders SavedPostsTab component
    ✓ passes correct props to SavedPostsTab
    ✓ passes isActive=false when saved tab is not active
    ✓ passes correct savedPageSize prop
  ✓ Tagged Tab (5)
    ✓ displays tagged posts
    ✓ renders tagged post with image
    ✓ displays empty state when no tagged posts
    ✓ renders text fallback for tagged posts without media
    ✓ renders multiple tagged posts
  ✓ Props Handling (3)
    ✓ passes username to View all posts link
    ✓ passes isOwner to SavedPostsTab
    ✓ passes profileId to SavedPostsTab
  ✓ Edge Cases (2)
    ✓ handles empty arrays
    ✓ handles readonly array types
  ✓ Grid Layout (1)
    ✓ applies correct grid classes
  ✓ Responsive Design (1)
    ✓ renders mobile-hidden text spans
```


IV. Profile edit Form

```
✓ src/components/profile/profile-edit-form.test.tsx (29) 1888ms
✓ ProfileEditForm (29) 1886ms
  ✓ Rendering (4)
    ✓ renders the form with user data
    ✓ displays avatar preview
    ✓ shows avatar fallback when no avatar URL
    ✓ renders Save and Cancel buttons
  ✓ Form Input Changes (5) 1110ms
    ✓ updates display name when input changes
    ✓ updates username when input changes
    ✓ updates bio when textarea changes
    ✓ updates website when input changes 386ms
    ✓ updates location when input changes
  ✓ Avatar Upload (2)
    ✓ updates avatar preview when file is selected
    ✓ handles file upload with FileHeader
  ✓ Cancel Action (2)
    ✓ calls onCancel when Cancel button is clicked
    ✓ does not save changes when cancelled
  ✓ Save Action - Without Avatar Upload (4)
    ✓ saves profile changes successfully
    ✓ calls onSave callback with updated user data
    ✓ shows success toast after saving
    ✓ includes updated at timestamp in update
  ✓ Save Action - With Avatar Upload (3)
    ✓ uploads avatar and updates profile
    ✓ handles storage bucket error
    ✓ generates unique filename for avatar upload
  ✓ Error Handling (4)
    ✓ shows error toast when user is not authenticated
    ✓ handles username uniqueness constraint violation
    ✓ shows generic error for other database errors
    ✓ does not call onSave when save fails
  ✓ Loading State (3)
    ✓ disables buttons while saving
    ✓ shows "Saving..." text while saving
    ✓ re-enables buttons after save completes
  ✓ Edge Cases (2)
    ✓ handles user with empty string fields
    ✓ handles user with null fields
```

V. Profile header:

```
✓ src/components/profile/profile-header.test.tsx (51) 1008ms
✓ ProfileHeader - mutation killers (4)
  ✓ shows verified badge only when isVerified true
  ✓ shows Edit Profile button only for own profile
  ✓ renders follower/following counts for non-own profile
  ✓ falls back to first letter when avatar missing
✓ src/components/profile/profile-header.test.tsx (51) 1008ms
✓ ProfileHeader (51) 998ms
  ✓ Rendering (9)
    ✓ renders user profile information correctly
    ✓ displays verified badge for verified users
    ✓ does not display verified badge for non-verified users
    ✓ displays post, followers, and following counts
    ✓ renders default bio when bio is null
    ✓ displays mutual followers count when provided
    ✓ displays singular mutual follower when count is 1
    ✓ does not display mutual followers for own profile
    ✓ displays website link when provided
  ✓ Own Profile Actions (9) 339ms
    ✓ displays edit and settings buttons for own profile
    ✓ displays logout button for own profile
    ✓ does not display follow button for own profile
    ✓ handles logout action
    ✓ opens edit mode when edit button is clicked
    ✓ saves profile changes and exits edit mode
    ✓ redirects when username changes after save
    ✓ cancels edit mode without saving
    ✓ navigates to settings when settings button is clicked
  ✓ Other User's Profile Actions (6)
    ✓ displays follow button for other user's profile
    ✓ displays message button for other user's profile
    ✓ displays report button for other user's profile
    ✓ does not display edit, settings, or logout buttons
    ✓ handles message button click
    ✓ does not navigate to messages when currentUserId is undefined
    ✓ does not navigate to messages for own profile
    ✓ refreshes follower counts after follow action
  ✓ Banned User Profile (5)
    ✓ displays suspended button for banned user
    ✓ disables follow button for banned user
    ✓ disables message button for banned user
    ✓ still displays report button for banned user
    ✓ does not navigate to messages for banned user
  ✓ Report Functionality (7)
    ✓ opens report dialog when report button is clicked
    ✓ does not open report dialog for own profile
    ✓ shows toast when non-logged-in user tries to report
    ✓ submits report successfully
    ✓ handles report submission error without throwing
    ✓ handles network error during report submission without throwing
    ✓ closes report dialog when close button is clicked
  ✓ Follow List Modal (4)
    ✓ opens followers modal when followers count is clicked
    ✓ opens following modal when following count is clicked
    ✓ does not open followers modal when canViewFollowers is false
    ✓ does not open following modal when canViewFollowing is false
  ✓ Private Profile (1)
    ✓ disables follower/following counts when user cannot view them
  ✓ Error Handling (3)
    ✓ handles refresh counts error gracefully
    ✓ uses previous counts when API returns null follower_count
    ✓ uses previous counts when API returns null following_count
  ✓ Edge Cases (5)
    ✓ handles missing optional fields
    ✓ uses fallback avatar when avatar url is null
    ✓ renders edit form with null/undefined fields
    ✓ handles undefined is_private and defaults to false
    ✓ handles undefined isVerified and defaults to false
```

VI. Saved post Tab

```
✓ src/components/profile/saved-posts-tab.test.tsx (23) 367ms
  ✓ SavedPostsTab (23) 366ms
    ✓ Access Control (2)
      ✓ shows access denied message when not the owner
      ✓ does not fetch posts when not the owner
    ✓ Tab Activation (2)
      ✓ shows placeholder message when tab is not active and no posts loaded
      ✓ does not fetch posts when tab is not active
    ✓ Loading State (1)
      ✓ shows loading spinner while fetching posts
    ✓ Data Fetching (8)
      ✓ fetches saved posts with correct query parameters
      ✓ displays saved posts after successful fetch
      ✓ renders post images when media is present
      ✓ renders text fallback when post has no media
      ✓ uses correct page size from props
      ✓ only fetches once even if rendered multiple times
    ✓ Error Handling (2)
      ✓ displays error message when fetch fails
      ✓ sets posts to empty array on error
    ✓ Empty State (2)
      ✓ shows empty state when no saved posts
      ✓ shows empty state when data is null
    ✓ Post Validation (2)
      ✓ filters out invalid posts without required fields
      ✓ filters out null and non-object posts
    ✓ Post Context (1)
      ✓ passes 'saved' context to PostClickHandler
    ✓ Grid Layout (1)
      ✓ renders posts in a grid layout
    ✓ Edge Cases (1)
      ✓ handles posts with null text
      ✓ handles posts with empty media array
      ✓ handles posts with null media
    ✓ Re-activation Behavior (1)
      ✓ does not re-fetch when tab becomes active again after initial load
```

VII. Scroll to Post Client

```
✓ src/components/profile/ScrollToPostClient.test.tsx (15)
  ✓ ScrollToPostClient (15)
    ✓ Rendering (1)
      ✓ renders nothing (returns null)
    ✓ Scroll Behavior (5)
      ✓ scrolls to element when targetId is provided and element exists
      ✓ prepends 'post-' to the targetId
      ✓ does not scroll when element is not found
      ✓ does not scroll when targetId is empty string
      ✓ handles targetId with special characters
    ✓ useEffect Behavior (2)
      ✓ re-runs scroll effect when targetId changes
      ✓ does not re-run effect when targetId remains the same
    ✓ runs effect on mount with valid targetId
    ✓ Edge Cases (4)
      ✓ handles numeric targetId
      ✓ handles targetId with UUID format
      ✓ does not throw error when document.getElementById returns null
      ✓ handles rapid targetId changes
    ✓ scrollIntoView Options (2)
      ✓ uses smooth scroll behavior
      ✓ uses start block alignment

test Files 8 passed (8)
Tests 228 passed (228)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	92.05	96.11	71.79	92.05	
components/feed	33.12	41.17	66.66	33.12	
follow-button.tsx	33.12	41.17	66.66	33.12	29-30,48-64,73-158,162-163,165-166,178-193,205-207
components/messages	82.02	66.66	12.5	82.02	
report-user-dialog.tsx	82.02	66.66	12.5	82.02	95-97,100-102,184-207
components/profile	99.19	97.79	96	99.19	
PostClickHandler.test.tsx	100	100	100	100	
PostClickHandler.tsx	100	100	100	100	
ScrollToPostClient.test.tsx	100	100	100	100	
ScrollToPostClient.tsx	100	100	100	100	
follow-list.test.tsx	97.95	96.26	71.42	97.95	219,386-390,393,571,671-680,995
follow-list.tsx	100	100	100	100	
profile-content-tabs.test.tsx	100	100	100	100	
profile-content-tabs.tsx	100	96.77	100	100	177
profile-edit-form.test.tsx	99.6	100	100	99.6	71-72
profile-edit-form.tsx	92.02	92.1	100	92.02	70-71,83-89,93-99
profile-header-animation.test.tsx	100	100	100	100	
profile-header.test.tsx	100	100	100	100	
profile-header.tsx	99.23	91.25	100	99.23	126-127
saved-posts-tab.test.tsx	100	100	100	100	
saved-posts-tab.tsx	100	100	100	100	
components/ui	66.08	93.33	25	66.08	
avatar.tsx	100	100	100	100	
badge.tsx	83.33	100	0	83.33	39-34
button.tsx	100	50	100	100	45
card.tsx	89.47	100	100	89.47	71-76
dialog.tsx	46.34	100	0	46.34	21-29,57-66,71-80,88-96,103-108
input.tsx	100	100	100	100	
label.tsx	100	100	100	100	
select.tsx	24.54	100	100	24.54	19-32,39-49,56-66,74-86,106-111,118-134,141-146
skeleton.tsx	100	100	100	100	
textarea.tsx	100	100	100	100	
hooks	25.95	100	12.5	25.95	
use-toast.ts	25.95	100	12.5	25.95	30-33,62-75,78-130,137-172,196-199
lib	60	100	50	60	
utils.ts	60	100	50	60	15-20

10. Search

I. Search Bar

```
✓ SearchBar Component (Mock Architecture) - optimized (50) 2192ms
  ✓ renders SearchBar
  ✓ triggers search for a valid query (profiles + posts are queried)
  ✓ navigates to user profile when clicking a user result
  ✓ navigates to post when clicking a post result
  ✓ does not search for short query (<2 chars)
  ✓ shows moderator users when current user is a moderator
  ✓ shows fallback User when author has no display_name or username
  ✓ does not render results or empty state when query is less than 2 chars
  ✓ shows 'No results found' when API returns nothing
  ✓ initially has an empty query
  ✓ initially has no users
  ✓ popover is closed initially
  ✓ is not loading initially
  ✓ closes popover on Escape press
  ✓ navigates to first user on Enter press
  ✓ navigates to first post on Enter when no users found
  ✓ handles Enter press with no results
  ✓ handles user click with null username but valid userId
  ✓ handles post click with null author username but valid postId
  ✓ handles username as string 'null'
  ✓ handles empty string username
  ✓ filters out moderator users when current user is not a moderator
  ✓ filters out moderator posts when current user is not a moderator
  ✓ shows moderator posts when current user is a moderator
  ✓ handles search error gracefully
  ✓ clears query when user is clicked
  ✓ clears query when post is clicked
  ✓ opens popover on focus if query is >= 2 chars
  ✓ does not open popover on focus if query is < 2 chars
  ✓ displays avatar fallback for user without avatar
  ✓ displays avatar fallback U when no display_name or username
  ✓ debounces search calls 320ms
  ✓ does not search when query is exactly 1 character
  ✓ searches when query is exactly 2 characters
  ✓ does not navigate when username is empty string
  ✓ does not navigate when username is exactly 'null' string
  ✓ navigates when username is valid non-empty string
  ✓ does not navigate to post when post author username is empty
  ✓ does not navigate to post when post author username is 'null'
  ✓ navigates to post when post author username is valid
  ✓ shows 'No results found' when query is exactly 2 characters and no results
  ✓ does not show 'Users' header when no users found
  ✓ does not show 'Posts' header when no posts found
  ✓ applies moderator filter when user is not logged in (profile null)
  ✓ orders posts by created at descending
  ✓ navigates on Enter with exactly 2 characters
  ✓ opens popover on focus with exactly 2 characters
  ✓ shows loader while waiting for debounce
  ✓ filters out moderator users when current user is null (client-side filter)
  ✓ filters out moderator posts when current user is null (client-side filter)
```

```
Test Files 1 passed (1)
Tests 50 passed (50)
Start at 12:41:26
Duration 6.82s (transform 256ms, setup 781ms, collect 1.39s, tests 2.19s, environment 870ms, prepare 345ms)
```

100% Report is generated

You can run `npx vite preview --outDir html` to see the test results.

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	99.87	98.2	88.23	99.87	
components/search	100	97.22	100	100	
search-bar.tsx	100	97.22	100	100	199,234
components/search/search-bar.test.tsx	99.56	86.87	80	99.56	
SearchBar.test.tsx	99.56	86.87	80	99.56	51-52,64
components/ui	100	100	100	100	
avatar.tsx	100	100	100	100	
input.tsx	100	100	100	100	
popover.tsx	100	100	100	100	
lib	60	100	50	60	
utils.ts	60	100	50	60	15-20

11. Settings

```
✓ src/components/settings/archived-posts-list-ui.test/ArchivePostList.test.tsx (15) 599ms
  ✓ ArchivedPostList Component (Mock Architecture Updated) (15) 596ms
    ✓ shows loading spinner initially
    ✓ displays archived posts with correct query parameters
    ✓ renders post with string media url
    ✓ renders post with object media url
    ✓ renders post without media
    ✓ unarchives a post and updates UI immediately
    ✓ unarchives one post from a list of multiple posts
    ✓ handles null data from fetch correctly
    ✓ displays specific error message from backend on unarchive failure
    ✓ shows spinner and hides refresh icon during unarchive action
    ✓ does not fetch posts if user is not logged in
    ✓ shows message for no archived posts
    ✓ handles load error gracefully and logs to console
    ✓ handles unarchive errors
    ✓ does not unarchive if user is not logged in
  ✓ src/components/settings/verification-status-card.test/VerificationStatusCard.test.tsx (4)
  ✓ VerificationStatusCard() - Updated Mock Architecture (4)
    ✓ Happy Paths (2)
    ✓ renders eligible verification state correctly
    ✓ renders ineligible verification state correctly
    ✓ Edge Cases (2)
    ✓ handles zero factors correctly
    ✓ handles factors without metadata
  ✓ src/components/settings/profile-form.test/ProfileForm.test.tsx (8) 386ms
  ✓ ProfileForm() - Mock Architecture Updated (8) 385ms
    ✓ renders ProfileForm with initial data
    ✓ updates profile successfully
    ✓ shows error when bio exceeds 500 characters
    ✓ shows error when display name is too short
    ✓ allows bio with exactly 500 characters
    ✓ allows display name with exactly 3 characters
    ✓ shows loading state while saving
    ✓ handles update failure correctly
```

```
Test Files 3 passed (3)
Tests 77 passed (77)
```

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	98.79	90	90.47	98.79	
components/settings	100	91.93	90	100	
archived-posts-list-ui.tsx	100	93.93	100	100	69,110
profile-form.tsx	100	81.25	80	100	21-23
verification-status-card.tsx	100	100	100	100	
components/settings/archived-posts-list-ui.test	100	100	100	100	
ArchivedPostList.test.tsx	100	100	100	100	
components/settings/profile-form.test	100	100	100	100	
ProfileForm.test.tsx	100	100	100	100	
components/settings/verification-status-card.test	100	100	100	100	
VerificationStatusCard.test.tsx	100	100	100	100	
components/ui	96.25	91.66	100	96.25	
button.tsx	100	50	100	100	45
card.tsx	89.47	100	100	89.47	71-76
input.tsx	100	100	100	100	
label.tsx	100	100	100	100	
progress.tsx	100	100	100	100	
textarea.tsx	100	100	100	100	
lib	60	100	50	60	
utils.ts	60	100	50	60	15-20

12. Stories

```

src/components/stories/StoryUploadForm.mutation.test.tsx (5)
✓ StoryUploadForm - mutation killers (5)
  ✓ blocks upload for non-image types and does not call upload
  ✓ blocks upload when file exceeds size and shows error toast
  ✓ uploads small image successfully and shows success toast
  ✓ shows error when storage upload fails and prevents duplicate submissions
  ✓ does not render when currentUserId is empty (guard)
src/components/stories/StoryUploadForm.test/StoryUploadForm.test.tsx (4)
✓ StoryUploadForm (Mock Architecture Version) (4)
  ✓ Happy Paths (1)
    ✓ handles successful file upload
  ✓ Edge Cases (3)
    ✓ does not render when currentUserId is empty
    ✓ handles upload failure gracefully
    ✓ does not call uploadStory when no file is selected

Test Files 2 passed (2)
Tests 9 passed (9)
Start at 12:43:48
Duration 4.60s (transform 121ms, setup 308ms, collect 2.44s, tests 425ms, environment 2.15s, prepare 1.56s)

HTML report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from vitest

```

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	97.78	91.3	100	97.78	
components/stories	100	93.1	100	100	
StoryUploadForm.mutation.test.tsx	100	100	100	100	
StoryUploadForm.test	100	97.5	100	100	40-43
components/stories/StoryUploadForm.test	100	100	100	100	
StoryUploadForm.test.tsx	100	100	100	100	
lib/supabase	87.17	66.66	100	87.17	
storyUpload.ts	87.17	66.66	100	87.17	32-34,54-55

13. UI

```

src/components/ui/avatar.test.tsx (2)
✓ Avatar (2)
  ✓ renders root with merged classes
  ✓ renders fallback content when AvatarFallback is present
src/components/ui/badge.test.tsx (2)
✓ Badge (2)
  ✓ renders children and forwards className
  ✓ applies variant classes
src/components/ui/button.test.tsx (3)
✓ Button (3)
  ✓ forwards props and renders children
  ✓ applies variant and size classes
  ✓ supports asChild prop by rendering passed element type
src/components/ui/input.test.tsx (2)
✓ Input (2)
  ✓ renders input and forwards props
  ✓ accepts type attribute and additional classes
src/components/ui/label.test.tsx (1)
✓ Label (1)
  ✓ renders text and forwards htmlFor
src/components/ui/textarea.test.tsx (2)
✓ Textarea (2)
  ✓ forwards props and renders initial value
  ✓ accepts className and merges classes
src/components/ui/dropdown-menu.test/dropdown-menu.test.tsx (5)
✓ dropdown-menu primitives (5)
  ✓ renders content inside a portal with default side offset
  ✓ adds chevron icon plus inset classes to sub trigger
  ✓ shows checkbox indicator when checked
  ✓ renders radio indicator and respects radio group
  ✓ merges classes in shortcut helper
src/components/ui/separator.test/separator.test.tsx (3)
✓ Separator (3)
  ✓ applies horizontal defaults
  ✓ switches styles for vertical orientation
  ✓ forwards refs to radix root
src/components/ui/sheet.test/sheet.test.tsx (4)
✓ Sheet primitives (4)
  ✓ renders overlay with merged classes
  ✓ renders content, overlay, and close affordance
  ✓ honors the side variant classes
  ✓ renders header and footer helpers
src/components/ui/switch.test/switch.test.tsx (3)
✓ Switch (3)
  ✓ merges class names and props
  ✓ renders thumb with translate classes
  ✓ forwards refs to radix root

src/components/ui/toast.test/toast.test.tsx (5)
✓ toast primitives (4)
  ✓ merges viewport classes
  ✓ applies destructive toast variant
  ✓ renders action and forwards events
  ✓ renders title and description content
✓ toaster integration (1)
  ✓ renders toast items from hook state
src/components/ui/tooltip.test/tooltip.test.tsx (3)
✓ tooltip primitives (3)
  ✓ defaults sideOffset to 4
  ✓ honors custom props and merges classes
  ✓ renders provider, trigger, and root structure

Test Files 12 passed (12)
Tests 35 passed (35)

```

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	94.89	100	85.71	94.89	
components/ui	93.03	100	100	93.03	
avatar.test.tsx	100	100	100	100	
avatar.tsx	82.35	100	100	82.35	27-32
badge.test.tsx	100	100	100	100	
badge.tsx	100	100	100	100	
button.test.tsx	100	100	100	100	
button.tsx	100	100	100	100	
dropdown-menu.test.tsx	74.07	100	100	74.07	47-55,83-92,147-156,163-168
input.test.tsx	100	100	100	100	
input.tsx	100	100	100	100	
label.test.tsx	100	100	100	100	
label.tsx	100	100	100	100	
separator.test.tsx	100	100	100	100	
sheet.test.tsx	100	100	100	100	
sheet.tsx	100	100	100	100	
switch.test.tsx	100	100	100	100	
switch.tsx	100	100	100	100	
textarea.test.tsx	100	100	100	100	
textarea.tsx	100	100	100	100	
toast.test.tsx	100	100	100	100	
toast.tsx	100	100	100	100	
tooltip.test.tsx	100	100	100	100	
components/ui/dropdown-menu.test.tsx	100	100	100	100	
dropdown-menu.test.tsx	100	100	100	100	
components/ui/separator.test.tsx	100	100	100	100	
separator.test.tsx	100	100	100	100	
components/ui/sheet.test.tsx	100	100	100	100	
sheet.test.tsx	100	100	100	100	
components/ui/switch.test.tsx	100	100	100	100	
switch.test.tsx	100	100	100	100	
components/ui/toast.test.tsx	100	100	100	100	
toast.test.tsx	100	100	100	100	
components/ui/tooltip.test.tsx	100	100	100	100	
tooltip.test.tsx	100	100	100	100	
lib	60	100	50	60	15-20
utils.ts	60	100	50	60	15-20

• App Header

✓ src/components/app-header.test.tsx (4)					
✓ Appheader (4)					
✓ renders search, moderator controls, and profile avatar					
✓ signs out and navigates to login when logout buttons are pressed					
✓ opens the create-post modal when the add button is clicked					
✓ falls back to user id when profile username is missing					
Test Files 1 passed (1)					
Tests 4 passed (4)					
Start at 12:48:09					
Duration 4.46s (transform 507ms, setup 142ms, collect 1.08s, tests 285ms, environment 642ms, prepare 1.14s)					
HTML Report is generated					
You can run <code>npx vite preview --outDir html</code> to see the test results.					
% Coverage report from v8					
File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	95.66	76.92	60	95.66	
components	96.77	75	62.5	96.77	
app-header.test.tsx	100	100	100	100	
app-header.tsx	94	61.53	57.14	94	27-29,35-36,110
logo.tsx	100	100	100	100	
components/ui	100	80	100	100	
avatar.tsx	100	100	100	100	
button.tsx	100	50	100	100	45
lib	60	100	50	60	15-20
utils.ts	60	100	50	60	15-20

• App Shell

✓ src/components/app-shell.test.tsx (1)					
✓ Appshell (1)					
✓ renders sidebar, header, and provided children					
Test Files 1 passed (1)					
Tests 1 passed (1)					
Start at 12:48:42					
Duration 2.13s (transform 60ms, setup 142ms, collect 170ms, tests 36ms, environment 977ms, prepare 155ms)					
HTML Report is generated					
You can run <code>npx vite preview --outDir html</code> to see the test results.					
% Coverage report from v8					
File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
app-shell.test.tsx	100	100	100	100	
app-shell.tsx	100	100	100	100	

• Banned

✓ src/components/banned-notice.mutations.test.tsx (3)					
✓ Bannednotice mutation killers (3)					
✓ handles invalid banneduntil and shows indefinite copy (kills invalid date branch)					
✓ passes checkboxes through and tries combined class string					
✓ logout button triggers navigation refresh					
✓ src/components/banned-notice.test.tsx (3)					
✓ Bannednotice (3)					
✓ renders reason, description, and formatted suspension summary					
✓ falls back to indefinite suspension copy when no date is provided					
✓ signs out and navigates to login when the button is clicked					
Test Files 2 passed (2)					
Tests 6 passed (6)					
Start at 12:49:24					
Duration 4.63s (transform 163ms, setup 332ms, collect 2.58s, tests 228ms, environment 1.65s, prepare 1.33s)					
HTML Report is generated					
You can run <code>npx vite preview --outDir html</code> to see the test results.					
% Coverage report from v8					
File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All Files	96.47	90	80	96.47	
components	100	100	100	100	
banned-notice.mutations.test.tsx	100	100	100	100	
banned-notice.test.tsx	100	100	100	100	
banned-notice.tsx	100	100	100	100	
components/ui	100	50	100	100	
button.tsx	100	50	100	100	45
lib	60	100	50	60	15-20
utils.ts	60	100	50	60	15-20

Hooks Folder:

• Use-mobile

```
✓ src/hooks/use-mobile.test/useMobile.test.tsx (9)
✓ useMobile() (9)
  ✓ uses correct matchMedia query string
  ✓ returns false when innerWidth === 768
  ✓ attaches event listener with event name 'change'
  ✓ removes event listener with event name 'change'
  ✓ effect runs only once even after rerender
  ✓ returns true when width < breakpoint
  ✓ returns false when width > breakpoint
  ✓ updates when resized below breakpoint
  ✓ cleanup removes event listener
```

• Use-Realtime

```
✓ src/hooks/use-realtime.test/useRealtime.string-mutants.test.tsx (11)
✓ useRealtime string literal mutants (1)
  ✓ uses channel name with user id and correct event strings
✓ src/hooks/use-realtime.test/useRealtime.test.tsx (12)
✓ useRealtime integration tests (12)
  ✓ Happy paths (5)
    ✓ should subscribe to conversation and message events when all handlers are provided
    ✓ should only subscribe to conversation event when only onConversationAdded is provided
    ✓ should unsubscribe and remove channel on unmount
    ✓ should call the correct handler when channel event fires
    ✓ should not wire up any handlers when none are provided
  ✓ Edge cases (4)
    ✓ should do nothing if currentUserid is missing
    ✓ should log error if subscription callback receives CHANNEL_ERROR
    ✓ should resubscribe when currentUserid changes
    ✓ should not break if handlers are removed on rerender
  ✓ calls onMessageUpdated when UPDATE event occurs and provides old/new
  ✓ logs an error when the subscribe callback signals CHANNEL_ERROR
  ✓ should call onMessageInserted when INSERT event for messages occurs
```

• Use-Toast

```
✓ src/hooks/use-toast.test/toast.test.tsx (21)
✓ reducer() (9)
  ✓ adds a toast
  ✓ respects TOAST_LIMIT = 1
  ✓ updates toast
  ✓ dismisses one toast
  ✓ dismisses all toasts
  ✓ removes specific toast
  ✓ remove-all sets empty array
  ✓ ignore update when ID does not match
  ✓ ignore unknown action entirely
  ✓ internal behavior: add/remove/queue & timeouts (2)
    ✓ does NOT create new timeout when toast already scheduled
    ✓ creates timeout when none exists
  ✓ useToast() integration (10)
    ✓ renders toast
    ✓ dismiss via button
    ✓ dismiss all
    ✓ updates toast via dispatch
    ✓ enforces TOAST_LIMIT = 1
    ✓ non-existent dismiss is safe
    ✓ onOpenChange(false) dismisses
    ✓ onOpenChange(true) does NOT dismiss
    ✓ rapid add + dismiss works
    ✓ updates a toast using the update() API (covers UPDATE_TOAST dispatch)
✓ src/hooks/use-toast.test/useToast.test.tsx (9)
✓ useToast Integration Tests (9)
  ✓ Happy paths (4)
    ✓ should add a toast and render it
    ✓ should dismiss all toasts
    ✓ should update a toast's content
    ✓ should enforce TOAST_LIMIT = 1
  ✓ Edge cases (5)
    ✓ dismiss non-existent toast - no error
    ✓ update non-existent toast - no effect
    ✓ remove non-existent toast - no error
    ✓ should allow toast with only title
    ✓ should handle rapid add + dismiss
```

• Use-User

```
✓ src/hooks/use-user.test/user.test.tsx (63) 905ms
✓ useUser() Hook - Vitest (63) 905ms
  ✓ Happy Paths (3)
    ✓ should fetch user + profile successfully
    ✓ should use cached profile and skip loading
    ✓ should update user/profile on SIGNED_IN
  ✓ Edge Cases (4)
    ✓ should handle no user returned
    ✓ should handle getuser error
    ✓ should handle sessionStorage.getItem error gracefully
    ✓ should clear user/profile on SIGNED_OUT
  ✓ should ignore unknown realtime auth events
  ✓ should unsubscribe realtime listener on unmount
  ✓ should not call profile query for other events
  ✓ Additional Branch Coverage (2)
    ✓ should throw when fetchProfile receives unexpected error.code
    ✓ should ignore cache write errors when writing sessionStorage
  ✓ handles Supabase error.code === 'PGRST110' without throwing
  ✓ retries fetchProfile until retries reach zero
  ✓ ignores sessionStorage.setItem error and still sets profile + loading false
  ✓ should handle thrown exception in getItem() (covers catch block)
  ✓ covers cache WRITE error catch block (sessionStorage.setItem throws)
  ✓ covers cache READ error catch block (sessionStorage.getItem throws)
  ✓ covers stale cache branch (isStale = true)
  ✓ triggers USER_UPDATED event branch
  ✓ re-fetches profile when profileRef.current is null on auth event
  ✓ does NOT re-fetch profile if profileRef.current is set and event is TOKEN_REFRESHED
  ✓ verifies profile query is called when user exists (kills if !data check)
  ✓ does not query profile when no user (kills branches)
  ✓ uses fresh cache (not stale) and still refreshes
  ✓ covers the cachedAt = undefined stale branch
  ✓ handles fetchProfile catch block for thrown exception
  ✓ handles PGRST110 error code exactly (string comparison)
  ✓ handles different error code (not PGRST110) and throws
  ✓ clears profile on SIGNED_OUT and sets loading false
  ✓ handles null cachedProfile gracefully
  ✓ handles empty cache value (res = null)
  ✓ asserts initial loading=true BEFORE data is fetched
  ✓ confirms SIGNED_IN event triggers fetchProfile and sets loading=true
  ✓ asserts cache staleness: exactly 5 min is stale because > comparison
  ✓ asserts cache staleness: just over 5 min is stale and triggers refresh
  ✓ verifies fetchProfile returns true on success (kills return false mutant)
  ✓ verifies cache is written to sessionStorage after successful fetch
  ✓ confirms SIGNED_OUT clears sessionStorage cache
  ✓ calls from/select/on with correct arguments
```

```

    asserts profiler.current prevents re-fetch on irrelevant events
    verifies setLoading(true) called on SIGNED_IN triggers loading=true
    handles amount before async completes (ifounted guard)
    verifies retry returns false (not true) when retrying 526ms
    verifies error message is set correctly (not empty string) 2538ms
    verifies cache is only read once per mount (hasInitializedFromCacheRef)
    asserts USER_UPDATED event specifically triggers refetch
    verifies setLoading(false) called after gettiser error
    verifies SIGNED_IN exact string match (not empty string)
    verifies setLoading(false) is called after retries exhausted (kills true mutant) 2553ms
    asserts fresh cache (< 5 min) does NOT trigger immediate loading state
    asserts stale cache (> 5 min) still sets loading=false after cache read
    verifies retry count exactly (retries > 0 vs >= 0) 2526ms
    confirms error thrown on unexpected error code sets error state
    asserts cachedProfile falsy guard (null cachedProfile in cache)
    asserts raw falsy guard (empty string in cache)
    verifies error state persists after catch block in fetchProfile
    isStale: cachedAt=0 is falsy so cache is stale
    isStale: very old cachedAt triggers refresh
    isStale: exactly at boundary uses > not >=
    isStale: 1ms over boundary is definitely stale
    verifies successful fetch after error clears error state
    verifies return true from fetchProfile when data is found

st Files 6 passed (6)
Tests 115 passed (115)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
all files	99.41	99.52	94.44	99.41	
hooks	97.82	96.84	100	97.82	
use-mobile.tsx	100	100	100	100	
use-realtime.ts	100	100	100	100	
use-toast.ts	96.18	100	100	96.18	67-71
use-user.ts	98.27	92.68	100	98.27	43,92
hooks/use-mobile.test	100	100	100	100	
hooks/use-mobile.test.tsx	100	100	100	100	
hooks/use-realtime.test	100	100	100	100	
hooks/use-realtime.test.tsx	100	100	100	100	
hooks/use-realtime.string-mutants.test.tsx	100	100	100	100	
hooks/use-realtime.test.tsx	100	100	100	100	
hooks/use-toast.test	100	100	100	100	
hooks/use-toast.test.tsx	100	100	100	100	
hooks/use-toast.test.tsx	100	100	100	100	
hooks/use-user.test	99.67	100	85.71	99.67	273-274,509,615,680-717
hooks/use-user.test.tsx	99.67	100	85.71	99.67	

Lib Folder

• Verification

```
✓ src/lib/verification/mutation.test.ts (7)
✓ verification.ts mutation killers (7)
  ✓ computes percentage correctly and sets eligible at threshold boundary
  ✓ returns not eligible when points exactly one below threshold
  ✓ treats posts without created at as non-recent and ensures >= comparison (not >)
  ✓ average calculations change when totalPosts=0 (guards ternary true/false mutants)
  ✓ avatar whitespace vs non-empty toggles achieved factors (kills hasAvatar mutants)
  ✓ bio empty vs non-empty toggles achieved factors (kills hasBio mutants)
  ✓ eligible is true when totalPoints equals threshold (kills >= vs > and eligible=false mutants)
✓ src/lib/verification.test/evaluateUserVerification.test.ts (20)
✓ evaluateUserVerification - numeric correctness (5)
  ✓ correctly computes likes, comments, engagement, and recent posts
  ✓ boundary: account age 29 days - NOT achieved
  ✓ boundary: account age 30 days - achieved
  ✓ follows threshold boundary - 499 no, 500 yes
  ✓ engagement threshold boundary
    ✓ reports available vs unavailable
✓ verification.ts uncovered branches (14)
  ✓ fetchProfileBasics returns null when error occurs (101-121)
  ✓ fetchProfileBasics returns null when data is null, no error (101-121)
  ✓ fetchProfileBasics sets createdAt=null when created_at missing (131-132)
  ✓ fetchPostStats returns zero stats when Spabase returns error (211-212)
  ✓ evaluateEngagement - totalPosts=0 - achieved=false (241-248)
  ✓ evaluateEngagement - engagement < threshold - achieved=false (241-248)
  ✓ evaluateRecentActivity - recentPosts < MIN_RECENT_POSTS (261-262)
  ✓ evaluatePostHistory returns achieved=true when data unavailable (307-312)
  ✓ evaluateUserVerification returns default result when profileBasics=null (covers 101-121)
  ✓ fetchProfileBasics sets createdAt=null when created_at is missing (covers 131-132)
  ✓ fetchPostStats treats posts without created at as non-recent (covers 201-202)
  ✓ evaluateCountdown handles null createdAt (covers 307-312)
  ✓ fetchProfileBasics correctly detects empty avatar_url when whitespace
  ✓ avatar_url length === 0 branch is handled correctly

Test Files 2 passed (2)
Tests 27 passed (27)
Start at 12:55:32
Duration 3.24s (transform 163ms, setup 323ms, collect 133ms, tests 20ms, environment 2.12s, prepare 1.71s)
```

INFO Report is generated

You can run **npx vite preview --outDir html** to see the test results.

% Coverage report from vit

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	97.81	92.34	92	97.8	
lib	96.2	89.3	75	96.2	
verification/mutation.test.ts	92.47	88.65	70	92.47	48, 90-93, 101, 149-152, 160, 202-205, 213, 256-259, 267
verification.ts	100	90.32	100	100	172-183, 218, 251-252, 295
lib/verification.test	100	100	100	100	
evaluateUserVerification.test.ts	100	100	100	100	

• Messages

```
✓ src/lib/messages/mutation.test.ts (11)
✓ messages.ts mutation killers (11)
  ✓ createOrGetConversationWith returns existing overlap when found
  ✓ createOrGetConversationWith creates when no overlap
  ✓ createOrGetConversationWith creates when existing select returns null (guards if(true) on existing)
  ✓ createOrGetConversationWith creates when overlap select returns null (guards if(true) on overlap)
  ✓ fetchMessages returns empty array on no data and handles limit
  ✓ subscribeToMessages wires up INSERT with correct filter and cleans up
  ✓ createOrGetConversationWith does not query overlap when existing is empty and creates conversation
  ✓ createOrGetConversationWith creates when overlap is empty despite existing convids
  ✓ fetchConversations returns [] and does not query conversations when convids undefined
  ✓ fetchConversations returns [] and does not query conversations when convids empty
  ✓ sendMessage throws when insert returns error (kills if(false) mutants)
✓ src/lib/messages.test.ts (32)
✓ createOrGetConversationWith() (5)
  ✓ unauthorized
  ✓ throws when first select fails
  ✓ returns existing overlap
  ✓ unauth returns empty
  ✓ select error throws
  ✓ empty list returns empty
  ✓ null conversation list becomes empty
  ✓ returns conversations
✓ fetchMessages() (2)
  ✓ returns message list
✓ subscribeToMessages() (17)
  ✓ unsubscribes correctly
  ✓ invokes callback
  ✓ ignores invalid payload
  ✓ subscribe error throws
  ✓ createOrGetConversationWith() throws when participant insert fails
  ✓ createOrGetConversationWith() returns conv.id when participant inserts succeed
  ✓ sendMessage() throws when insert returns null data
  ✓ sendMessage() returns message when insert succeeds
  ✓ createOrGetConversationWith() throws when second select fails
  ✓ fetchConversations() throws when conversations query fails
  ✓ fetchConversations() returns [] when conversations query returns null
  ✓ sendMessage() rejects messages that become empty after trim
  ✓ subscribeToMessages() ignores payloads without 'new' field
  ✓ fetchMessages() returns [] when data is null
  ✓ sendMessage() throws the correct error message when insert returns null data
  ✓ fetchMessages() goes through error branch
  ✓ fetchMessages() covers the no-error branch
```

Test Files 2 passed (2)

Tests 41 passed (41)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	97.76	89.15	100	97.76	
messages/mutation.test.ts	92.62	97.61	100	92.62	45, 60, 74, 88, 129, 155, 170-173, 189-192, 207
messages.test.ts	100	100	100	100	
messages.ts	100	100	100	100	

• Notifications


```
✓ src/lib/notifications.test.ts (18)
✓ Notifications Logic (100% White Box Coverage) (18)
  ✓ fetchNotifications (5)
    ✓ builds exact query and logs correct length for non-empty list
    ✓ should return empty array and log if no user is found
    ✓ should fetch and return notifications for the user
    ✓ should return empty array and log error if fetch fails
    ✓ logs literal label 'notifications' in fetch log (kills StringLiteral)
  ✓ markAllAsRead (2)
    ✓ should return early if no user is logged in
    ✓ should update all notifications for the current user
  ✓ markAsRead (1)
    ✓ should update a specific notification by ID
  ✓ getUnreadCount (2)
    ✓ should return 0 if no user is logged in
    ✓ should return the correct count of unread notifications
  ✓ subscribeToNotifications (3)
    ✓ should set up channel subscription with exact options and return an unsubscribe function
    ✓ should call oninsert with fetched profile data on new message
    ✓ should call oninsert with raw data if profile fetch fails (catch block coverage)
  ✓ subscribeToNotifications - catch block (lines 116-117) (1)
    ✓ calls oninsert with raw row when profile fetch fails (catch block)
  ✓ fetchNotifications returns [] and logs error when DB returns error
  ✓ getUnreadCount returns 0 when count is null (kills ?? mutant)
  ✓ fetchNotifications logs 0 when data is empty (covers '|| 0')
  ✓ fetchNotifications returns [] when data is null (line 59 fallback)

Test Files 1 passed (1)
Tests 18 passed (18)
Start at 12:57:22
Duration 3.06s (transform 170ms, setup 164ms, collect 78ms, tests 17ms, environment 946ms, prepare 799ms)

📄 Report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from all

| File                  | % Stats | % Branch | % Funcs | % Lines | Uncovered Line #s |
|-----------------------|---------|----------|---------|---------|-------------------|
| All files             | 100     | 100      | 100     | 100     |                   |
| notifications.test.ts | 100     | 100      | 100     | 100     |                   |
| notifications.ts      | 100     | 100      | 100     | 100     |                   |


```

• Utils

```
✓ src/lib/utils.mutation.test.ts (3)
✓ utils.ts mutation killers (3)
  ✓ cn merges classes with precedence
  ✓ getBucketOrThrow returns configured bucket
  ✓ getBucketOrThrow throws when bucket missing
✓ src/lib/utils.test.ts (4)
✓ cn utility function (100% line/function coverage) (2)
  ✓ should merge simple string classes correctly
  ✓ should resolve fallback conflicts, prioritizing the rightmost class
✓ getBucketOrThrow (100% Branch Coverage Test) (2)
  ✓ should return the correct bucket name when configured
  ✓ should throw an error for an invalid bucket kind

Test Files 2 passed (2)
Tests 7 passed (7)
Start at 12:58:06
Duration 2.65s (transform 121ms, setup 469ms, collect 88ms, tests 48ms, environment 1.47s, prepare 1.12s)

📄 Report is generated
You can run npx vite preview --outDir html to see the test results.
% Coverage report from all

| File                   | % Stats | % Branch | % Funcs | % Lines | Uncovered Line #s |
|------------------------|---------|----------|---------|---------|-------------------|
| All files              | 100     | 100      | 100     | 100     |                   |
| utils.mutation.test.ts | 100     | 100      | 100     | 100     |                   |
| utils.test.ts          | 100     | 100      | 100     | 100     |                   |
| utils.ts               | 100     | 100      | 100     | 100     |                   |


```