PROJECT REPORT

ON

NAVONMESH : HACKATHON HOSTING PLATFORM

FOR

SKILLI

BY

ANIS YASIN KHAN

UNIVERSITY OF PUNE

MASTER IN COMPUTER APPLICATION

Maharashtra Education Society's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411038

2025-2026

MAHARASHTRA EDUCATION SOCIETY'S
(SINCE 1860)

INSTITUTE OF MANAGEMENT & CAREER COURSES (IMCC)
(AUTONOMOUS)

Approved by AICTE and Recognized by Savitribai Phule Pune University, Pune
IMCC Campus, 131, Mayur Colony, Kothrud, Pune 411038, Maharashtra, India | Ph.: 020-2546 6271 / 73 | e-mail: info.imcc@mespune.in | https://imcc.mespune.in

NAAC Accredited with Grade A+

Ref. No. MES IMCC /442/2025-26 **Date:06/05/2026**

CERTIFICATE

This is to certify that the Project Report entitled

"Navonmesh - Hackathon Hosting Platform"

is prepared by

Anis Yasin Khan

M.C.A. Semester IV Course for the Academic Year 2025-26 at M.E. Society's Institute of Management & Career Courses (IMCC), Pune - 411038. M.C.A Course is autonomous and affiliated to Savitribai Phule Pune University.

To the best of our knowledge, this is an original study done by the said student and important sources used by him/her have been duly acknowledged in this report.

The report is submitted in partial fulfillment of M.C.A Course for the Academic Year 2025-26 as per the rules and prescribed guidelines of Savitribai Phule Pune University.

Dr Shilpa Parlikar
Project Coordinator

Ms. Manasi Shirurkar
Programme Coordinator

Dr. Ravikant Zirmite
Head, Dept of MCA MES IMCC

Dr. Santosh Deshpande
Director, MES IMCC

External Examiner

Skilli
Your Idea, Our Execution.

Project Recognition Letter

1st Feb 2026

To,
Concerned Authority

Subject: Recognition Letter for Academic Project Work

This is to certify that Mr. Anis Khan has successfully worked with SKILLI on the development of "Navonmesh", a hackathon hosting and management platform.

Navonmesh is a digital platform designed to streamline and simplify the complete hackathon management process, including participant registration, project submissions, evaluation workflows, judging processes, and result management.

During the project period, Mr. Anis Khan contributed to the successful development and completion of the platform by actively participating in software development tasks, feature implementation, testing, and project coordination activities.

The project has been successfully completed and delivered, providing valuable practical industry exposure and hands-on experience in real-world software product development.

This letter is being issued upon request for academic documentation and project submission purposes.

Sincerely,

Prathmesh

Prathmesh Ghatmal
Proprietor

SKILLI
Flat C1, C2, Anandbaug Society
Vinayak Nagar, Burudgaon Road
Ahilyanagar, Dist. Ahmednagar
Maharashtra - 414001

+91 8830727116
prathmesh@skilli.in
skilli.in
Flat C1, C2, Anand Baug Society, Vinayak Nagar, Burudgaon Road, Nagar, District Ahilyanagar, Maharashtra - 414001

Maharashtra Education Society's

Institute of Management and Career Courses (IMCC), Pune

AUTONOMOUS

[CGPA-3.32] in Cycle 1

CERTIFICATE

This is to certify that Anis Yasin Khan has completed the project work entitled "***Navonmesh – Hackathon Hosting Platform***" under my guidance. The report is submitted in partial fulfillment of M.C.A. Course for the Academic Year 2025-2026 as per the rules & prescribed guidelines.

His/her work is found to be satisfactory and complete in all respects.

Date:09/05/2026

SSParlikar

Dr. Shilpa Parlikar

(Internal Project Guide)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone who supported and guided me during the successful completion of this project report titled "**Navonmesh : Hackathon hosting platform.**"

I am deeply thankful to the faculty members of M.E. Society's Institute of Management & Career Courses (IMCC), Pune, for providing valuable guidance, encouragement, and continuous support throughout the project work.

I would especially like to thank my project guide for their guidance, suggestions, and motivation, which helped me complete this project successfully.

I also extend my gratitude to all individuals, mentors, friends, and well-wishers who directly or indirectly contributed to the completion of this project.

Finally, I express my heartfelt thanks to my family for their constant support, encouragement, and motivation throughout this journey.

Anis Khan
M.C.A. Semester IV
Academic Year 2025–2026

1. Application Development Project

Chapter No	Details	Page no
1	**Introduction**	
1.1	Company Profile	**1**
1.2	Problem Statement	**2**
1.3	Objectives of Proposed System	**3**
1.4	Scope of the Proposed System	**5**
2	**Analysis and Design**	
2.1	Entity Relationship Diagram (ERD)	**9**
2.2	Business Use Case Diagram	**10**
2.3	Table Structure / Data Dictionary	**11**
3	**Testing**	
3.1	Test Cases with Results	**16**
3.2	Defect Report / Test Log	**21**
4	**Limitations of Proposed System**	**23**
5	**Proposed Enhancements**	**24**
6	**References**	**25**
7	**User Manual (All input and output screens with consistent data)**	**27**

CHAPTER 1

INTRODUCTION

1.1 Company Profile

Skilli is a premier product engineering studio headquartered in Pune, India. Founded in 2026, Skilli specializes in building scalable, future-proof digital solutions that drive business growth. From startups to enterprises, Skilli serves as a trusted technical partner — transforming ideas into reality through cutting-edge technology, exceptional design, and reliable engineering.

Skilli's mission is to empower businesses with cutting-edge technology and exceptional design. The studio offers a comprehensive range of services including Web Development (React, Next.js, MERN Stack), Mobile App Development (cross-platform), AI Integration (LLMs, Machine Learning), UI/UX Design, MVP Development, DevOps Solutions, Automation, and QA Testing. With 10+ projects delivered and a 100% client satisfaction rate, Skilli has rapidly established itself as a trusted name in the custom software development domain.

Parameter	Details
Company Name	Skilli
Tagline	Transforming Ideas into Reality
Founded	2026
Headquarters	Pune, Maharashtra, India
Industry	IT Services and IT Consulting / Software Development
Company Type	Product Engineering Studio
Website	https://skilli.in
Email	enquiry.skilli@gmail.com
Phone	+91 8830727116
Key Services	Web Development, App Development, AI Solutions, UI/UX Design, MVP Development, DevOps, Automation, QA Testing, SEO Optimization
Projects Delivered	10+

Client Satisfaction	100%

As part of its commitment to innovation and community building, Skilli has partnered with Maharashtra Education Society's Institute of Management and Career Courses (IMCC), Pune, as the Official Technology Partner for Navonmesh 2026 — an inter-college national-level hackathon focused on Fintech, Health Tech, Sustainability, Industrial Automation, Agritech, and bonus innovation topics.

To manage the complex logistics of this large-scale event — from registration and payments to multi-round evaluations and real-time operations — Skilli has designed and developed a dedicated, full-stack Hackathon Management Platform. This documentation provides an in-depth overview of the system's design, implementation, testing, and operational guidelines.

1.2 Problem Statement

Managing a large-scale hackathon involves coordinating multiple parallel activities including team registrations, payment processing, multi-round evaluations, real-time communication, QR-based attendance tracking, report generation, and public result announcements — all within a secure, scalable, and real-time environment.

Without a dedicated management platform, organizers are forced to rely on fragmented tools such as spreadsheets, email chains, and manual attendance registers. This approach leads to significant challenges:

- Data Inconsistency: Manual data entry across multiple spreadsheets introduces errors and version conflicts.
- Communication Delays: Email chains and group messages fail to provide real-time coordination between organizers, evaluators, and participants.

• Security Vulnerabilities: Sharing credentials and sensitive data through unencrypted channels poses security risks.

• Scalability Issues: As participant count grows, manual processes become unmanageable, causing delays in evaluation, payment tracking, and result publication.

• Poor Participant Experience: Lack of a unified dashboard leads to confusion regarding team status, payment confirmation, submission deadlines, and round progression.

• No Audit Trail: Without systematic logging, tracking who did what and when becomes nearly impossible.

The Navonmesh 2026 platform addresses all of these challenges by providing a unified, end-to-end digital solution for hackathon lifecycle management.

1.3 Objectives

The primary objectives of the Navonmesh 2026 Hackathon Management Platform are as follows:

1. Unified Platform: Provide a single platform for end-to-end hackathon lifecycle management — from registration to result announcement.

2. Automated Registration: Automate team registration, credential distribution, and payment collection via Razorpay integration.

3. Multi-Round Evaluation: Enable a structured multi-round evaluation workflow (Round 1 Screening → Round 2 Submission & Scoring → Round 3 Grand Finale) with assigned evaluators and criteria-based scoring.

4. QR-Based Attendance: Implement encrypted QR-based attendance tracking with real-time scan verification and duplicate detection.

5. Real-Time Communication: Deliver community-wide real-time communication via Socket.IO-based chat and admin monitoring dashboards.

6. Public Result Pages: Provide public result pages with encrypted, cached API responses for secure and performant access.

7. PDF Report Generation: Generate professional evaluation reports in PDF format using Puppeteer for offline reference.

8. Admin Control Panels: Offer granular role-based admin dashboards with permission-scoped access and system-wide feature toggles.

9. Security: Ensure end-to-end security through JWT authentication, AES-256 encryption, CAPTCHA protection, and brute-force mitigation.

10. Observability: Provide real-time server health monitoring through Prometheus metrics, Grafana dashboards, and Socket.IO broadcasting.

1.4 Scope

The following table delineates the boundaries of the Navonmesh 2026 system, clearly identifying features that are within scope and those that fall outside the current implementation.

In Scope	Out of Scope
Team registration & member management	Self-registration by participants
Razorpay payment integration	Multiple payment gateways
Multi-round evaluation with criteria-based scoring	AI-based automatic project evaluation
QR code generation & attendance tracking	Biometric attendance systems
Real-time community chat (Socket.IO)	Video conferencing integration
Admin dashboard with role-based access	Native mobile application
Bulk email service with queue management	SMS/WhatsApp notifications
Public results & report verification	Third-party result publishing platforms
Server health monitoring (Prometheus/Grafana)	Full CI/CD pipeline integration

1.4.1 Technologies Used

Layer	Technology
Frontend	React 18, TypeScript, Vite, TailwindCSS, ShadCN UI (Radix), Framer Motion, React Router v6, React Query, Socket.IO Client, Recharts, Axios
Backend	Node.js, Express 5, Socket.IO, Mongoose ODM
Database	MongoDB (Atlas)
Caching / Sessions	Redis (via ioredis)
Job Queue	Bull (Redis-backed)
Payments	Razorpay (Orders + Webhooks)
Email	Nodemailer (Gmail / App Passwords)
Security	JWT, bcryptjs, AES-256-CBC, reCAPTCHA v2, Helmet, express-rate-limit
Monitoring	Prometheus (prom-client), Grafana
PDF Generation	Puppeteer
QR Codes	qrcode library
File Handling	Multer (uploads), xlsx (Excel export/import)

CHAPTER 2

ANALYSIS AND DESIGN

2.1 System Architecture

The Navonmesh 2026 platform follows a three-tier architecture pattern with a clear separation between the client layer, application layer, and data layer. The system communicates through RESTful APIs over HTTPS and real-time WebSocket connections for live features such as chat and server health monitoring.

2.1.1 Architecture Components

Component	Role
React SPA (Vite + TypeScript)	All user interfaces — landing page, login portals (Admin, Admin Viewer, Evaluator, Team Leader), dashboards, payment page, public results
Express REST API (/api/v1/*)	Business logic, CRUD operations, authentication, authorization, payment processing, report generation
Socket.IO Server	Real-time community chat, admin health monitoring, flash message broadcasting, email queue statistics
Bull Job Queue	Asynchronous email sending with retry logic and rate limiting
MongoDB (Atlas)	Persistent storage for all domain entities (teams, users, evaluations, payments, attendance, chat messages, etc.)
Redis	Session storage (JTI-based), API response caching, Bull queue backing store, failed login attempt tracking
Razorpay	Payment order creation, payment verification (client + webhook), and payment status tracking
Nginx	Reverse proxy, SSL termination, static file serving, WebSocket upgrade handling

2.1.2 Architecture Flow Description

The client layer consists of the React Single Page Application (SPA) and static public pages, both served through the Nginx reverse proxy. The SPA connects to the Express REST API for data operations and to the Socket.IO server for real-time

events. The application layer processes business logic and interacts with MongoDB for persistence and Redis for caching and session management. External services include Razorpay for payment processing, Gmail SMTP for email delivery, and Google reCAPTCHA for bot protection.

2.2 Entity Relationship Diagram (ERD)

The Navonmesh 2026 system uses MongoDB as its primary database with Mongoose ODM for schema validation. The system comprises 12 collections that model the complete hackathon lifecycle. Below is a description of all entities and their relationships.

2.2.1 Entities Overview

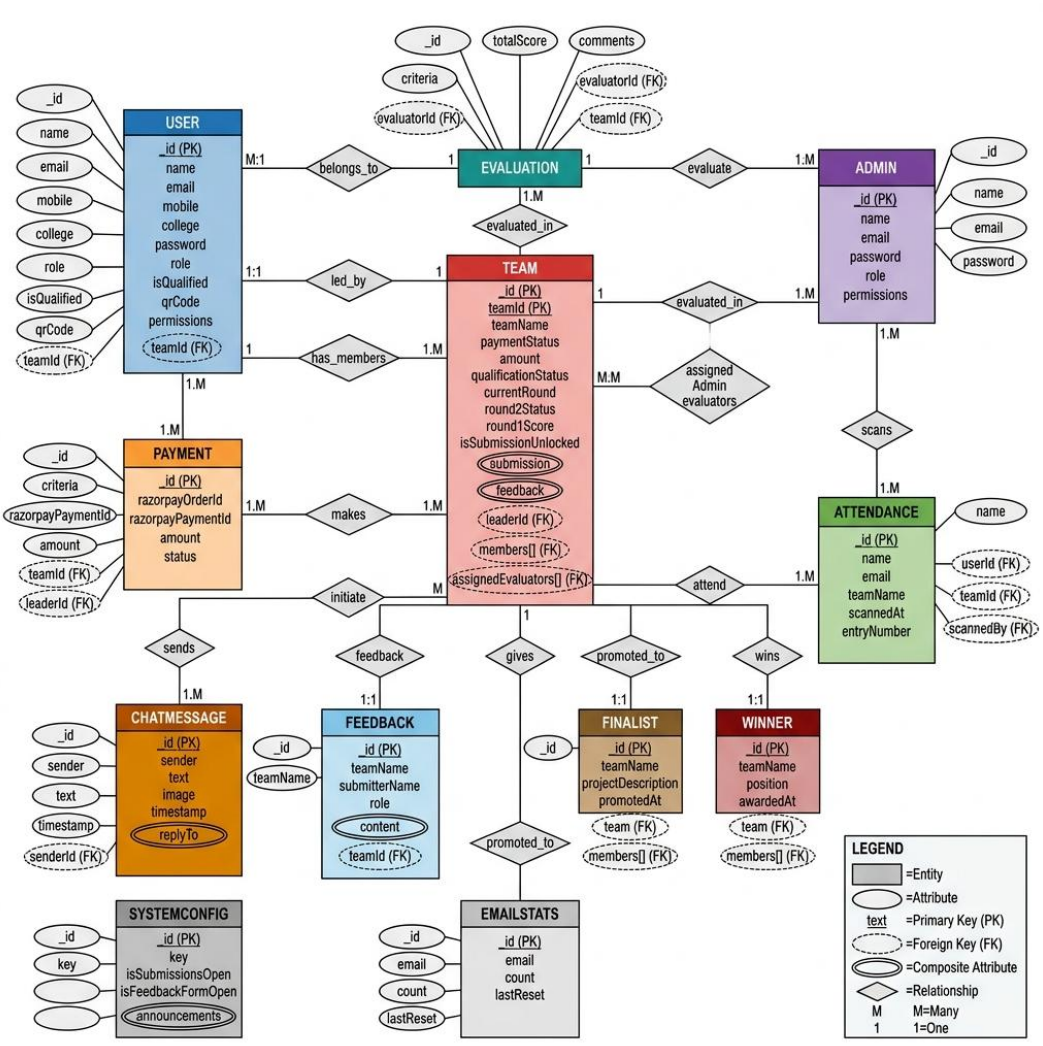
Entity	Description	Key Fields
User	Participants (Team Leaders & Members)	_id, name, email, role, teamId, isQualified, qrCode
Team	Hackathon teams with members and submission data	_id, teamId, teamName, leaderId, members, paymentStatus, submission
Admin	System administrators, viewers, and evaluators	_id, name, email, role, permissions
Evaluation	Scoring records per evaluator per team per round	_id, evaluatorId, teamId, criteria, totalScore, evaluationRound
Payment	Payment transaction records (Razorpay)	_id, teamId, leaderId, razorpayOrderId, amount, status
Attendance	QR scan-based attendance records	_id, userId, teamId, scannedBy, scannedAt, entryNumber
ChatMessage	Community chat messages	_id, sender, senderId, text, image, replyTo, timestamp
SystemConfig	Feature toggles and announcement settings	key, announcements, visibility toggles, flash messages

Feedback	Event feedback from participants (1-5 ratings)	_id, teamName, submitterName, content
Finalist	Teams promoted to grand finale	_id, team, teamName, members, projectDescription
Winner	Final winners with positions	_id, team, teamName, position (1st/2nd/3rd)
EmailStats	Daily email send tracking per address	_id, email, count, lastReset

2.2.2 Key Relationships

Relationship	Cardinality	Description
User → Team	Many-to-One	Each user belongs to at most one team
Team → User (leader)	One-to-One	Each team has exactly one leader
Team → User (members)	One-to-Many	Each team has multiple members (up to 4)
Team → Evaluation	One-to-Many	A team can have multiple evaluations (per evaluator, per round)
Admin → Evaluation	One-to-Many	An evaluator submits scores for multiple teams
Team → Payment	One-to-Many	Multiple payment attempts possible per team
User → Attendance	One-to-Many	Multiple scan entries per user (re-entry tracking)
Team → Finalist	One-to-One	A team can be promoted to finalist once
Team → Winner	One-to-One	A team can win one position
Team → Feedback	One-to-One	Each team submits feedback once

Figure 2.2: Entity Relationship Diagram (Chen Notation)

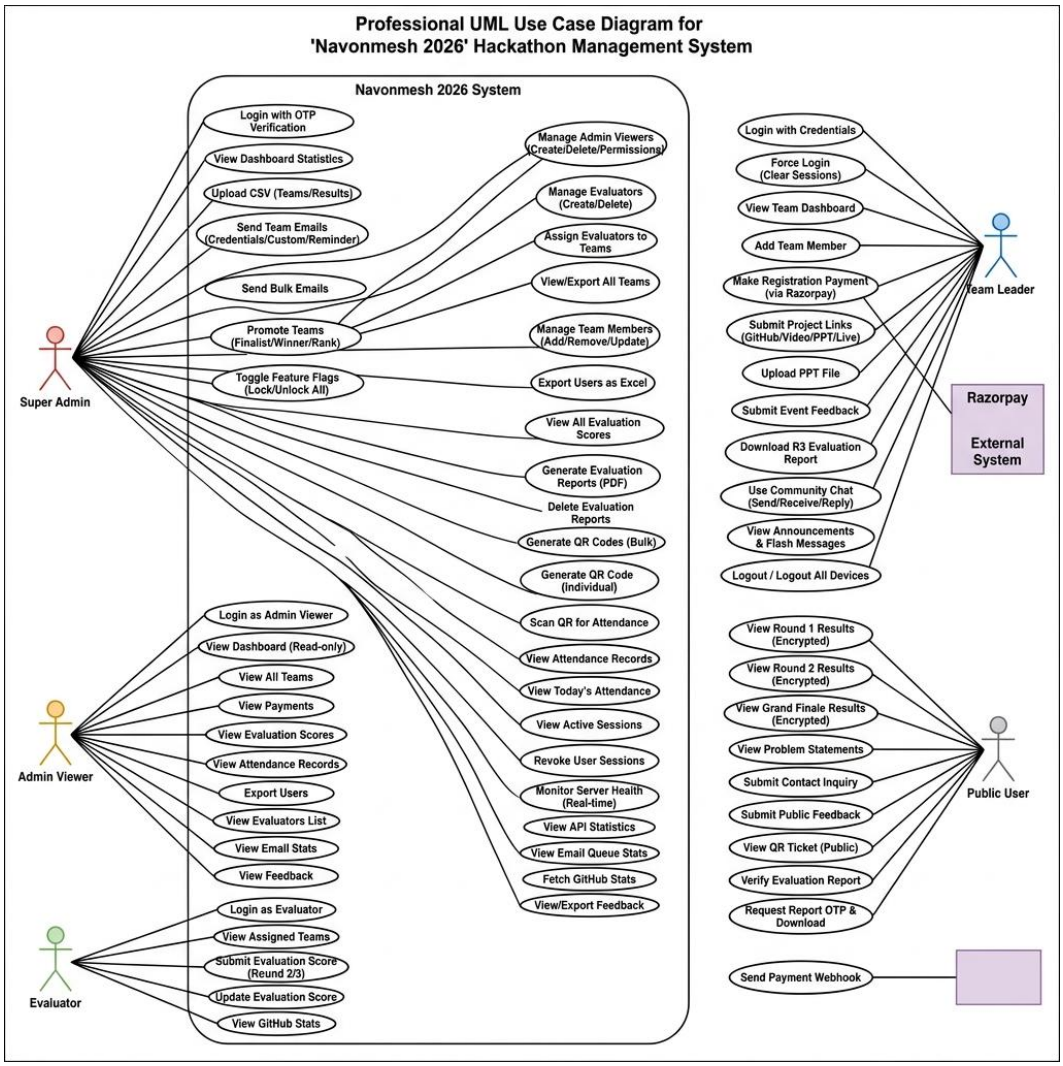


The above diagram uses Chen Notation: Rectangles represent entities, ovals represent attributes (underlined = Primary Key, dashed = Foreign Key, double oval = Composite Attribute), and diamonds represent relationships with cardinality labels (1:M, 1:1, M:M).

2.3 Use Case Diagram

The Navonmesh 2026 platform supports seven distinct actor types. Each use case listed below is directly mapped to an actual controller function and API route in the codebase, ensuring complete traceability between the design and implementation.

Figure 2.3: Use Case Diagram



2.4 Data Dictionary

The following data dictionary provides a comprehensive description of all collections, their fields, data types, constraints, and relationships in the Navonmesh 2026 database.

2.4.1 User Collection

Field	Data Type	Description	Constraints
_id	ObjectId	Auto-generated primary key	PK, Auto
name	String	Full name of the user	Required, Trimmed
email	String	Email address (login credential)	Required, Unique, Lowercase
mobile	String	Mobile phone number	Default: ""
college	String	College/Institution name	Default: ""
password	String	Bcrypt-hashed password	Required
plainPassword	String	Readable password (admin credential resend)	Nullable
role	String (Enum)	User role	Enum: ADMIN, TEAM_LEADER, TEAM_MEMBER, ADMIN_VIEWER
teamId	ObjectId (FK)	Reference to the user's team	FK → Team, Nullable
isQualified	Boolean	Whether user is qualified	Default: true
qrCode	String	Encrypted QR code data	Nullable
permissions	[String]	Granular permissions	Enum: MANAGE_TEAMS, SEND_EMAILS, etc.
createdAt	Date	Record creation timestamp	Auto (Mongoose)
updatedAt	Date	Last update timestamp	Auto (Mongoose)

2.4.2 Team Collection

Field	Data Type	Description	Constraints
_id	ObjectId	Auto-generated primary key	PK, Auto
teamId	String	Custom team ID (e.g., NVM-001)	Unique, Sparse, Trimmed
teamName	String	Display name of the team	Required, Trimmed
leaderId	ObjectId (FK)	Reference to the team leader	FK → User, Nullable
members	[ObjectId]	Array of team member references	FK → User[]
paymentStatus	String (Enum)	Payment state	Enum: PAID, UNPAID. Default: UNPAID
amount	Number	Payment amount (lowest currency unit)	Default: 0
razorpayOrderId	String	Razorpay order identifier	Optional
isQualified	Boolean	Whether team passed Round 1	Default: false
qualificationStatus	String (Enum)	Round 1 status	Enum: PENDING, QUALIFIED, DISQUALIFIED, WILDCARD
submission	Object	Project submission links	Contains: githubLink, videoLink, pptLink, liveLink, domain
assignedEvaluators	[ObjectId]	Round 2 evaluators	FK → Admin[]
assignedEvaluatorsR3	[ObjectId]	Round 3 evaluators	FK → Admin[]
currentRound	Number	Current competition round	Default: 1 (1, 2, or 3)
round2Status	String (Enum)	Round 2 qualification	Enum: PENDING, QUALIFIED, DISQUALIFIED
round1Score	Number	Score from Round 1	Default: 0.0, Min: 0, Max: 10

feedback	Object	Team's event feedback	Contains: submitted, content, submittedAt

2.4.3 Evaluation Collection

Field	Data Type	Description	Constraints
_id	ObjectId	Auto-generated primary key	PK, Auto
evaluatorId	ObjectId (FK)	The evaluator who scored	Required, FK → Admin
teamId	ObjectId (FK)	The team being evaluated	Required, FK → Team
criteria	Map<String, Number>	Scoring criteria key-value pairs	Required
totalScore	Number	Sum of all criteria scores	Required, Min: 0, Max: 100
comments	String	Evaluator's written feedback	Default: ""
evaluationRound	Number (Enum)	Evaluation round	Required, Enum: 2, 3

Unique Index: {evaluatorId, teamId, evaluationRound} — ensures one evaluation per team per evaluator per round.

2.4.4 Payment Collection

Field	Data Type	Description	Constraints
_id	ObjectId	Auto-generated primary key	PK, Auto
teamId	ObjectId (FK)	Team making the payment	Required, FK → Team
leaderId	ObjectId (FK)	Team leader initiating payment	Required, FK → User
razorpayOrderId	String	Razorpay order ID	Required
razorpayPaymentId	String	Razorpay payment ID	Optional

razorpaySignature	String	Razorpay HMAC signature	Optional
amount	Number	Amount in paise (INR × 100)	Required
status	String (Enum)	Payment processing state	Enum: PENDING, SUCCESS, FAILED, AUTHORIZED

2.4.5 Attendance Collection

Field	Data Type	Description	Constraints
_id	ObjectId	Auto-generated primary key	PK, Auto
userId	ObjectId (FK)	Attendee	Required, FK → User
teamId	ObjectId (FK)	Attendee's team	Required, FK → Team
name	String	Attendee's name (denormalized)	Required
email	String	Attendee's email (denormalized)	Required
teamName	String	Team name (denormalized)	Required
scannedBy	ObjectId (FK)	Admin who scanned	Required, FK → Admin
scannedByName	String	Scanner's name	Required
scannerRole	String (Enum)	Role of scanner	Enum: ADMIN, ADMIN_VIEWER
scannedAt	Date	Scan timestamp	Default: Date.now
entryNumber	Number	Which entry (1st, 2nd, etc.)	Default: 1

2.4.6 Additional Collections

Collection	Key Fields	Description
Admin	name, email, password, role, permissions	System administrators, viewers, and evaluators with RBAC

ChatMessage	sender, senderId, text, image, replyTo, timestamp	Community chat messages with reply threading
SystemConfig	key, announcements, visibility, flash messages	Singleton config document for feature toggles
Feedback	teamName, submitterName, role, content	Event feedback with Likert scale ratings (1-5)
Finalist	team (unique FK), teamName, members, projectDescription	Teams promoted to grand finale
Winner	team (unique FK), teamName, position, awardedAt	Final winners (1st, 2nd, 3rd)
EmailStats	email (unique), count, lastReset	Daily email send tracking per address

CHAPTER 3

TESTING

This chapter documents the testing strategy, test cases, and defect reports for the Navonmesh 2026 platform. Testing was conducted across all major modules including authentication, team management, payment processing, evaluation, QR attendance, and administrative functions.

3.1 Test Cases

3.1.1 Authentication Module Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-001	Valid Team Leader Login	User account exists in DB	1. Enter valid email 2. Enter valid password 3. Select 'Team Leader' role 4. Click Login	User redirected to Team Dashboard with JWT token stored	Pass
TC-002	Invalid Password Login	User account exists	1. Enter valid email 2. Enter wrong password 3. Click Login	Error message: 'Invalid credentials'. Failed attempt counter incremented in Redis	Pass
TC-003	CAPTCHA After 3 Failures	3 failed login attempts from same IP+Device	1. Attempt 4th login	reCAPTCHA widget appears and must be completed before login	Pass
TC-004	Admin OTP Login	Admin env credentials configured	1. Enter admin email & password 2. Submit login 3. Enter OTP from email	OTP sent to email; valid OTP grants access with full permissions	Pass
TC-005	Session Conflict Detection	Team Leader has active session	1. Login from new device without force flag	Response: hasActiveSession=true;	Pass

				user prompted to force login or cancel	
TC-006	Force Login (Clear Sessions)	Active session exists	1. Login with force=true	Previous sessions cleared; new session created successfully	Pass
TC-007	Logout (Single Device)	User authenticated	1. Click logout	JWT session removed from Redis; token invalidated	Pass
TC-008	Logout All Devices	Multiple active sessions	1. Click 'Logout All Devices'	All sessions cleared from Redis	Pass

3.1.2 Payment Module Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-009	Create Payment Order	Team Leader logged in, team unpaid	1. Click 'Make Payment' 2. System calls Razorpay API	Razorpay order created; Payment record saved as PENDING	Pass
TC-010	Successful Payment	Order created	1. Complete payment in Razorpay modal 2. System verifies HMAC signature	Payment status → SUCCESS; Team paymentStatus → PAID	Pass
TC-011	Payment Webhook Verification	Payment completed	1. Razorpay sends webhook 2. Server verifies signature	Idempotent update; payment and team status confirmed	Pass
TC-012	Failed Payment Retry	Payment failed	1. Retry payment	New order created; previous order unaffected	Pass

TC-013	Duplicate Payment Prevention	Team already PAID	1. Attempt payment again	System rejects with message: 'Payment already completed'	Pass

3.1.3 Team Management Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-014	Add Team Member	Team exists, < 4 members	1. Enter member email 2. Submit	Member added to team; user account created if new	Pass
TC-015	Add 5th Member (Limit)	Team has 4 members	1. Attempt to add 5th member	Error: 'Maximum team size reached'	Pass
TC-016	Submit Project Links	Submissions unlocked by admin	1. Enter GitHub, video, PPT links 2. Select domain 3. Submit	Submission data saved to team record	Pass
TC-017	Upload PPT File	Submissions open	1. Upload .ppt/.pptx file via form	File uploaded via Multer; path saved to team submission	Pass
TC-018	Submit When Locked	Submissions not unlocked	1. Attempt to submit links	Error: 'Submissions are currently closed'	Pass

3.1.4 Evaluation Module Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-019	Submit Evaluation Score	Evaluator assigned, access enabled	1. Select team 2. Score each criterion (0-10) 3. Submit	Evaluation record created/updated; totalScore calculated	Pass

TC-020	Re-submit Score (Update)	Evaluation already submitted	1. Modify scores 2. Re-submit	Existing evaluation updated (upsert operation)	Pass
TC-021	Score When Access Disabled	evaluatorR2Access.enabled = false	1. Attempt to submit score	Error: 'Evaluation access is currently disabled'	Pass
TC-022	Invalid Score Range	Evaluator logged in	1. Enter score > 10 for a criterion	Validation error: 'Score must be between 0 and 10'	Pass

3.1.5 QR & Attendance Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-023	Generate QR for Paid Users	Teams with PAID status exist	1. Admin clicks 'Generate All QR' 2. System generates encrypted QR	QR codes generated for all paid team members; emails queued	Pass
TC-024	Scan Valid QR Code	QR generated, admin logged in	1. Scan QR code with camera	Attendance recorded; entry number incremented if re-entry	Pass
TC-025	Scan Invalid QR Code	Admin logged in	1. Scan tampered/invalid QR	Error: 'Invalid QR code'; HMAC verification fails	Pass
TC-026	View Attendance Report	Attendance records exist	1. Navigate to attendance report	Filtered list of attendance records displayed	Pass
TC-027	Public Ticket View	QR generated for user	1. Access public ticket URL	Ticket page displays team name, member name, and QR code	Pass

3.1.6 Admin & Settings Test Cases

TC ID	Test Case	Pre-Conditions	Steps	Expected Result	Status
TC-028	Upload CSV for Teams	Admin logged in with MANAGE_TEAMS	1. Upload CSV file 2. System processes	Teams and user accounts created; stats returned	Pass
TC-029	Toggle Feature Flag	Admin with TOGGLE_SUBMISSIONS	1. Toggle R1 Announcement 2. Save	SystemConfig updated; Socket.IO broadcasts change to all clients	Pass
TC-030	Promote Team to Finalist	Admin with PROMOTE_TEAMS	1. Select team 2. Click 'Promote'	Finalist record created; team currentRound updated	Pass
TC-031	Permission-Denied Access	Admin Viewer without SEND_EMAILS	1. Attempt bulk email	403 Forbidden: Insufficient permissions	Pass
TC-032	Revoke User Session	Admin with MANAGE_USERS	1. View active sessions 2. Revoke target	Session removed from Redis; user logged out	Pass

3.2 Defect Report

The following table documents defects identified during system testing, along with their severity, root cause, and resolution status.

Defect ID	Module	Severity	Description	Root Cause	Resolution	Status
DEF-001	Authentication	High	Team Leader could login on multiple devices simultaneously without force flag	Redis session check query was not atomic	Implemented Redis SISMEMBER check before session creation	Resolved
DEF-002	Payment	Critical	Webhook payments were processed twice causing duplicate PAID status updates	Missing idempotency check in webhook handler	Added check for existing SUCCESS payment before processing	Resolved
DEF-003	Chat	Medium	Chat messages with large base64 images caused request timeout	Express body parser limit was set to 10MB default	Increased body limit to 100MB for chat endpoints	Resolved
DEF-004	Evaluation	Medium	Evaluator could submit scores for teams not assigned to them	Missing evaluator-team assignment validation	Added assignment verification in score submission middleware	Resolved
DEF-005	QR System	High	QR codes could be reused across different events due to missing	QR payload did not include	Added generatedAt timestamp to encrypted QR	Resolved

			timestamp validation	generation timestamp	payload with 24h validity	
DEF-006	Public API	Low	Public results page showed stale data for up to 5 minutes	Redis cache TTL was too long	Reduced cache TTL and added version-based cache invalidation on settings change	Resolved
DEF-007	Email	Medium	Bulk email queue would stall if Gmail SMTP rate limit was hit	Bull queue retry strategy did not have exponential backoff	Implemented exponential backoff with jitter in retry strategy	Resolved
DEF-008	Admin	Low	CSV upload with special characters in team names caused parsing errors	csv-parser did not handle UTF-8 BOM	Added BOM stripping and explicit UTF-8 encoding in CSV parser	Resolved

CHAPTER 4

LIMITATIONS OF PROPOSED SYSTEM

While the Navonmesh 2026 platform provides comprehensive hackathon management capabilities, the following limitations have been identified in the current implementation:

1. Single Payment Gateway: The system currently integrates only with Razorpay for payment processing. Users who prefer alternative payment methods (UPI apps, net banking through other gateways, international payment methods) may face inconvenience.

2. No AI-Based Evaluation: Project evaluation is entirely manual and depends on assigned evaluators. There is no automated code quality analysis, plagiarism detection, or AI-assisted scoring mechanism to supplement human judgment.

3. Limited Offline Capability: The platform requires a continuous internet connection for all operations. In case of network disruption during the event (especially QR scanning at the venue), attendance recording and real-time features become unavailable...

CHAPTER 5

PROPOSED ENHANCEMENTS

Based on the identified limitations and user feedback collected during the Navonmesh 2026 event, the following enhancements are proposed for future iterations of the platform:

1. Multi-Payment Gateway Support: Integrate additional payment gateways such as Stripe, PayPal, and PhonePe/Google Pay direct integrations. This would expand accessibility for international participants and provide fallback options in case of primary gateway downtime.

2. AI-Assisted Evaluation: Incorporate AI-powered tools for automated code quality analysis (SonarQube integration), plagiarism detection (MOSS algorithm), and preliminary scoring based on GitHub metrics (commit frequency, code complexity, test coverage). This would provide evaluators with data-driven insights to complement their manual assessment.

3. Progressive Web App (PWA) with Offline Support: Convert the web application to a PWA with service workers for offline access. Critical features like viewing team status, cached QR tickets, and pre-loaded problem statements should work without an internet connection.

4. SMS and WhatsApp Notifications: Add SMS (Twilio) and WhatsApp Business API integrations for critical notifications such as payment confirmations, round advancement alerts, and event-day reminders, reaching participants who may not be monitoring email.

CHAPTER 6

REFERENCES

The following resources, documentation, and technologies were referenced during the design, development, and deployment of the Navonmesh 2026 platform:

6.1 Framework and Library Documentation

- React.js Official Documentation — https://react.dev/
- TypeScript Handbook — https://www.typescriptlang.org/docs/
- Vite Build Tool — https://vitejs.dev/guide/
- Express.js v5 Documentation — https://expressjs.com/
- Node.js Documentation — https://nodejs.org/docs/latest/api/
- Socket.IO Documentation — https://socket.io/docs/v4/
- Mongoose ODM Documentation — https://mongoosejs.com/docs/
- TailwindCSS Documentation — https://tailwindcss.com/docs/
- ShadCN UI Components — https://ui.shadcn.com/
- Framer Motion — https://www.framer.com/motion/
- React Query (TanStack Query) — https://tanstack.com/query/latest/
- Recharts Charting Library — https://recharts.org/en-US/

6.2 Database and Caching

- MongoDB Official Documentation — https://www.mongodb.com/docs/
- MongoDB Atlas Documentation — https://www.mongodb.com/docs/atlas/
- Redis Documentation — https://redis.io/docs/
- ioredis Client Library — https://github.com/redis/ioredis
- Bull Queue Documentation — https://github.com/OptimalBits/bull

6.3 Security and Authentication

- JSON Web Tokens (JWT) — https://jwt.io/introduction
- bcryptjs Password Hashing — https://github.com/dcodeIO/bcrypt.js

• Helmet.js Security Headers — https://helmetjs.github.io/

• Google reCAPTCHA v2 — https://developers.google.com/recaptcha/docs/display

• OWASP Security Best Practices — https://owasp.org/www-project-web-security-testing-guide/

• express-rate-limit — https://github.com/express-rate-limit/express-rate-limit

6.4 Payment and External Services

• Razorpay API Documentation — https://razorpay.com/docs/api/

• Razorpay Webhooks — https://razorpay.com/docs/webhooks/

• Nodemailer Documentation — https://nodemailer.com/

• Google Gmail SMTP Settings — https://support.google.com/a/answer/176600

6.5 Monitoring and DevOps

• Prometheus Monitoring — https://prometheus.io/docs/

• Grafana Dashboard — https://grafana.com/docs/grafana/latest/

• prom-client (Node.js Prometheus Client) — https://github.com/siimon/prom-client

• Puppeteer (PDF Generation) — https://pptr.dev/

• Docker Documentation — https://docs.docker.com/

CHAPTER 7

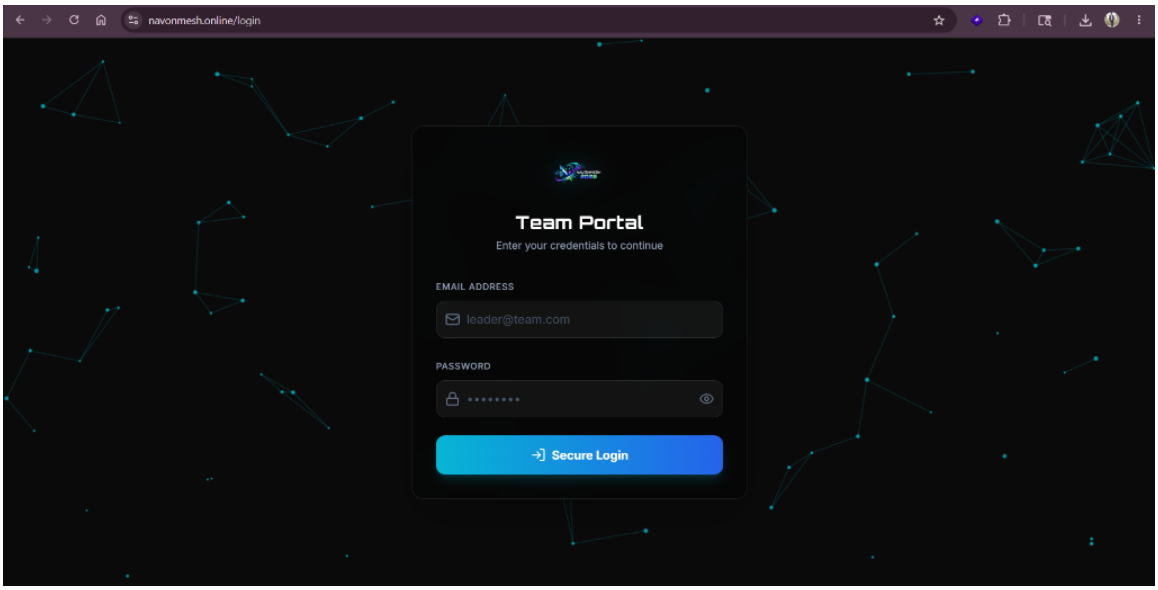
USER MANUAL

This chapter provides a visual walkthrough of the Navonmesh 2026 Hackathon Management Platform with step-by-step instructions for each screen.

7.1 Team Leader Guide

This section covers the Team Leader portal – login, dashboard, team management, submission, payment, and community chat features.

7.1.1 Team Leader Login

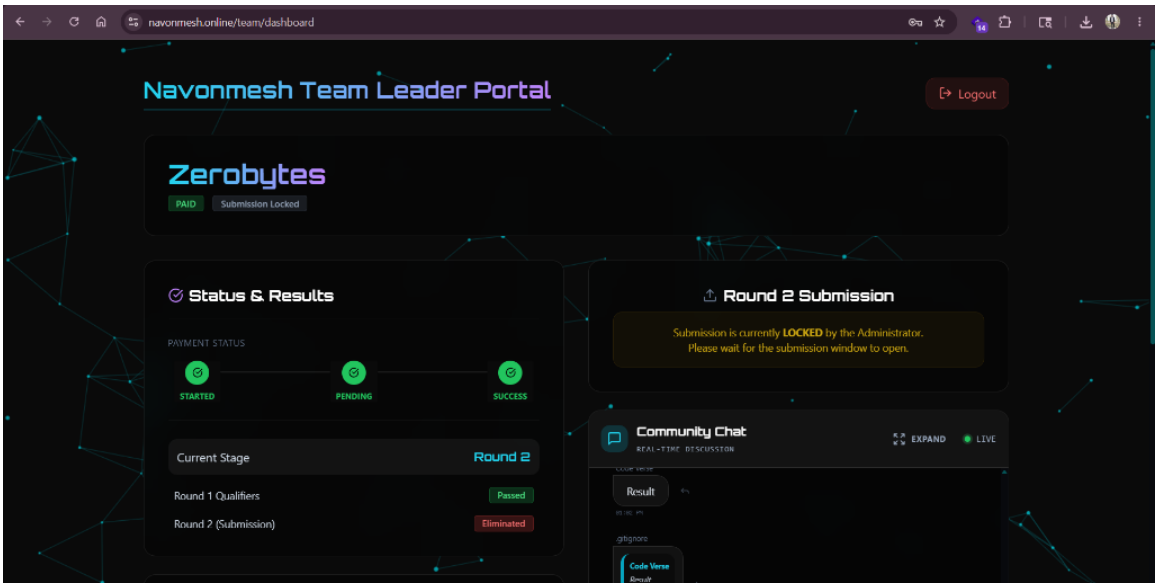


1. Navigate to navonmesh.online/login to access the Team Portal login page.
2. Enter the Email Address provided by the hackathon organizers via email in the 'EMAIL ADDRESS' field.
3. Enter the Password in the 'PASSWORD' field. Click the eye icon to toggle password visibility.
4. Click the 'Secure Login' button to authenticate.
5. If prompted with a CAPTCHA (after 3 failed attempts), complete the verification challenge before logging in.

6. If a session conflict is detected (you are logged in on another device), choose either 'Force Login' to terminate the previous session or 'Cancel' to keep the existing session.

7. Upon successful login, you will be redirected to the Team Leader Dashboard.

7.1.2 Team Leader Dashboard – Overview



1. After logging in, you are redirected to the Navonmesh Team Leader Portal at navonmesh.online/team/dashboard.

2. The top section displays your team name (e.g., 'Zerobytes') along with status badges: PAID/UNPAID and Submission Locked/Unlocked.

3. Click 'Logout' at the top right to securely log out of your session.

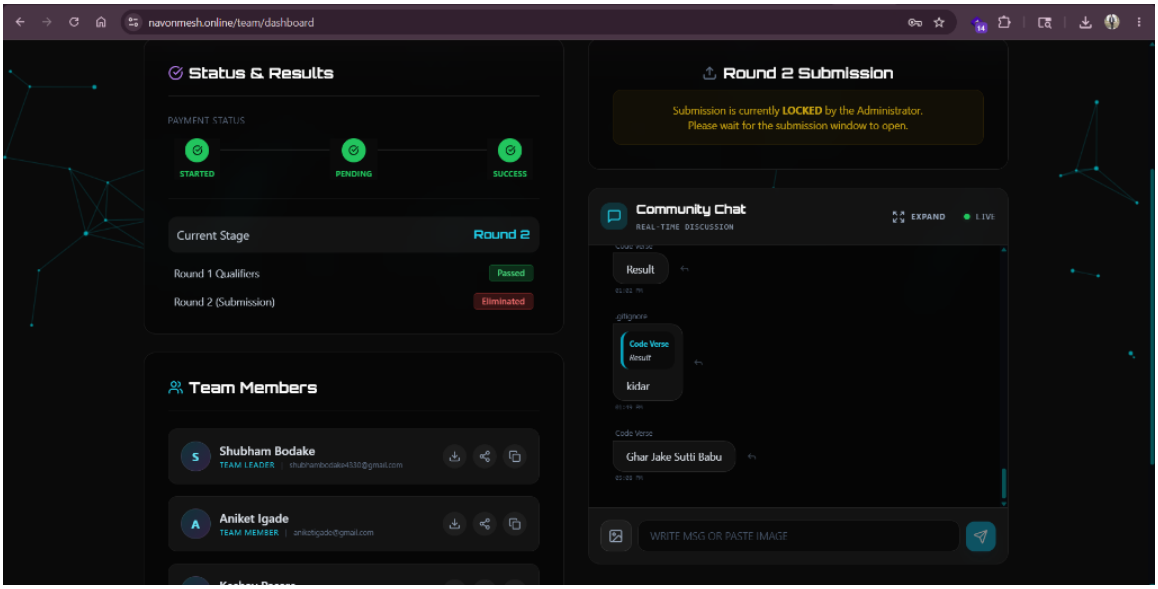
4. The 'Status & Results' section shows your Payment Status with three stages: STARTED, PENDING, and SUCCESS.

5. View your Current Stage (e.g., Round 2) and qualification results for each round (Passed/Eliminated badges).

6. The 'Round 2 Submission' section shows whether submissions are currently LOCKED or OPEN by the Administrator. When open, you can submit your project links here.

7. The 'Community Chat' panel on the right shows the real-time discussion. Click 'EXPAND' to open the full chat interface. The green 'LIVE' indicator confirms real-time connectivity.

7.1.3 Team Leader Dashboard – Team Members & Chat



1. Scroll down on the Team Leader Dashboard to view the 'Team Members' section on the left.

2. Each team member is displayed with their name, role badge (TEAM LEADER / TEAM MEMBER), and email address.

3. Use the action icons next to each member to: Download their QR ticket, Share their profile link, or Copy their details.

4. The Community Chat panel on the right shows the full chat with messages from all participating teams.

5. Type your message in the 'WRITE MSG OR PASTE IMAGE' input field at the bottom.

6. Click the send button (arrow icon) to post your message to the community chat.

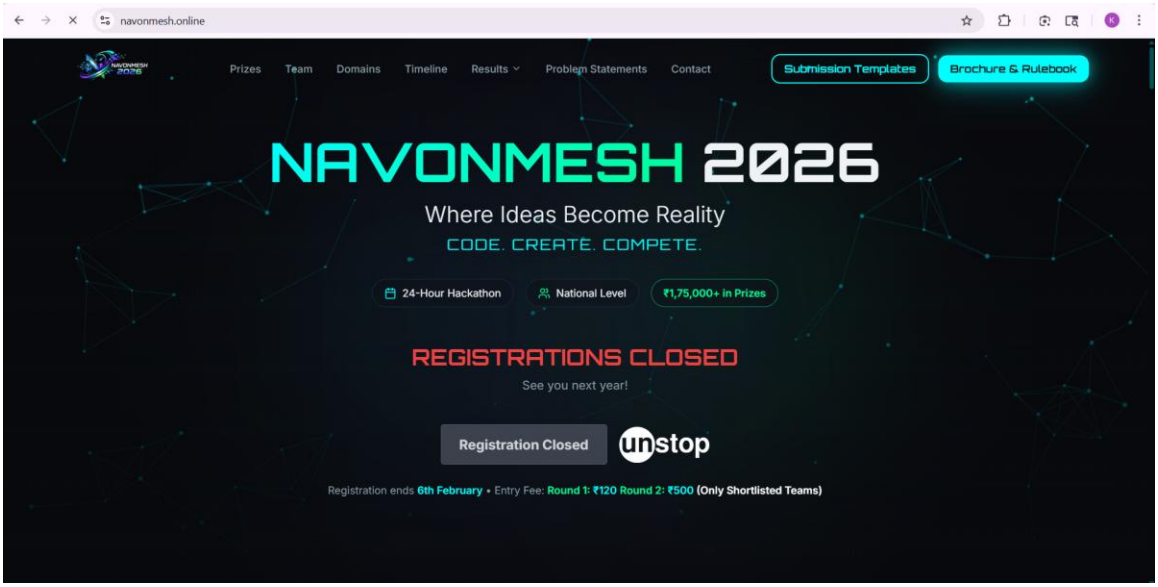
7. You can reply to specific messages, send images, and scroll up to load older messages.

8. Note: A spam protection mechanism limits you to 15 messages per 10 seconds.

7.2 Public Website

This section covers the public-facing website – landing page, prizes, domains, results, grand finale champions, and problem statements.

7.2.1 Landing Page (Hero Section)



1. Open a web browser and navigate to navonmesh.online to access the Navonmesh 2026 public website.

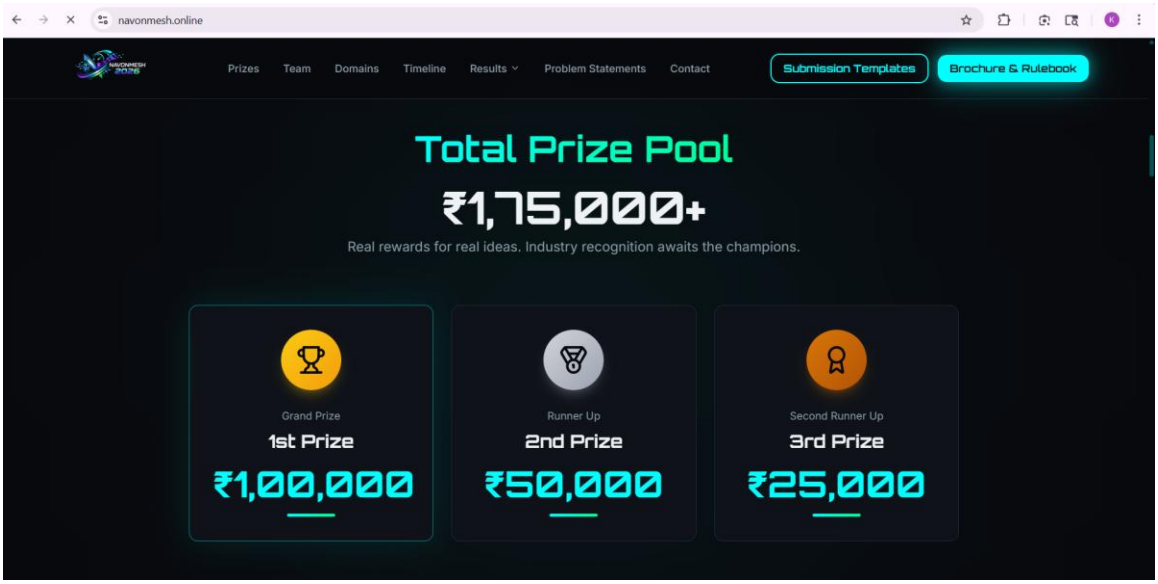
2. The Hero Section displays the event title 'NAVONMESH 2026' with the tagline 'Where Ideas Become Reality – CODE. CREATE. COMPETE.'

3. View the event highlights: 24-Hour Hackathon, National Level, and ₹1,75,000+ in Prizes.

4. Use the top navigation bar to access: Prizes, Team, Domains, Timeline, Results, Problem Statements, Contact, Submission Templates, and Brochure & Rulebook.

5. Check the registration status and entry fee details at the bottom of the hero section.

7.2.2 Prize Pool Section



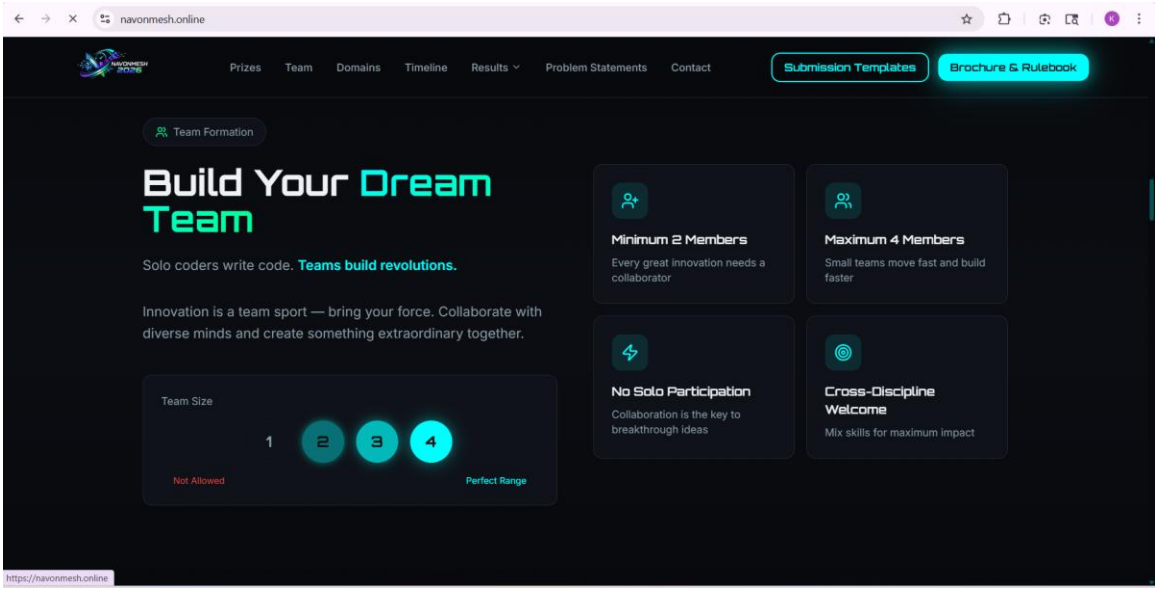
1. Scroll down from the Hero Section to view the Prize Pool section.

2. The total prize pool of ₹1,75,000+ is prominently displayed at the top.

3. View individual prizes: Grand Prize (1st Place) – ₹1,00,000, Runner Up (2nd Place) – ₹50,000, and Second Runner Up (3rd Place) – ₹25,000.

4. Each prize category is displayed in a distinct card with trophy and medal icons.

7.2.3 Team Formation Guidelines



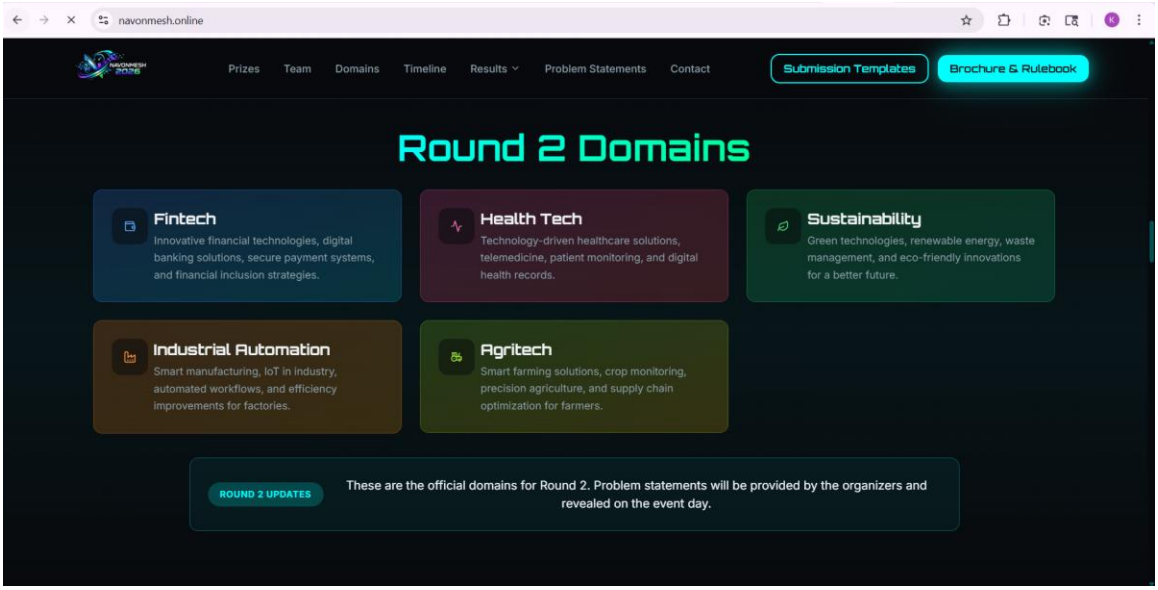
1. Navigate to the 'Team' section from the navigation bar or scroll down.

2. Review the team size requirements: Minimum 2 Members and Maximum 4 Members.

3. Note the key rules: No Solo Participation allowed; Cross-Discipline teams welcome.

4. Use the visual team size selector to understand the allowed range (2-4 members).

7.2.4 Round 2 Domains



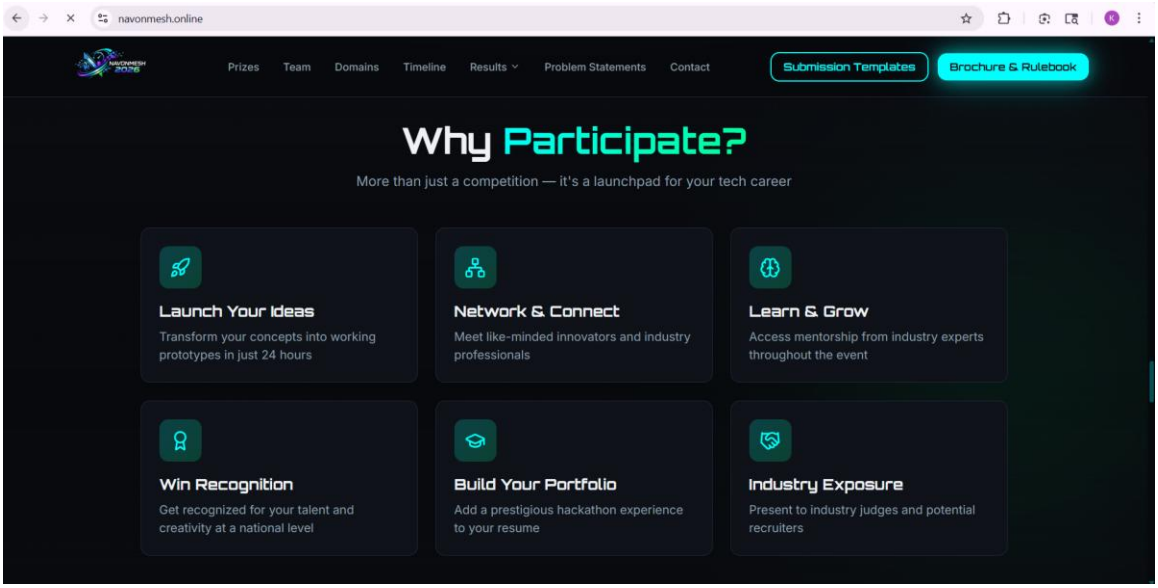
1. Scroll to the Domains section or click 'Domains' in the navigation bar.

2. Review the five competition domains: Fintech, Health Tech, Sustainability, Industrial Automation, and Agritech.

3. Each domain card displays a brief description of the problem area and expected solutions.

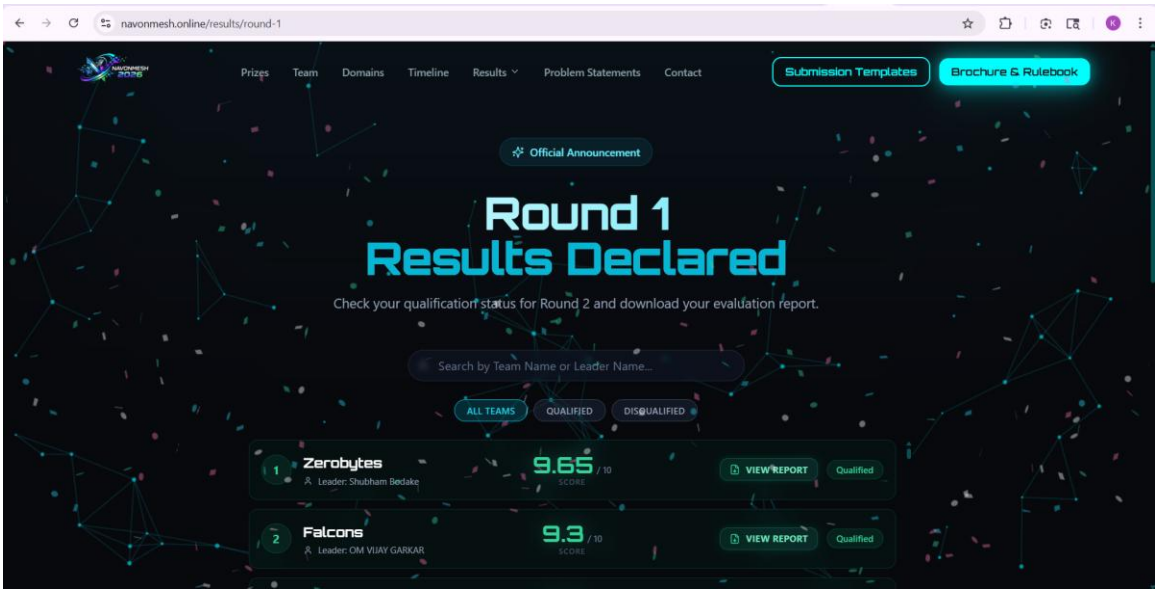
4. Click on any domain card to view the detailed problem statements available.

7.2.5 Why Participate Section



1. Scroll down to the 'Why Participate?' section.
2. Review six key benefits: Launch Your Ideas, Network & Connect, Learn & Grow, Win Recognition, Build Your Portfolio, and Industry Exposure.
3. Each benefit is displayed as a styled card with an icon and brief description.

7.2.6 Round 1 Results Page



1. Navigate to navonmesh.online/results/round-1 or click 'Results' → 'Round 1' from the navigation bar.

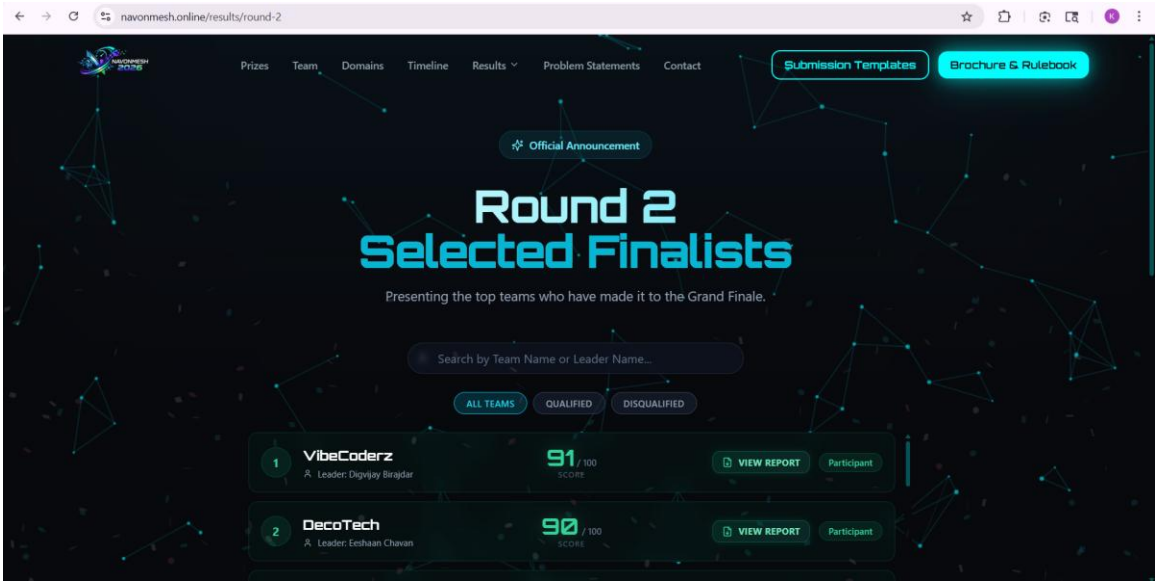
2. Use the search bar to search by Team Name or Leader Name.

3. Click filter tabs: ALL TEAMS, QUALIFIED, or DISQUALIFIED to filter the results.

4. Each team entry shows: rank, team name, leader name, score out of 10, and qualification status badge.

5. Click 'VIEW REPORT' to download the detailed evaluation report as a PDF.

7.2.7 Round 2 Selected Finalists Page



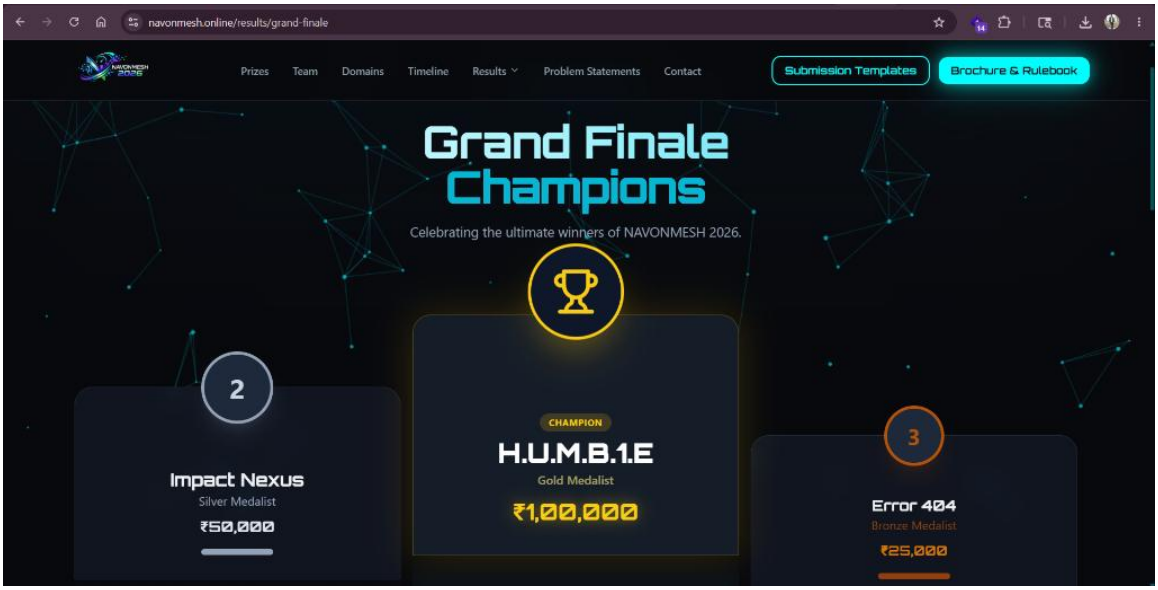
1. Navigate to navonmesh.online/results/round-2 or click 'Results' → 'Round 2' from the navigation bar.

2. Use the search bar and filter tabs (ALL TEAMS, QUALIFIED, DISQUALIFIED) to find specific teams.

3. Each team entry displays: rank, team name, leader name, and score out of 100.

4. Click 'VIEW REPORT' to download the Round 2 evaluation report.

7.2.8 Grand Finale Champions Page



1. Navigate to navonmesh.online/results/grand-finale or click 'Results' → 'Grand Finale' from the navigation bar.

2. The page displays 'Grand Finale Champions' with the subtitle 'Celebrating the ultimate winners of NAVONMESH 2026.'

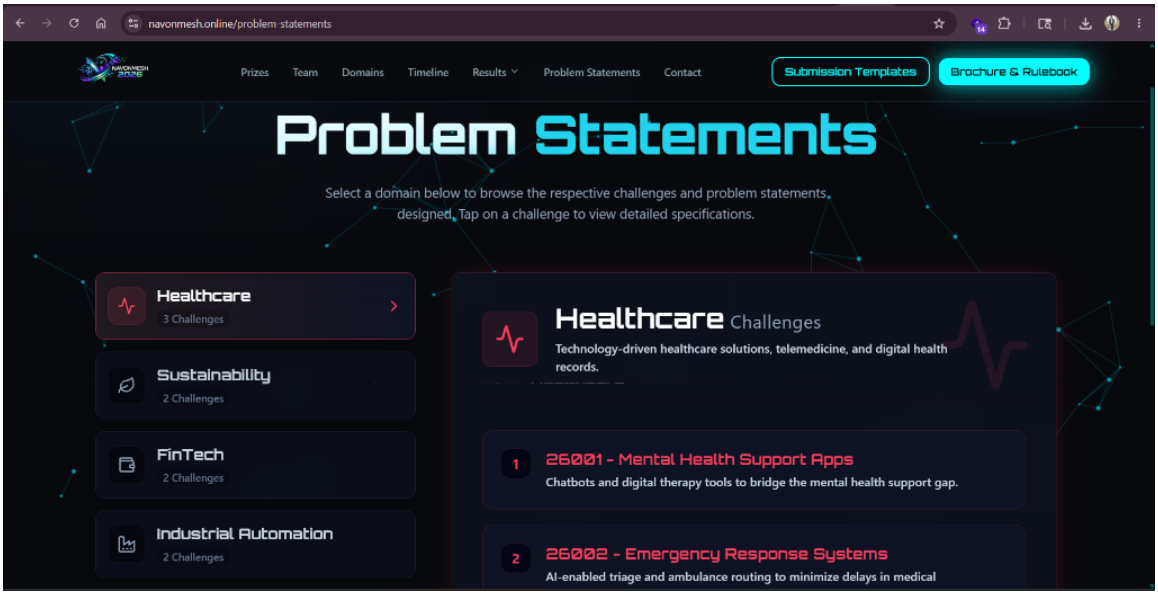
3. The Champion (1st Place – Gold Medalist) is displayed in the center with their team name and prize amount (₹1,00,000).

4. The Runner Up (2nd Place – Silver Medalist) is displayed on the left with prize amount (₹50,000).

5. The Second Runner Up (3rd Place – Bronze Medalist) is displayed on the right with prize amount (₹25,000).

6. Use the navigation bar at the top to access Submission Templates or Brochure & Rulebook.

7.2.9 Problem Statements Page



1. Navigate to navonmesh.online/problem-statements or click 'Problem Statements' from the navigation bar.

2. The left sidebar lists all available domains: Healthcare (3 Challenges), Sustainability (2 Challenges), FinTech (2 Challenges), and Industrial Automation (2 Challenges).

3. Click on any domain in the sidebar to view its challenges on the right panel.

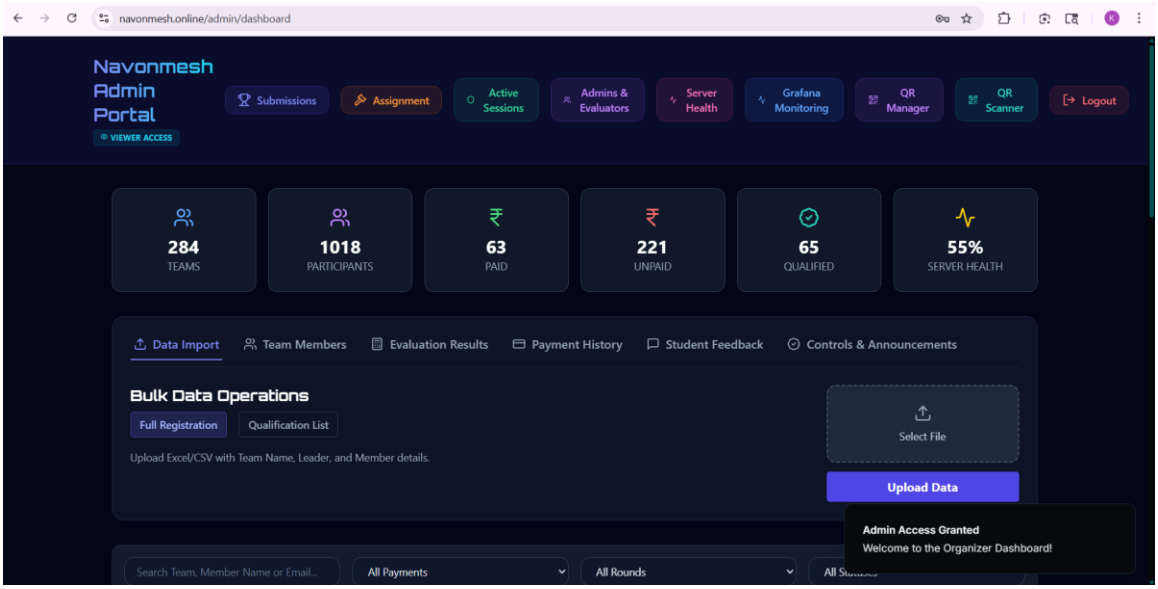
4. Each challenge shows a numbered ID (e.g., 26001), title (e.g., 'Mental Health Support Apps'), and brief description.

5. Tap on a challenge to view its detailed specifications and requirements.

7.3 Admin Panel

This section covers the admin dashboard – team management, evaluations, payments, QR attendance, server monitoring, and feature toggles.

7.3.1 Admin Dashboard Overview



1. Log in as Super Admin to access the Admin Dashboard at navonmesh.online/admin/dashboard.

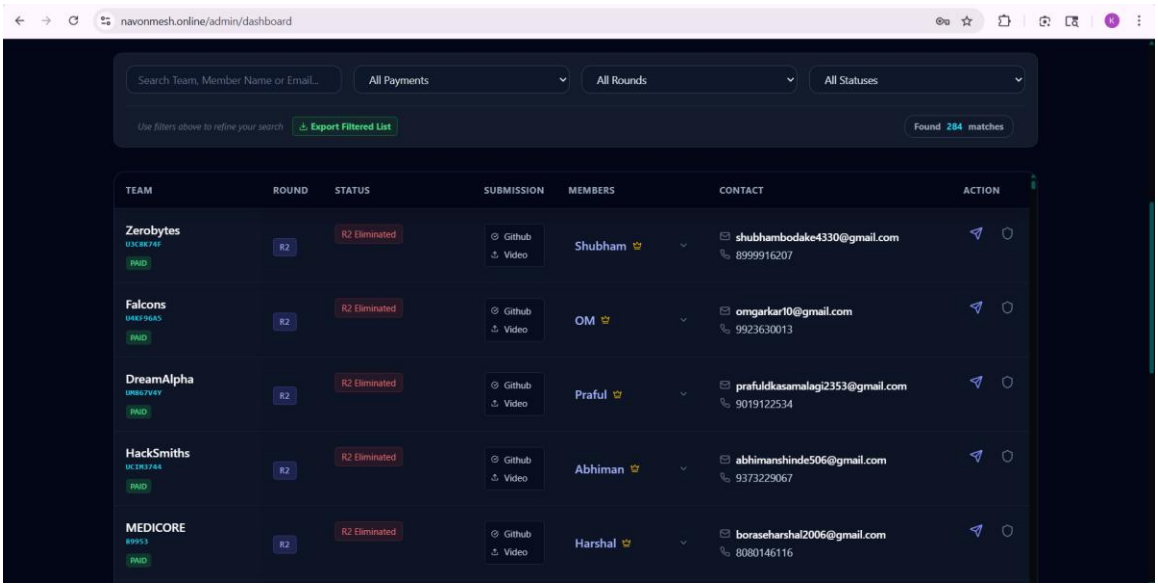
2. The top navigation bar provides quick-access: Submissions, Assignment, Active Sessions, Admins & Evaluators, Server Health, Grafana, QR Manager, QR Scanner, and Logout.

3. View six statistics cards: Total Teams, Participants, Paid, Unpaid, Qualified, and Server Health.

4. Use the tabbed sections below: Data Import, Team Members, Evaluation Results, Payment History, Student Feedback, Controls & Announcements.

5. In 'Data Import', upload CSV/Excel files for Full Registration or Qualification List imports.

7.3.2 Teams Table View



1. Click the 'Team Members' tab on the Admin Dashboard to view all registered teams.

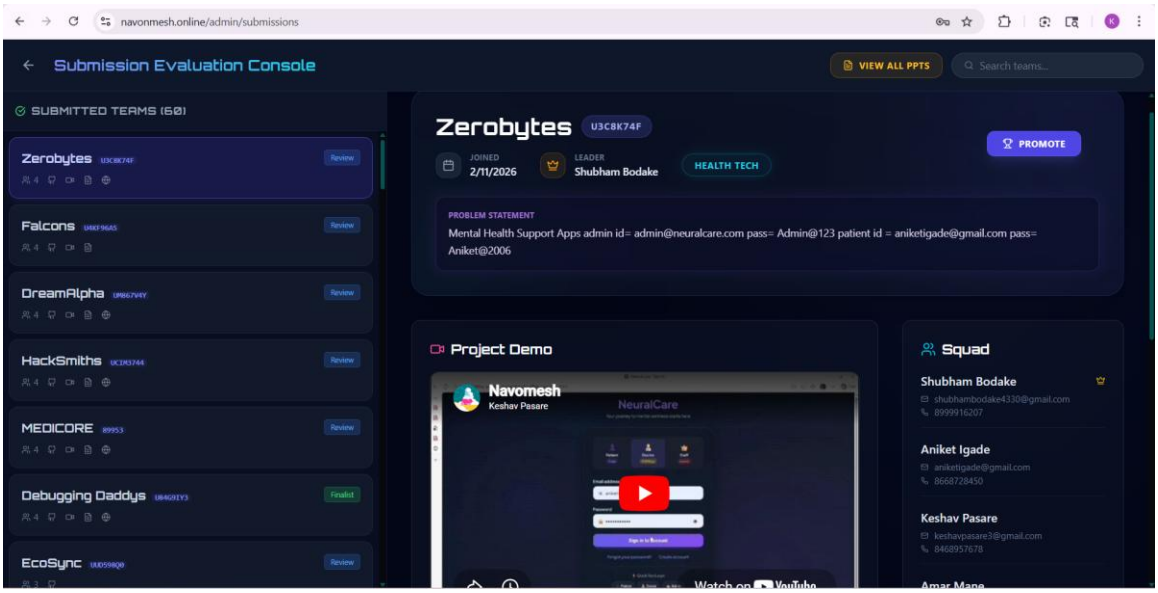
2. The table shows: Team (name, ID, payment badge), Round, Status, Submission links, Members, Contact, and Actions.

3. Use the search bar to search by team name, member name, or email.

4. Apply dropdown filters: Payment status, Round, and Status.

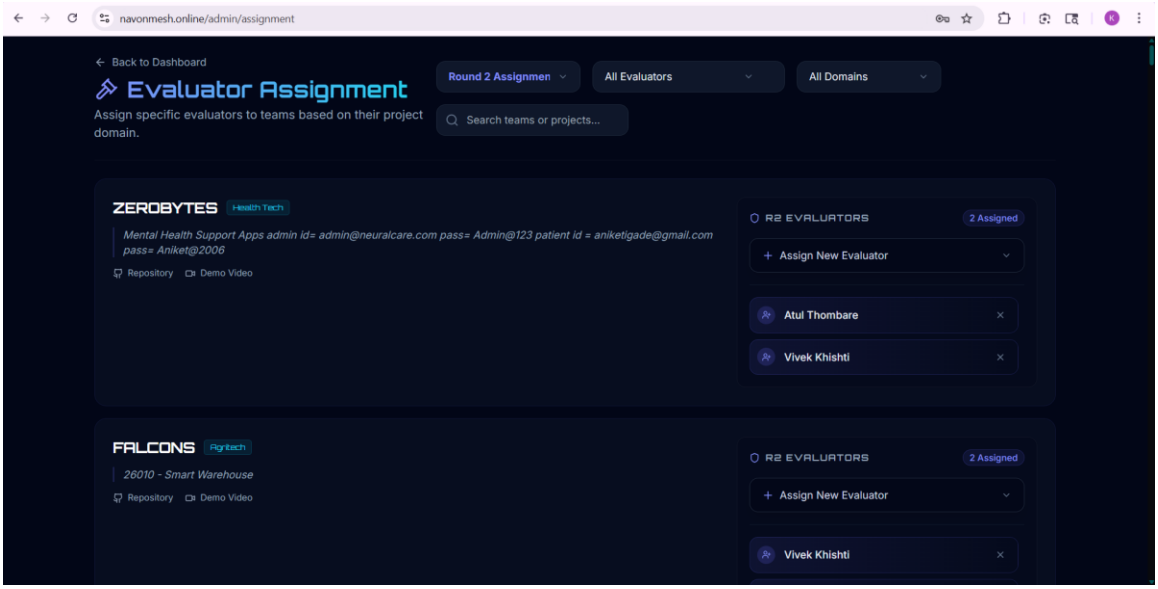
5. Click 'Export Filtered List' to download the filtered data as Excel.

7.3.3 Submission Evaluation Console



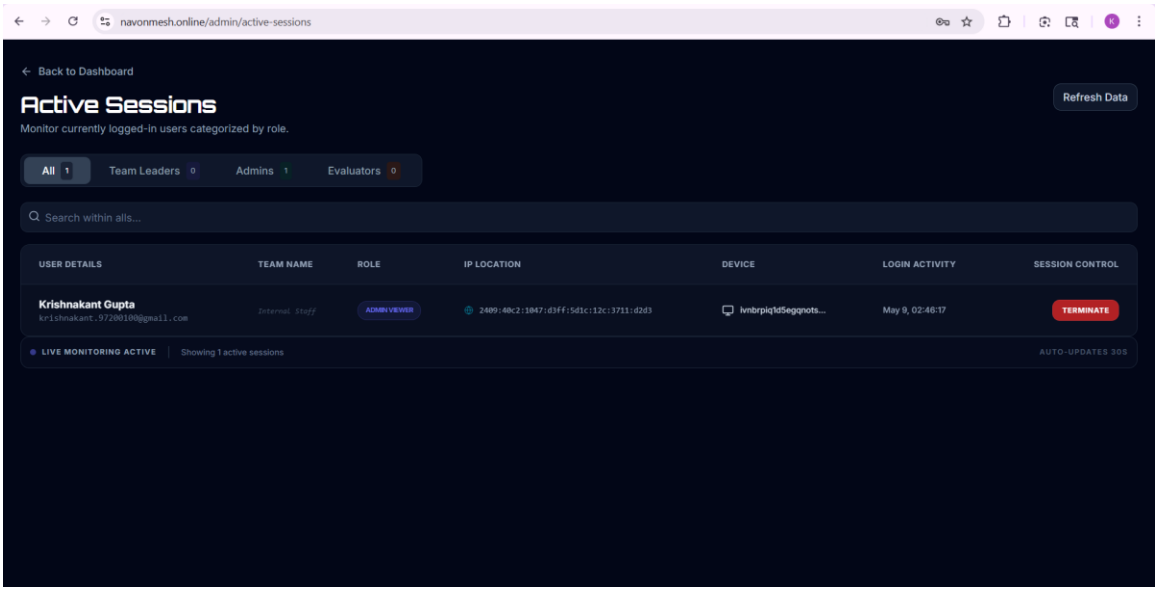
1. Click 'Submissions' from the top navigation to open navonmesh.online/admin/submissions.
2. The left panel lists all submitted teams with status badges (Review/Finalist).
3. Click a team to load their submission details, embedded demo video, and squad information in the right panel.
4. Click 'PROMOTE' to promote a team to Finalist or Winner status.
5. Click 'VIEW ALL PPTS' to browse all submitted presentation files.

7.3.4 Evaluator Assignment Page



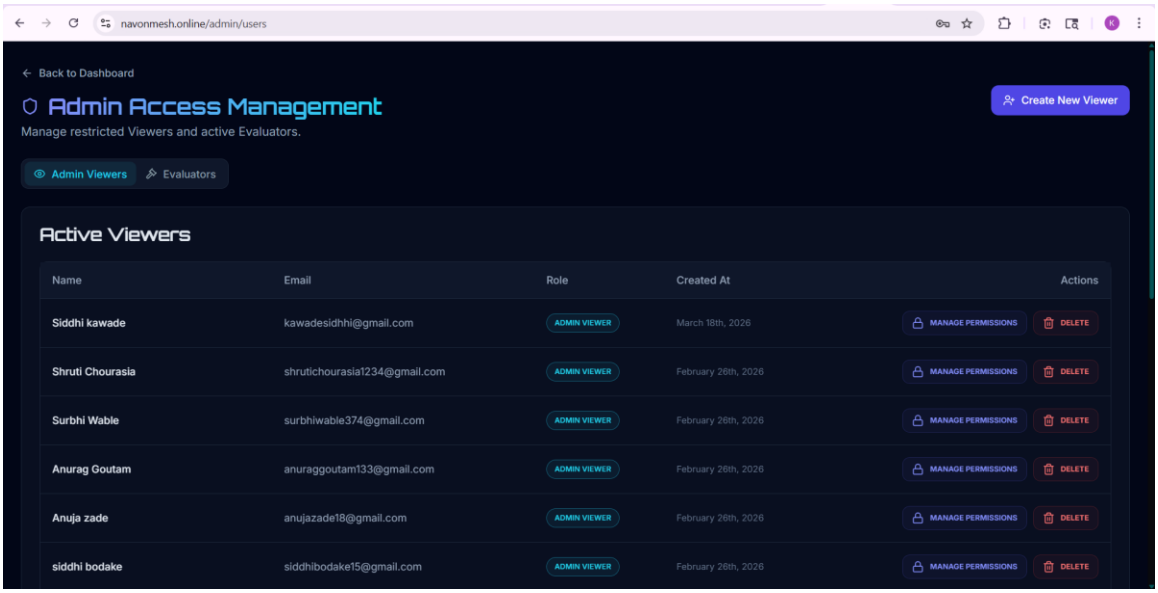
1. Click 'Assignment' from the navigation to access navonmesh.online/admin/assignment.
2. Filter by Round (Round 2/3), specific evaluators, and domains.
3. Each team card shows team name, domain, problem statement, and Repository/Demo links.
4. Use the evaluator dropdown on each card to add or remove assigned evaluators.

7.3.5 Active Sessions Monitor



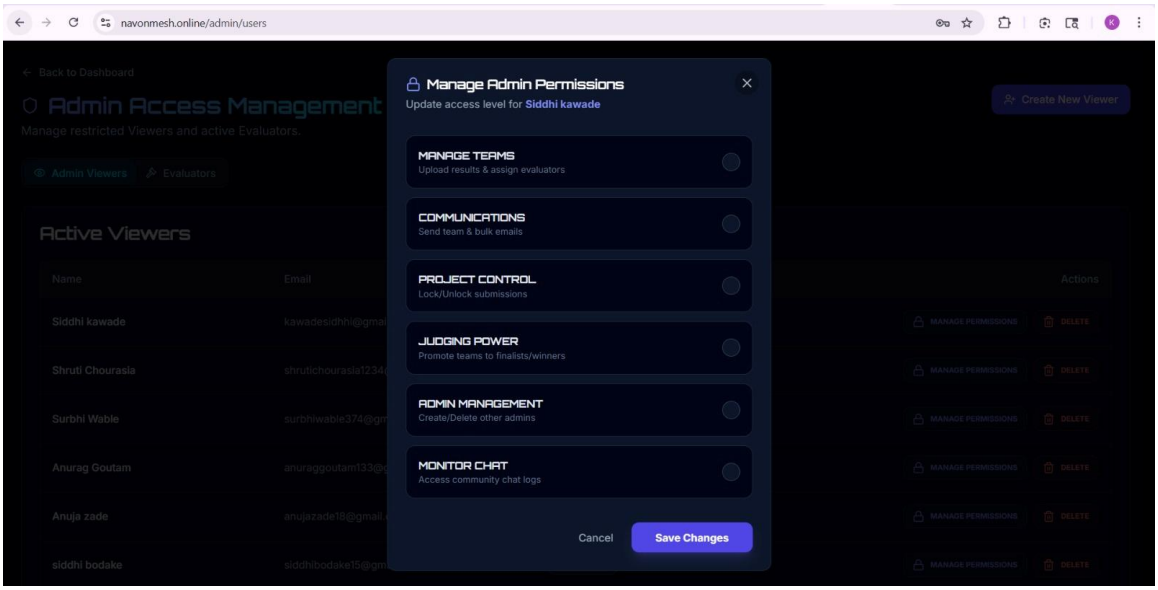
1. Click 'Active Sessions' from the navigation to access navonmesh.online/admin/active-sessions.
2. View logged-in users by role tabs: All, Team Leaders, Admins, Evaluators.
3. The table shows: User Details, Team Name, Role, IP Location, Device, Login Time, and Session Control.
4. Click 'TERMINATE' to forcefully log out any user.
5. The 'LIVE MONITORING ACTIVE' indicator confirms auto-refresh every 30 seconds.

7.3.6 Admin Access Management – Admin Viewers



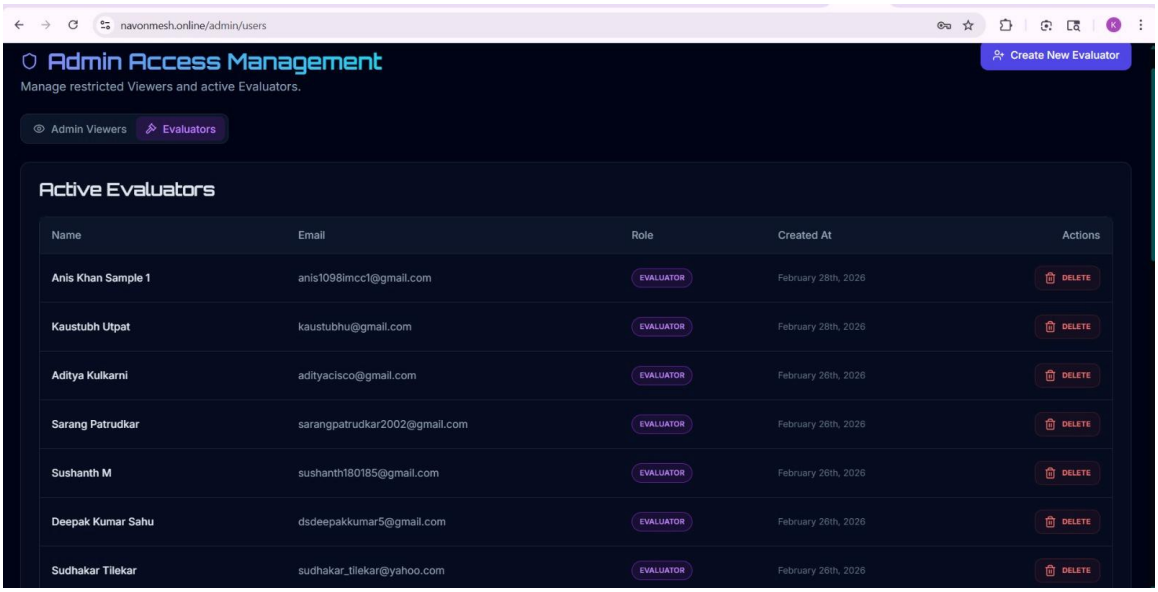
1. Click 'Admins & Evaluators' from the navigation to access navonmesh.online/admin/users.
2. The 'Admin Viewers' tab shows all restricted admin viewers with Name, Email, Role, Created At, and Actions.
3. Click 'Create New Viewer' to add a new admin viewer account.
4. Click 'MANAGE PERMISSIONS' to configure granular access for a viewer.
5. Click 'DELETE' to remove an admin viewer account.

7.3.7 Managing Admin Permissions



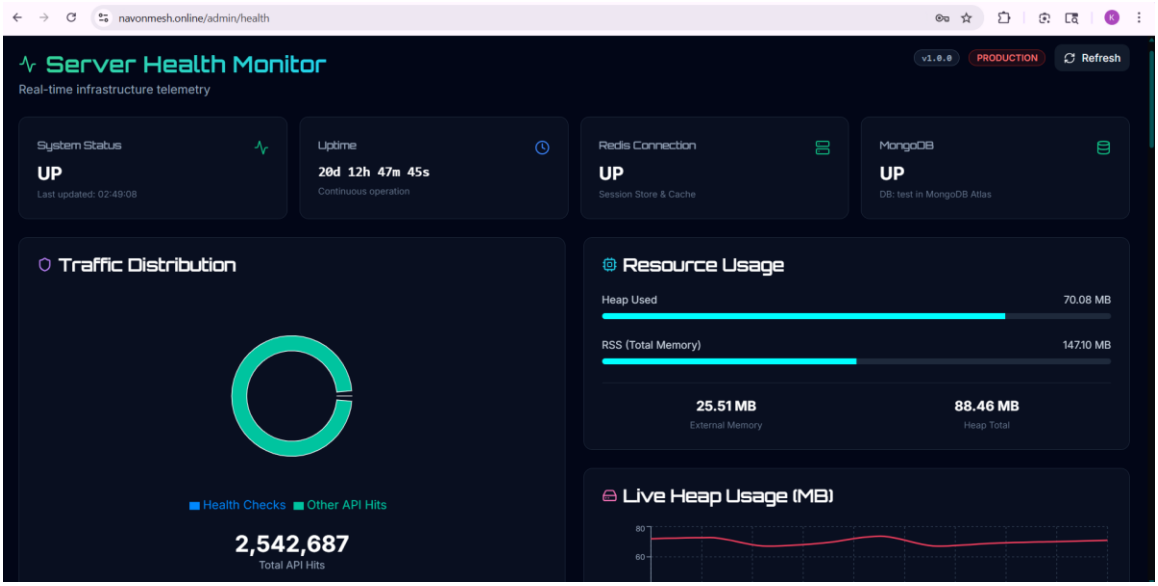
1. From the Admin Viewers tab, click 'MANAGE PERMISSIONS' to open the permissions modal.
2. Toggle each permission ON or OFF:
   a. MANAGE TEAMS – Upload results & assign evaluators.
   b. COMMUNICATIONS – Send team & bulk emails.
   c. PROJECT CONTROL – Lock/Unlock submissions.
   d. JUDGING POWER – Promote teams to finalists/winners.
   e. ADMIN MANAGEMENT – Create/Delete other admins.
   f. MONITOR CHAT – Access community chat logs.
3. Click 'Save Changes' to apply, or 'Cancel' to discard.

7.3.8 Admin Access Management – Evaluators Tab



1. Click the 'Evaluators' tab on the Admin Access Management page.
2. View all active evaluators with Name, Email, Role (EVALUATOR badge), Created At, and Actions.
3. Click 'Create New Evaluator' to add a new evaluator with auto-generated credentials.
4. Share the generated credentials with the evaluator securely.
5. Click 'DELETE' to remove an evaluator account (submitted evaluations are preserved).

7.3.9 Server Health Monitor – Overview



1. Click 'Server Health' from the navigation to access navonmesh.online/admin/health.

2. View four status cards: System Status, Uptime, Redis Connection, and MongoDB status.

3. The Traffic Distribution donut chart shows Health Checks vs. Other API Hits.

4. Resource Usage panel displays: Heap Used, RSS Total Memory, External Memory, and Heap Total.

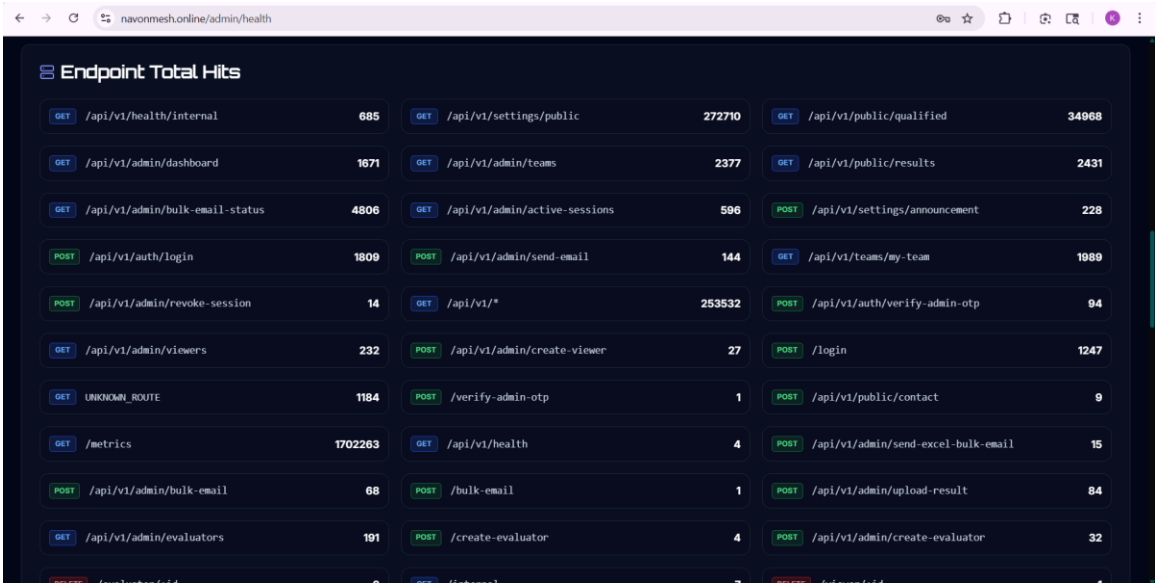
5. Monitor the Live Heap Usage chart for real-time memory trends.

7.3.10 Server Health Monitor – API Traffic



1. Scroll down on the Server Health page to view the Real-time API Traffic chart.
2. The chart visualizes request rates per second across endpoints with color-coded lines.
3. Below, the Endpoint Total Hits section shows cumulative hit counts per API endpoint.
4. Use this data to identify high-traffic endpoints and optimize performance.

7.3.11 Server Health Monitor – Endpoint Total Hits



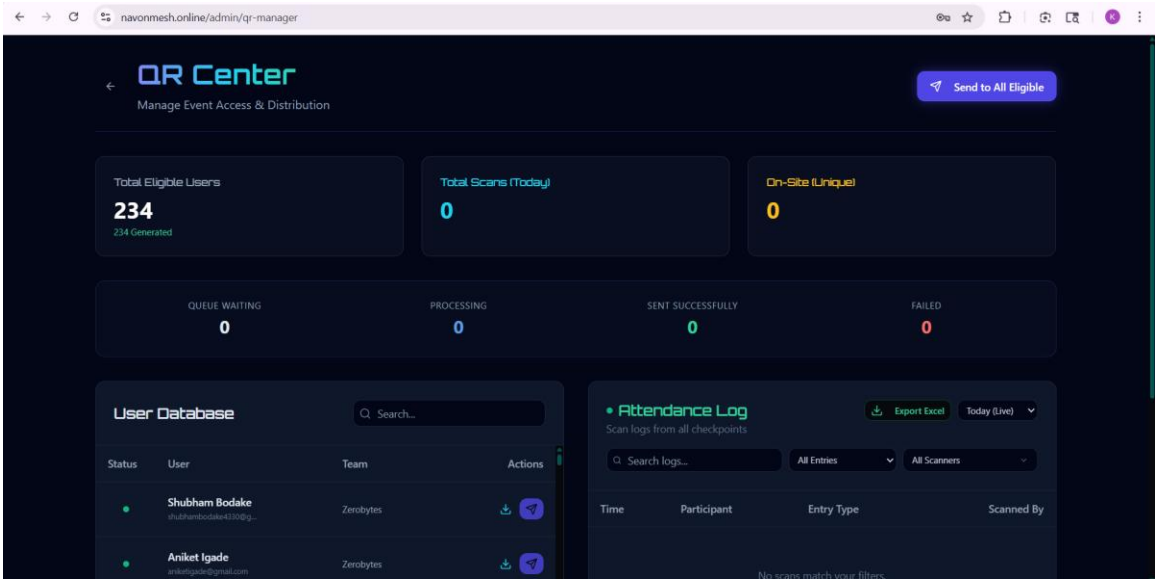
1. Continue scrolling to view the complete Endpoint Total Hits breakdown.

2. Endpoints are color-coded by HTTP method: GET (green), POST (blue/purple), DELETE (red).

3. Key endpoints: /api/v1/auth/login, /api/v1/admin/teams, /api/v1/admin/active-sessions, /metrics.

4. Use this data to identify unusual traffic patterns or endpoints requiring optimization.

7.3.12 QR Center – QR Manager



1. Click 'QR Manager' from the navigation to access navonmesh.online/admin/qr-manager.

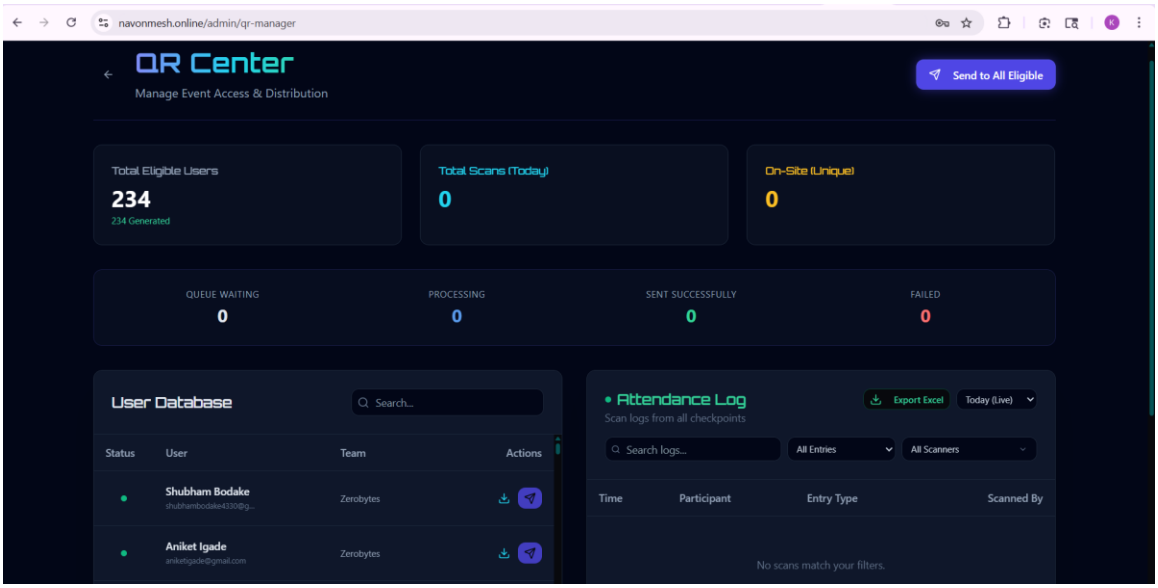
2. View metrics: Total Eligible Users, Total Scans Today, and On-Site Unique count.

3. Check the processing pipeline: Queue Waiting, Processing, Sent Successfully, Failed.

4. In the User Database panel, search participants and use Download/Send buttons.

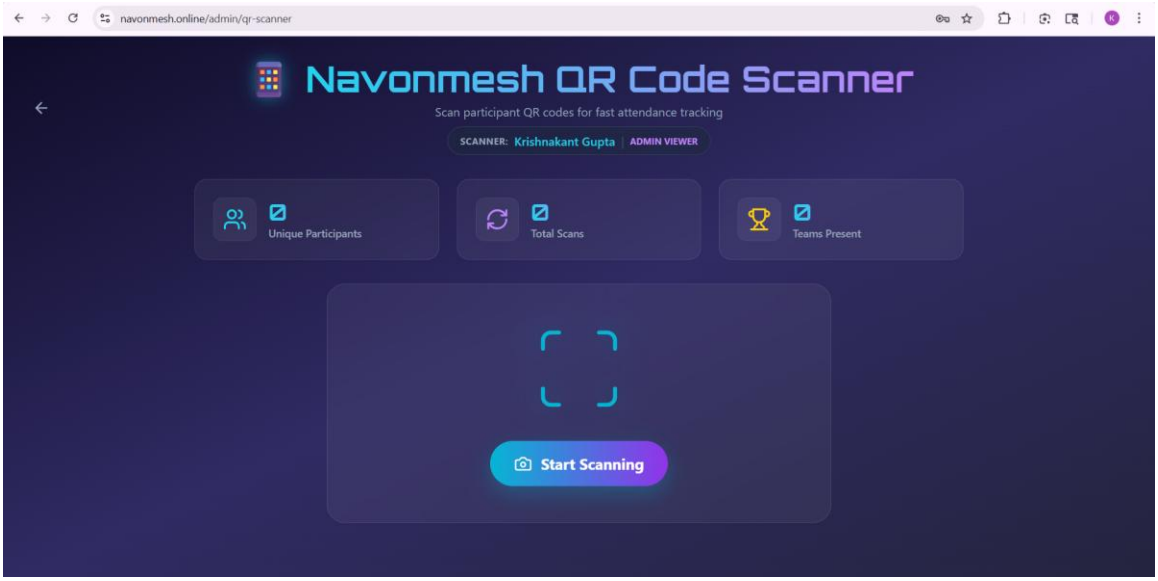
5. In the Attendance Log panel, view scan logs and click 'Export Excel' to download data.

7.3.13 QR Center – Bulk QR Distribution



1. From the QR Center, click 'Send to All Eligible' for bulk QR distribution via email.
2. Monitor the pipeline status to track bulk email delivery progress.
3. Individual QR codes can also be downloaded or emailed using per-participant action buttons.
4. Ensure all team payments are verified before initiating bulk distribution.

7.3.14 QR Code Scanner



1. Click 'QR Scanner' from the navigation to access navonmesh.online/admin/qr-scanner.

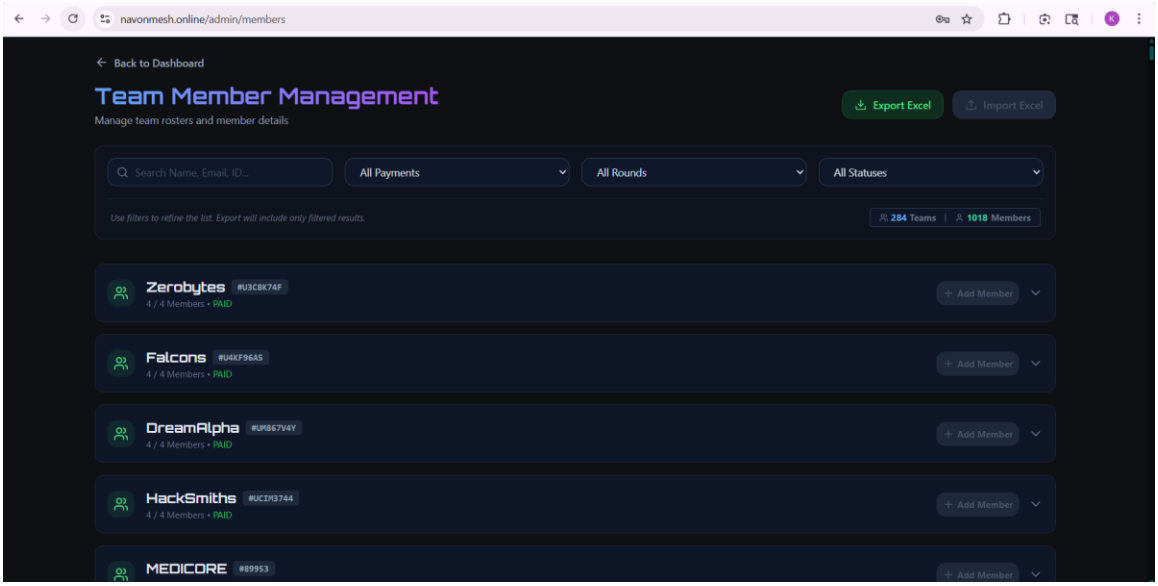
2. Your scanner identity (name and role) is displayed for audit purposes.

3. View stats: Unique Participants, Total Scans, and Teams Present.

4. Click 'Start Scanning' to activate the camera.

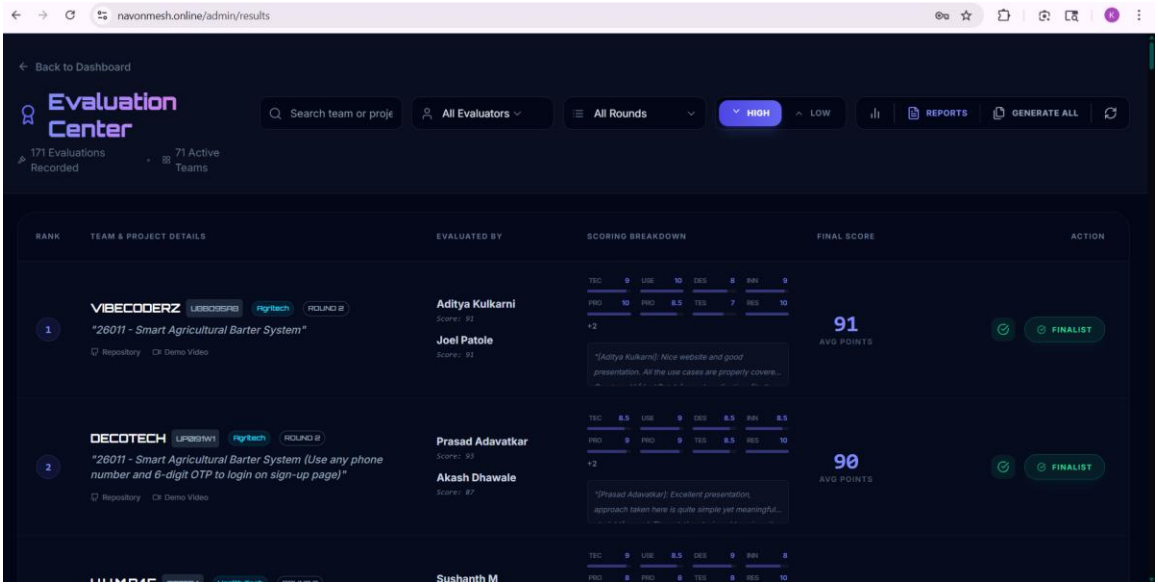
5. Point the camera at a participant's QR code to scan. The system decrypts, verifies, and records attendance automatically.

7.3.15 Team Member Management



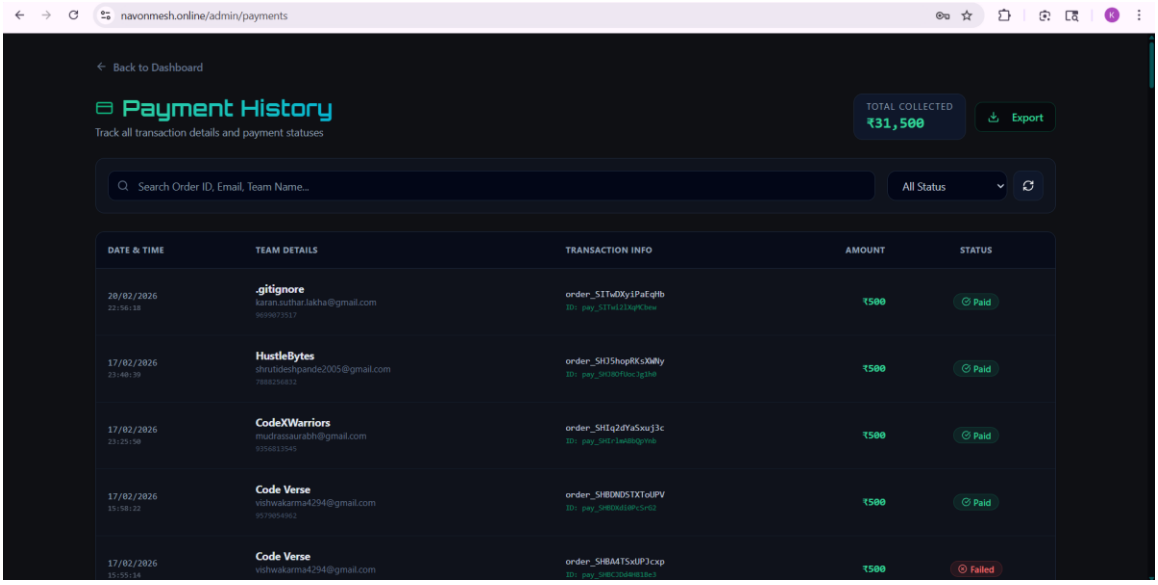
1. Navigate to navonmesh.online/admin/members or click 'Team Members' tab.
2. Search by Name, Email, or Team ID. Filter by Payments, Rounds, and Statuses.
3. View summary badge showing total Teams and Members count.
4. Expand any team to view individual member details.
5. Click '+ Add Member' to add members. Use 'Export Excel' or 'Import Excel' for bulk operations.

7.3.16 Evaluation Center



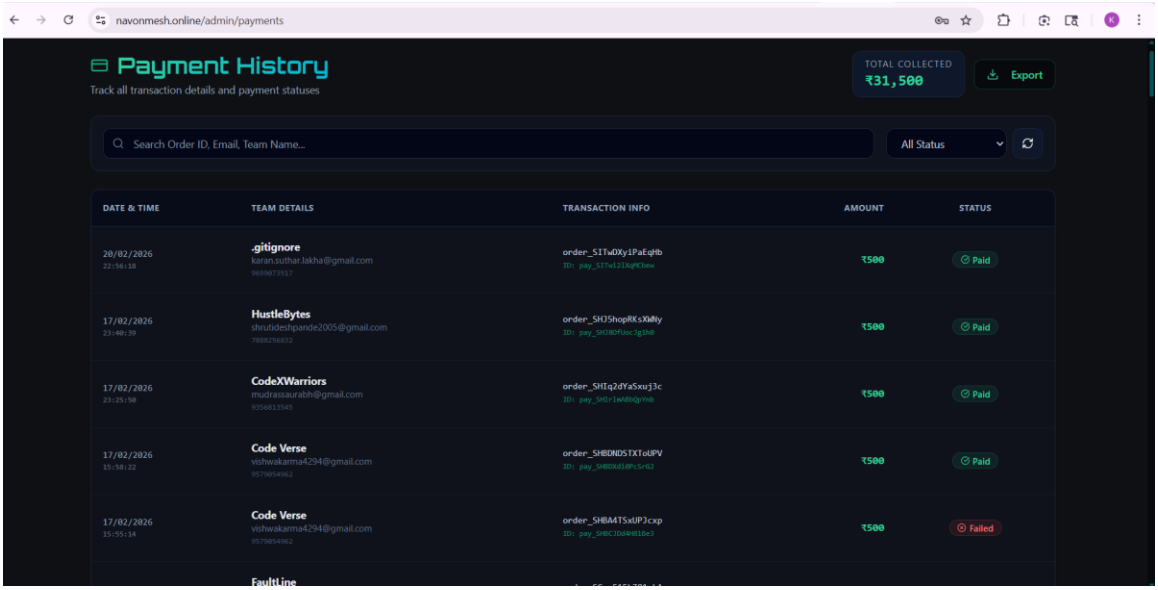
1. Click 'Evaluation Results' tab or navigate to navonmesh.online/admin/results.
2. Search teams and filter by evaluator, round, or score order (HIGH/LOW).
3. Each entry shows: Rank, Team Details, Evaluated By, Scoring Breakdown (TEC, USE, DES, INN, PRO, TES, RES), and Final Score.
4. Click 'REPORTS' to generate reports or 'GENERATE ALL' for batch generation.
5. Use the Action column to manage team Finalist/Winner status.

7.3.17 Payment History



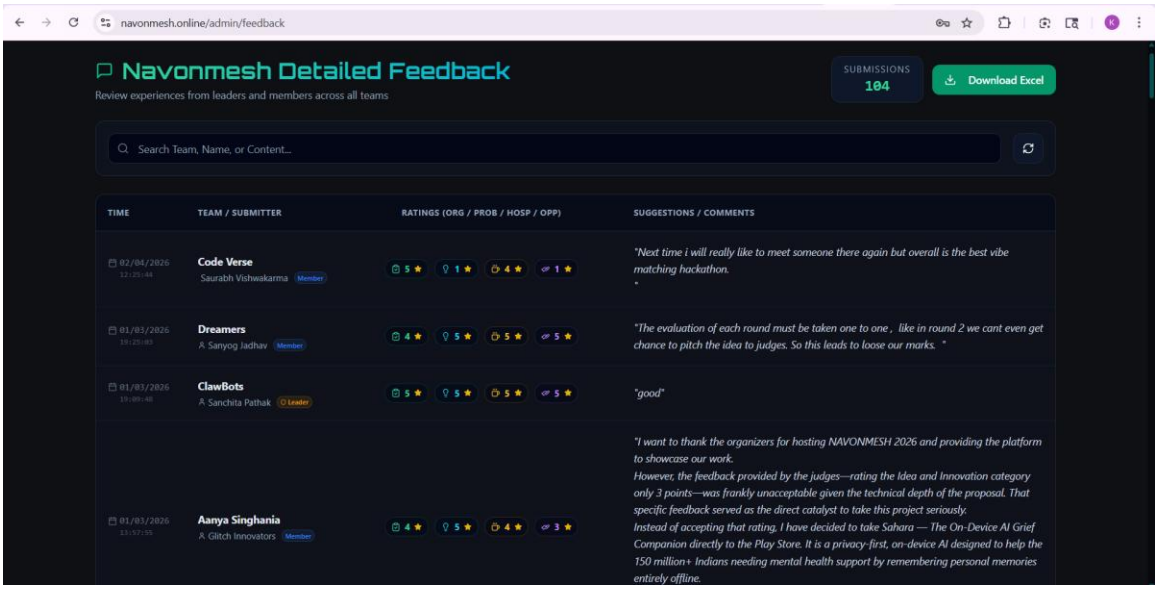
1. Click 'Payment History' tab or navigate to navonmesh.online/admin/payments.
2. View 'TOTAL COLLECTED' badge showing total revenue.
3. Table shows: Date & Time, Team Details, Transaction Info (Razorpay IDs), Amount, and Status (Paid/Failed).
4. Search by Order ID, Email, or Team Name. Filter by status dropdown.
5. Click 'Export' to download payment data as Excel.

7.3.18 Payment History – Transaction Details



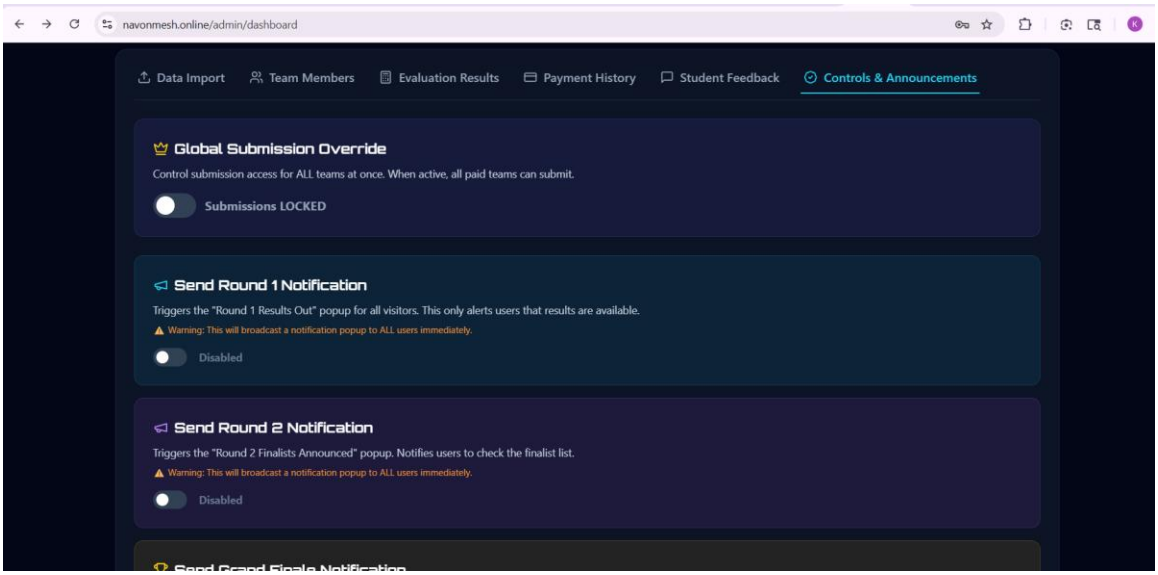
1. Scroll down to view additional transaction records.
2. Each entry shows complete Razorpay trail: order ID and payment ID.
3. Green 'Paid' badges indicate successful transactions; red 'Failed' badges indicate failed attempts.
4. Failed payments require the team leader to retry from their dashboard.

7.3.19 Student Feedback Dashboard



1. Click 'Student Feedback' tab or navigate to navonmesh.online/admin/feedback.
2. View total submissions count. Search by Team Name or Submitter Name.
3. Each entry shows: Time, Team/Submitter (with Leader/Member badge), star ratings for ORG, PROB, HOSP, OPP (1-5 each), and Suggestions/Comments.
4. Click 'Download Excel' to export all feedback data.

7.3.20 Controls & Announcements – Submissions and Notifications



1. Click the 'Controls & Announcements' tab on the Admin Dashboard.

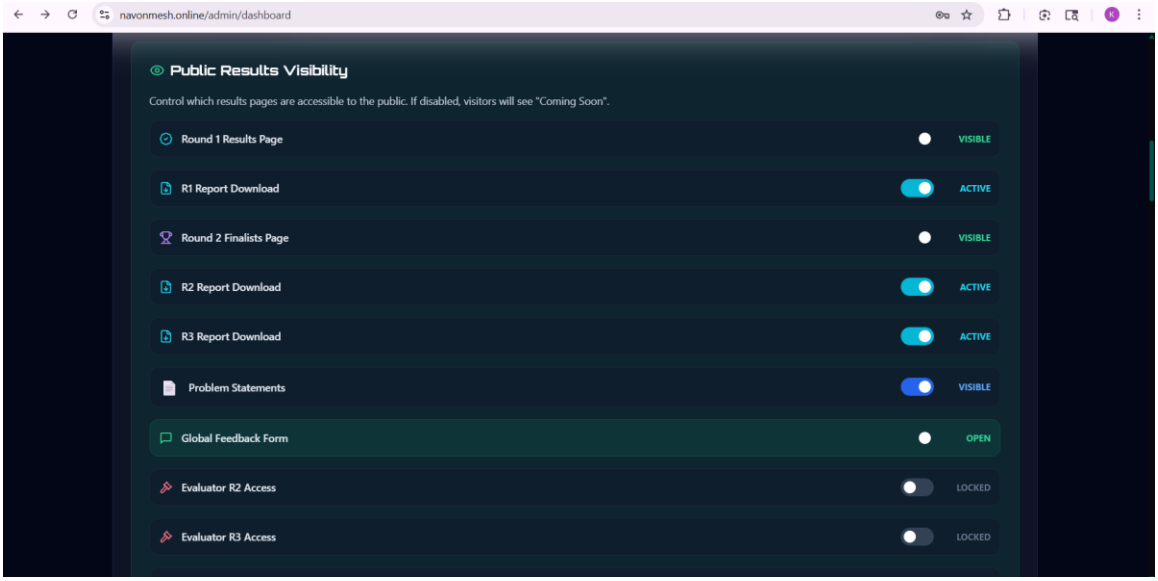
2. Toggle 'Global Submission Override' ON to allow all paid teams to submit; OFF to lock submissions.

3. Toggle 'Send Round 1 Notification' ON to broadcast the 'Round 1 Results Out' popup to all users.

4. Toggle 'Send Round 2 Notification' ON to broadcast the 'Round 2 Finalists Announced' popup.

5. Warning: Each notification broadcasts to ALL users immediately when enabled.

7.3.21 Controls & Announcements – Public Results Visibility



1. Scroll down to the 'Public Results Visibility' section.

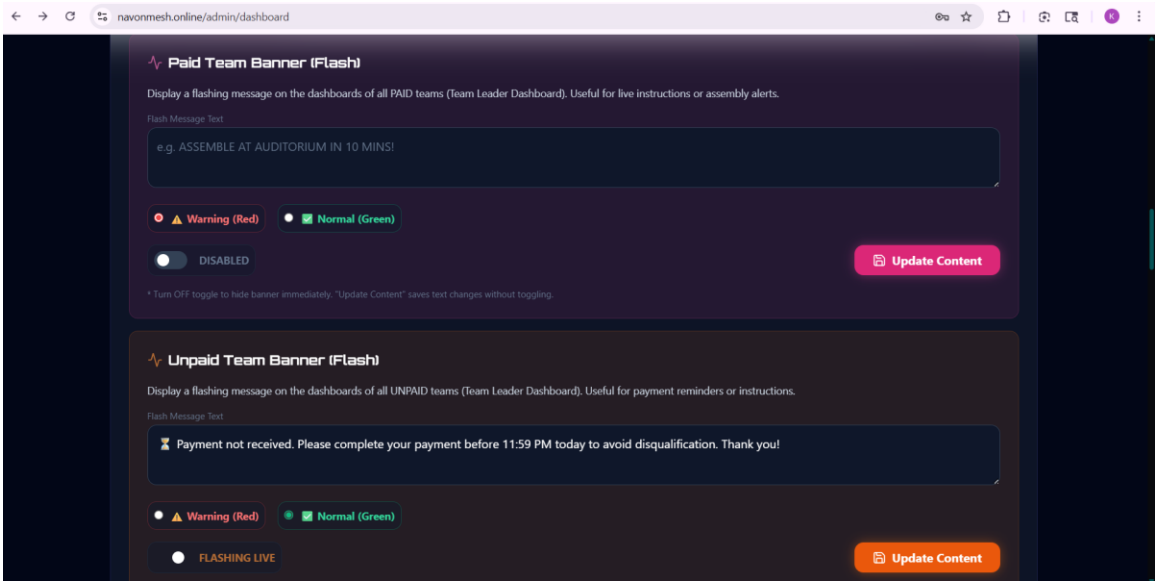
2. Toggle individual pages: Round 1 Results, R1/R2/R3 Report Downloads, Round 2 Finalists, Problem Statements.

3. Toggle 'Global Feedback Form' to open or close public feedback submissions.

4. Toggle 'Evaluator R2 Access' and 'Evaluator R3 Access' to grant/revoke scoring access per round.

5. Disabled pages show 'Coming Soon' to public visitors.

7.3.22 Controls & Announcements – Flash Banners



1. Scroll to the Flash Banner controls section.

2. Use 'Paid Team Banner' to display a flashing message on PAID team dashboards (e.g., assembly alerts).

3. Use 'Unpaid Team Banner' to display payment reminders on UNPAID team dashboards.

4. Select style: 'Warning (Red)' for urgent alerts or 'Normal (Green)' for informational messages.

5. Toggle ON to make the banner live. Click 'Update Content' to save text without toggling.