

Yzone App- Business Requirements Document (BRD)

(NiFo stack: React.js + Node.js (TypeScript) + PostgreSQL on Azure; file storage on Azure Blob; secrets in Azure Key Vault; notifications via WhatsApp)

Executive Summary

Yzone is an **industry-academia internship and problem-solving program** conceptualized and operated by **The Yellow Network (TYN)**. It is designed to bridge the persistent gap between **engineering education** and **industry readiness** by embedding students in **real-world problem contexts**, governed by industry-style execution discipline.

Unlike traditional training platforms or Learning Management Systems (LMS), Yzone functions as a **program execution, governance, and assessment system**. Its primary purpose is to **observe, guide, evaluate, and document how students think, act, communicate, and execute** over a sustained period of time.

The **Yzone software module** is implemented as a **core module within TYN's NiFo platform**, leveraging a shared authentication, authorization, and infrastructure layer. The system supports **multiple colleges, multiple cohorts, and multiple concurrent projects**, while ensuring strict data isolation and role-based access control.

Key Characteristics of Yzone

- **Reflection-first learning model**
Reflection (daily and weekly) is treated as equally important as technical build output. The system is explicitly designed to capture thinking evolution, assumption changes, and learning agility.
- **Multi-project structure**
Each cohort comprises:
 - Individual learning tasks
 - Individual mini projects
 - One major team-based industry project (4–5 students)
- **Controlled expert participation**
Faculty members and industry mentors can be invited to provide feedback and advisory scores **only during facilitator-enabled review windows**, ensuring structure and accountability.
- **Evidence-based assessment**
All evaluations are mapped to a **100-point rubric** that measures behavior, consistency, communication, ownership, and technical competence.

- **Operational realism**
Attendance, deadlines, review windows, reminders, and dashboards mirror real-world internship and early-career expectations.
- **Multi-stakeholder visibility**
Principals, Placement Officers, faculty, facilitators, mentors, and TYN executives each have **appropriate, auditable visibility** into student and cohort progress.

Yzone is deployed in both:

- **CSR model** (e.g., GCT), and
- **Commercial models** (e.g., SRIT),

without compromising governance, rigor, or learning philosophy.

1) Objectives & Success Metrics

1.1 Program-Level Objectives

The Yzone program exists to address systemic shortcomings in undergraduate engineering education by focusing on **how students work**, not just **what they build**.

Primary Objectives

1. **Develop Industry-Ready Engineers**
 - Enable students to function effectively in ambiguous, real-world problem settings.
 - Build habits of consistency, accountability, and reflection.
2. **Shift Learning from Passive to Applied**
 - Replace content-heavy, exam-oriented learning with project-based execution.
 - Encourage decision-making, trade-off analysis, and stakeholder awareness.
3. **Strengthen Communication & Documentation**
 - Improve written articulation, spoken explanation, and technical documentation.
 - Prepare students for interviews, reviews, and professional interactions.
4. **Expose Students to Industry Feedback Loops**
 - Introduce structured reviews, critique, and course correction.
 - Normalize feedback as a continuous improvement mechanism.
5. **Enable Faculty & Institutions to Observe Industry Learning**
 - Provide visibility without requiring faculty to teach new technologies.
 - Support academic oversight through audit and feedback mechanisms.

1.2 Platform-Level Objectives

The Yzone module must operationalize the above program goals through software that enforces discipline, transparency, and scalability.

Platform Objectives

1. **Multi-Tenant Program Governance**
 - o Support 3–5 colleges concurrently.
 - o Each college may run multiple cohorts with different durations (10–20 weeks).
 2. **End-to-End Program Lifecycle Management**
 - o Student admission (selection + wild-card)
 - o Curriculum & project setup
 - o Attendance and reflection tracking
 - o Reviews, assessments, certification
 3. **Role-Based Visibility & Control**
 - o Students see only self and team data.
 - o Faculty and Principals see individual, department, and cohort views.
 - o Facilitators control review participation.
 - o TYN Executives have full oversight.
 4. **Replace Informal Tracking with Structured Evidence**
 - o Eliminate ad-hoc spreadsheets and subjective judgments.
 - o Ensure all progress and evaluations are recorded, timestamped, and auditable.
 5. **Reduce Coordination Overhead**
 - o Centralize dashboards, notifications, and reminders.
 - o Use WhatsApp for operational communication, integrated with system events.
-

1.3 Success Metrics (KPIs)

Student Success Metrics

- **≥ 80% attendance consistency** across active program weeks.
 - **≥ 85% daily reflection compliance** during non-exam weeks.
 - **≥ 70% improvement** between interim and final assessments.
 - 100% completion of:
 - o Individual mini project
 - o Major team project (for active participants)
-

Program Execution Metrics

- Attendance recorded for **100% of sessions**.
 - Daily reflection records present for **≥90% of active days**.
 - Weekly reflections reviewed for **≥75% of students**.
 - All assessments mapped to rubric dimensions.
-

Institutional & TYN Metrics

- Principals and Placement Officers can:
 - View department-level performance at any time.
 - Export cohort summaries without manual intervention.
- Zero cross-college or cross-cohort data leakage.
- Certificates generated for **100% of eligible students** within closure window.

2) Roles, Tenancy & Access Model

2.1 Role Philosophy

The Yzone platform enforces **clear separation of responsibility** while allowing **controlled collaboration**.

Key principles:

- **Authority is centralized** (Facilitator / TYN Exec)
 - **Expert input is advisory** (Faculty / Industry Mentor)
 - **Participation is time-bound**
 - **All actions are auditable**
-

2.2 User Roles (Detailed)

2.2.1 TYN Executive (Super Admin)

Purpose: Platform owner and ultimate authority.

Capabilities:

- Full facilitator privileges across all cohorts.
 - Assign and revoke any role.
 - Override assessments with mandatory justification.
 - Act as:
 - Facilitator for a cohort
 - Industry mentor for specific projects
 - Access cross-cohort analytics and audits.
-

2.2.2 Facilitator (Cohort Owner)

Purpose: Operational owner of a cohort.

Core Responsibilities:

- Admit students (selection + wild-card).
- Form teams (4–5 members).
- Assign tasks and projects.
- Mark attendance.
- Open/close review windows.
- Provide binding rubric scores.
- Generate certificates.

Control Powers:

- Enable faculty or mentors **only during defined review periods**.
 - Define scope of feedback and scoring permissions.
-

2.2.3 Faculty / Principal / Placement Officer

All academic users share the same access rights.

Default Access:

- View-only access to:
 - Individual student records
 - Department aggregates
 - Cohort dashboards

Conditional Access (Facilitator-controlled):

- Provide qualitative feedback.
- Contribute **advisory scores** to selected rubric dimensions.

- Participate only during enabled review sessions.
-

2.2.4 Industry Mentor

Purpose: Domain expert providing practical guidance.

Access Scope:

- Assigned projects, teams, or individuals only.
- No cohort-wide visibility.

Capabilities (when enabled):

- Project-specific feedback.
 - Advisory scoring for relevant rubric dimensions.
-

2.2.5 Student

Purpose: Program participant.

Access:

- Own attendance, reflections, feedback.
- Team project artefacts and discussions.
- Public leaderboard visibility.

Restrictions:

- Cannot see other teams' data.
 - Cannot see raw advisory scores.
-

2.3 Tenancy & Data Isolation

- Each college is a **tenant**.
- Each cohort is isolated within tenant.
- All data entities are tagged with:
 - tenant_id
 - cohort_id

Hard enforcement at:

- API layer
 - Query layer
 - UI rendering layer
-

2.4 Security & Audit

- JWT-based authentication.
 - RBAC enforced server-side.
 - Audit logs for:
 - Role assignment
 - Review window changes
 - Score overrides
 - Certificate generation
-

3) Major Features (Functional Requirements)

This section defines **all functional capabilities** required in the Yzone module, grouped by logical subsystems. Each feature description includes **scope**, **rules**, and **constraints** to eliminate ambiguity during development.

3.1 MoU & Governance

- Create college (tenant) with logo, POC(s).
- Create **MoU**: parties (TYN=1st, College=2nd, optional 3rd), scope notes, validity dates, PDFs in **Azure Blob**.
- Capture related approvals: **Principal/Placement approval**, **hostel warden approval** for extended hours (file uploads).
- MoU status timeline (Draft → Review → Signed → Active → Expired).
- Cohort must link to an **Active** MoU.

3.2 Application & Selection

- **Form builder** per cohort: text/paragraph/MCQ/file-upload questions; deadline; eligibility flags.
- Students apply; data saved to **application_response**.
- **Rubric** with weighted parameters; assign evaluators (TYN/Faculty/Sponsor).
- Evaluators score blind (no prior scores shown until submit).
- Auto-rank (weighted sum); manual override with reason (logged).
- Change state: **Applied** → **Shortlisted** → **Selected** → **Waitlist/Rejected**.

- WhatsApp notification to student on state change.

B. Wild-Card Admission (Facilitator-only)

Purpose:

Allow late entrants, special cases, or institutional nominees.

Rules:

- Only **Facilitator** or **TYN Exec** can add a student via wild-card.
- Wild-card entries:
 - Skip application & evaluation stages.
 - Must have reason captured (mandatory text field).
- System logs:
 - Who added the student
 - Date/time
 - Justification

Post-Admission:

- Wild-card students are treated identically post-enrollment.
- Included in attendance, tracker, assessments, certificates.

3.3 Cohort, Team & Project Structure

Cohort - A cohort represents:

- One college
- One program run
- One time-bounded execution window (10–20 weeks configurable)

Each cohort contains:

- Curriculum
- Students
- Teams
- Tasks
- Projects
- Assessments

Team Scope:

- Shared:

- Major project
 - Team artefacts
 - Team-level feedback
- Individual:
 - Attendance
 - Daily tracker
 - Individual rubric scores

Project Types

Each cohort must have:

1. Individual Learning Tasks

- Short-duration activities.
- Non-graded or lightly graded.
- Examples:
 - Reading + reflection
 - Tool exploration
 - Prompting exercises

2. Individual Mini Project

- Owned fully by one student.
- Duration: 1–2 weeks.
- Evaluated on:
 - Execution
 - Reflection
 - Documentation

3. Major Group Project

- Team-based (4–5 members).
- Duration: 4–6 weeks.
- Has:
 - Defined milestones (M1, M2, M3)
 - Architecture review
 - Final demo & documentation
 -

3.2.4 Visibility Rules

Entity	Visible To
Individual task	Student, Facilitator, Faculty (audit)
Mini project	Student, Facilitator, Assigned Mentor
Group project	Team members, Facilitator, Assigned Mentor

Entity	Visible To
Project feedback	Only respective student/team
Scores	Student sees aggregate, not individual evaluator splits

3.3 Curriculum & Sessions

- **Program → Cohort → Curriculum** structure.
- Curriculum supports **100 days** or any custom length.
- Each **Session**: title, date/time (local IST), mode (Lab/Remote), assets (URLs/files), expected output, optional team linkage.
- Different Yzones may use **different curricula**; cloning supported.
- Teams (3–5 students) inside a cohort: assign sessions/tasks per-team when needed. Facilitator assigns teams manually.

3.4 Attendance (Facilitator-only)

- Only **Facilitator** records attendance (no student self-mark).
- Record for session (date/time) at **individual** or **team** level (stores per-student).
- Reason codes for absence (Class/Exam/Health/Travel/Other); evidence upload optional.
- Students see attendance **read-only**.
- Export attendance CSV; daily/weekly compliance charts.

3.5 Daily Progress Tracker

- Student submits for scheduled days:
Date | What I learnt today? | What issue I faced today? | What I plan to do tomorrow? | (Facilitator remarks)
- Facilitator bulk review with quick-reply chips + free-text remarks. Facilitator remarks can be added anytime.
- SLA: if not submitted by **22:00 IST**, send WhatsApp reminder; escalate after 24h.

3.6 Badges & Leaderboard

- Badge rules (configurable in v1 as pre-sets):
 - **Streak-5/10** (consecutive tracker submissions)
 - **Attendance-Star** ($\geq 90\%$ sessions)
 - **Helper** (facilitator tags student for peer support ≥ 3)
 - **On-Time** (0 late trackers in last 14 days)
- Points per badge; **leaderboard** per cohort; student can opt-out from **public rank** (still collects badges).

3.6.1 Weekly Reflection

Once per week:

- Student submits:
 - Key learning
 - What changed in thinking
 - What they would do differently

3.6.2 Review Windows

- Facilitator enables review window for a specific date/time.
- During window:
 - Faculty & Industry Mentors can:
 - Add qualitative feedback
 - Add advisory scores to selected rubric dimensions
- Outside window:
 - Read-only

All review inputs are timestamped.

3.6.3 Badges

Examples:

- Attendance Star
- Reflection Streak
- Ownership Champion
- Documentation Pro

Badges auto-awarded or manually awarded by Facilitator.

3.6.4 Leaderboard

- Visible to **everyone**.
- Ranked by cumulative badge points.
- No cohort-to-cohort comparison.

3.7 Notifications (WhatsApp)

- All notifications over WhatsApp (individual + group channels).

- Events: session reminders, missing tracker nudges, selection status change, badge awarded, schedule changes, certificate ready.
- Template library with variables; consent status per user.
- Retry & fallback to email if WhatsApp fails (optional flag; default off).

3.8 Certificates

- Auto-generate **Internship Completion Certificate** PDF (tenant-branded “TYN standard”; no per-college theming).
- Includes digital signature image(s), student details, hours, dates, project title, performance summary line.
- Stored in Azure Blob; downloadable by student & visible to college admin.

3.9 Import / Export

- Import existing **attendance** and **tracker** spreadsheets (CSV/XLSX) → guided column mapping.
- Exports for selection, attendance, tracker, badges, certificates list.

3.10 Reports & Dashboards

- **Cohort Overview:** attendance %, tracker completion %, late submissions, badges earned, team-level view.
- **Student Reporter:** timeline of attendance & trackers; remarks log; badges.
- **Selection Funnel:** applicants → shortlisted → selected → joined.
- **Sponsor View:** progress & completion rate; anonymized if required.
- **MoU Compliance:** milestones checklist (kick-off, mid-review, final demo, certificate generation).

4) System Design (Architecture & Data)

4.1 Hosting & Infra

- **Azure App Service / AKS** (as per NiFo baseline) for Node/React.
- **PostgreSQL** (Azure Database for PostgreSQL).
- **Azure Blob Storage** for files (MoU PDFs, approvals, session assets, certificate PDFs).
- **Azure Key Vault** for secrets (DB URI, WhatsApp bot token, signing keys).
- **Time zone:** Asia/Kolkata.
- WhatsApp Business API
-

4.2 Key Entities (ERD Outline)

(Table → important fields)

- tenant → id, name, logo_url, primary_contact
- party → id, type(TYN/COLLEGE/THIRD), tenant_id, contact
- mou → id, tenant_id, party_ids[], start_date, end_date, status, file_url
- approval_document → id, tenant_id, student_id?, type(principal_approval, hostel_warden), file_url, issued_on
- program → id, tenant_id, name, desc
- cohort → id, program_id, name, start_dt, end_dt, curriculum_id, status
- team → id, cohort_id, name
- user → id, phone, email, name, active
- user_tenant_role → id, user_id, tenant_id, role
(EXEC/FACILITATOR/COLLEGE/FACULTY/SPONSOR/STUDENT)
- enrollment → id, cohort_id, user_id, team_id?, status
- application_form → id, cohort_id, schema_json
- application_response → id, cohort_id, user_id, answers_json, submitted_at, status
- rubric → id, cohort_id, weights_json
- evaluation → id, application_response_id, evaluator_id, scores_json, total, notes
- curriculum → id, cohort_id?, name, structure_json
- session → id, cohort_id, title, start_ts, end_ts, mode, assets_json, expected_output, team_id?
- attendance → id, session_id, student_id, status(P/A), reason?, evidence_url?, marked_by, marked_at
- progress_log → id, cohort_id, student_id, date, learned_text, issue_text, plan_text, facilitator_remark, remark_by, remark_at
- badge_rule → id, cohort_id?, key, name, desc, points, rule_json(*predefined in v1*)
- badge_award → id, cohort_id, student_id, badge_key, awarded_at
- leaderboard_entry → id, cohort_id, student_id, points, updated_at, opt_out_rank
- notification_template → id, key, channel(WhatsApp/email), content, variables
- notification_event → id, template_key, recipient_user_id, payload_json, status, sent_at, response_meta
- certificate → id, cohort_id, student_id, status(generated/ready), file_url, issued_at
- audit_log → id, actor_id, tenant_id, action, entity, entity_id, payload, ts

Note: All access filters by tenant_id on server side; cohort scoping enforced in queries.

4.3 Services (Backend Modules)

- **Auth & RBAC** (NiFo shared)
- **Tenancy/MoU**
- **Applications & Evaluations**
- **Cohort/Curriculum/Session**
- **Attendance**
- **Tracker**
- **Badges/Leaderboard**
- **Notifications (WhatsApp)**

- Certificates
- Reports & Exports
- Audit

4.4 Data Isolation Strategy

- Tenant ID enforced at:
 - API layer
 - ORM query builder
- No cross-tenant joins allowed.

4.5 Notifications (WhatsApp)

- Configure a WhatsApp Bot + chat IDs (student direct, cohort group).
 - Persist mapping: `user_id ↔ WhatsApp_chat_id`. Consent flag.
 - Queue + retry (exponential backoff). Optional email fallback.
-

5) User Workflows (Step-by-step)

5.1 Create College & MoU (Exec)

1. Exec → **Create Tenant** (logo, POCs).
2. Upload **MoU** PDF + start/end dates; set status → “Active”.
3. Upload **approval documents** (principal, hostel warden).
4. Create **Program** and **Cohort**; assign TYN **Facilitator and ****Faculty-in-charge**.

5.2 Application & Selection

1. Facilitator → **Create Application Form** (questions, deadline).
2. Students apply via invite link (login OTP) → submit.
3. Facilitator chooses manual **evaluators** who score via **rubric** Or chooses Agent (LLM) based evaluation. LLM uses the set prompt instructions and delivers results; system totals & ranks.
4. Facilitator **shortlists/selects**; posts reasons for overrides. Notifies Staff-in-charge via email and WhatsApp app.
5. WhatsApp notifies students of outcome; **enrollment** records created for selected.

5.3 Curriculum & Sessions

1. Facilitator imports/clones or creates **curriculum**.
2. Create weekly **sessions** with assets and outcomes; optionally bind to **team**.
3. Students view sessions; **WhatsApp reminder** 30–60 min before start.

5.4 Facilitator Flow (Daily and weekly)

1. During/after session, Facilitator opens **Attendance UI**.
2. Mark **Present/Absent**, add reason for absentees; upload evidence if provided. It should be a calendar view with student names as rows & dates as column header. Each cell marked with the attendance legend. A bottom row showing day-wise total. Logically colour-coded cells, based on the entry value.
3. Save → updates dashboards; students see read-only status.
4. Collect faculty/mentor input, Close review window, Update assessments

5.4.1 Faculty / Mentor Review Flow

1. Access enabled review
2. View assigned students/projects
3. Submit qualitative feedback
4. Submit advisory score
5. Exit review window

5.5 Daily Tracker

1. Students open **Tracker** for the day, submit the 3 fields (learned/issue/plan).
2. At 20:30 IST, if missing → WhatsApp **reminder**.
3. Facilitator reviews in **Bulk Review**: quick replies or custom remark.
4. Late (>22:00) flagged; second reminder next day if still missing.

5.6 Badges & Leaderboard

1. Nightly job computes badge eligibility; awards and updates **leaderboard**.
2. WhatsApp notifies “Badge unlocked □”.

5.7 Certificates

1. On cohort completion (attendance \geq threshold and tracker \geq threshold), Facilitator clicks **Generate Certificates**.
2. PDFs created with **digital signatures**, stored in Blob; students get WhatsApp link to download.
3. College Admin can bulk-download zip.

5.8 Imports/Exports

- Facilitator uploads CSV/XLSX for past **attendance** or **trackers** → field mapping wizard → preview → commit.
 - Reports export to CSV; certificates export list to CSV; attendance/trackers per-week.
-

6) UI Overview (Wireframe-level, concise)

- **Staff Landing (NiFo → Yzone)**: Tenants/Cohorts list → pick cohort.
- **Cohort Dashboard**: cards for Attendance %, Tracker %, Late count, Highlights of progress till date (generated or Facilitator entered), Next 3 sessions, Leaderboard top 5.
- **Applications**: list + filters; open response; evaluator pane with system generated score and provision for manual score - rubric sliders; rank table.
- **Curriculum**: week calendar; add/edit session modal; attach files/links.
- **Attendance**: table (students × session); mark present/absent; reasons.
- **Tracker Review**: inbox with student cards; quick-reply chips; remark box.
- **Badges**: rule list (read-only in v1); awards list; leaderboard toggle opt-out.
- **Reports**: filters + export buttons.
- **Certificates**: generate → list with status & download link.

Student-side:

- **Today**: sessions (upcoming), tracker (submit), progress badges, leaderboard peek.
- **My Records**: attendance read-only timeline, tracker history, certificate tab.

This section defines **screen responsibilities, data visibility rules, interaction states, and validations**.

It intentionally avoids visual styling and focuses on **what each screen must do**.

6.1 UI Design Principles

1. **Role-driven rendering**
 - UI components appear strictly based on role + enabled privileges.
2. **Cohort-scoped navigation**
 - A user always operates inside a selected cohort context.
3. **Reflection-first UX**
 - Tracker, reflections, and feedback are surfaced *before* project artefacts.
4. **Low cognitive load**
 - Students see only “Today”, “This Week”, and “My Work”.
5. **Audit-friendly**
 - All facilitator and mentor actions are visible via timestamps & history.

6.2 Entry & Navigation

6.2.1 Login Routing

User Role	Landing Page
Student	My Yzone Dashboard
Facilitator	Cohort Control Dashboard

User Role	Landing Page
Faculty / Mentor Review Dashboard (read-only unless enabled)	
TYN Exec	Multi-Cohort Executive Dashboard

6.2.2 Cohort Selector (Exec / Facilitator)

- Dropdown of accessible cohorts.
- Context switch reloads all widgets.
- No cross-cohort data mixing.

6.3 Student UI

6.3.1 Student Dashboard

Primary Sections (Top to Bottom):

1. **Today's Focus**
 - Session (if any)
 - Task due
 - Tracker status (Submitted / Pending)
2. **Daily Tracker Panel**
 - Inline editor for 3 tracker fields
 - Save state indicator
 - Lock indicator after 23:59 IST
3. **Feedback Highlights**
 - Latest facilitator/mentor remarks
 - Click to expand full history
4. **My Projects**
 - Mini Project (individual)
 - Major Project (team)
 - Milestone status
5. **Badges & Leaderboard**
 - Earned badges
 - Rank position

6.3.2 Student Tracker History

- Calendar or list view.
- Each entry shows:
 - Submission time
 - Facilitator remark (if any)
 - Late flag

No edit option beyond same day.

6.4 Facilitator UI

6.4.1 Facilitator Dashboard (Cohort Level)

Widgets:

- Attendance % (weekly & cumulative)
- Tracker submission heatmap
- Students needing attention (auto-flagged)
- Upcoming reviews & milestones

6.4.2 Attendance Screen

Matrix View:

- Rows → Students
- Columns → Dates/Sessions
- Cells → Present / Absent / Reason

Rules:

- Editable only by Facilitator.
- Save is atomic per day.

6.4.3 Tracker Review Console

- Filter by:
 - Date
 - Student
 - Team
 - Late / Missing

Actions:

- Add remark
- Mark “Needs Follow-up”
- Tag for mentor review

6.4.4 Review Window Control

Facilitator can:

- Enable review window:
 - Date & time range
 - Select participants (faculty / mentors)

- Select rubric dimensions allowed
- Disable review window:
 - Locks further inputs
 - Preserves audit trail

6.5 Faculty / Industry Mentor UI

6.5.1 Review Dashboard

- Visible only during enabled windows.
- Shows:
 - Assigned students/projects
 - Tracker history (read-only)
 - Project artefacts

6.5.2 Feedback Entry

Mentor can submit:

- Qualitative feedback (mandatory)
- Advisory score (optional)

Restrictions:

- Cannot see other evaluators' inputs.
- Cannot edit after submission.

6.6 Executive (TYN) UI

6.6.1 Executive Dashboard

Cross-Cohort View:

- Active cohorts
- Attendance & engagement trend
- Completion forecast
- Risk flags (dropout, inactivity)

6.6.2 Drill-Down Views

Exec can drill into:

- Cohort → Department → Student
- Project → Team → Individual

Exec actions:

- Assign / reassign facilitators
- Act as mentor
- Override scores (with reason)

6.7 UI Validations & Guardrails

- Tracker submission requires all 3 fields.
- Attendance cannot be edited after 48 hrs.
- Review feedback allowed only in enabled window.
- Wild-card student addition prompts mandatory justification.

7) Reports (v1)

1. **Cohort Summary:** Attendance trend, Tracker heatmap, On-time ratio, Badge distribution.
2. **Student Scorecard:** Attendance %, Tracker %, Streak, Badges, Facilitator remarks list.
3. **Selection Funnel:** Applied/Shortlisted/Selected/Joined counts, avg rubric scores.
4. **Sponsor Dashboard:** % progress, % completion, average attendance, anonymized highlights.
5. **MoU Checklist:** planned vs completed milestones with dates.

Reports are **not decorative**.

They exist to drive **decisions, governance, and accountability**.

7.1 Report Categories

- A. Operational Reports - Used by Facilitator.
- B. Governance Reports - Used by Principal / Placement / Faculty.
- C. Executive Reports - Used by TYN leadership & sponsors.

7.2 Core Reports

7.2.1 Cohort Performance Report

Dimensions:

- Attendance %
- Tracker compliance %
- Reflection quality index
- Project milestone completion

Views:

- Overall cohort
- Department-wise
- Team-wise

Export: CSV

7.2.2 Individual Student Scorecard

Includes:

- Attendance
- Tracker quality
- Mini project score
- Group project score
- Reflection score
- Final weighted score (100)

Visible to:

- Student (own)
- Facilitator
- Faculty
- Exec

7.2.3 Reflection Quality Report

Aggregates:

- Weekly reflection submissions
- Facilitator feedback density
- Improvement trend

Purpose:

- Ensure “thinking” is not skipped.

7.2.4 Project Progress Report

For each project:

- Team members
- Milestones (M1–M3)
- Current status

- Risks / blockers

Used in:

- Review meetings
- Sponsor updates

7.2.5 Attendance & Engagement Report

- Day-wise presence
- Consecutive absence flags
- Holiday vs working day trends

7.2.6 Faculty / Mentor Contribution Report

Shows:

- Feedback given
- Reviews participated
- Advisory scores submitted

Used for:

- Institutional transparency
- Recognition

7.3 Report Access Matrix

Report	Student	Facilitator	Faculty	Exec
Student Scorecard	✓(self)	✓	✓	✓
Cohort Performance	✗	✓	✓	✓
Reflection Report	✗	✓	✓	✓
Project Progress	✓(own)	✓	✓	✓
Attendance	✓(view)	✓	✓	✓

7.4 Export & Audit

- All reports exportable as CSV.
- Every export logged:
 - User
 - Timestamp
 - Purpose (optional text)

8) Badges, Streaks & Leaderboard

This section defines **how Yzone nudges behaviour** without micromanagement. Badges and streaks reward **consistency, reflection, and ownership**, not just output.

- **Streak-5:** 5 consecutive days on-time trackers. (10 pts)
- **Streak-10:** 10 consecutive days on-time tracker. (25 pts)
- **Attendance-Star:** ≥90% sessions present. (30 pts)
- **Helper:** Facilitator tags “helped peer” 3 times. (15 pts)
- **Quality:** Facilitator scores on a scale of 1-10 for each day. Cumulative average taken on any given day (20 pts)

Leaderboard = sum(points). Weekly reset option **off** in v1.

8.1 Design Philosophy

1. **Visible to everyone**
 - Badges, streaks, and leaderboard are public within the cohort.
2. **Effort > Brilliance**
 - Consistency and reflection are weighted more than one-time excellence.
3. **No gaming**
 - Rules are deterministic, auditable, and time-bound.
4. **Intrinsic + Extrinsic**
 - Badges create recognition; leaderboard creates momentum.

8.2 Badge Types

8.2.1 System-Awarded Badges (Automatic)

Badge	Trigger Condition	Points
Attendance Star	≥90% attendance	30
Reflection Streak – 5	5 consecutive daily trackers	10
Reflection Streak – 10	10 consecutive daily trackers	25
On-Time Champion	Zero late trackers in 14 days	15
Consistency Pro	No missing tracker in 21 days	20

8.2.2 Facilitator-Awarded Badges (Manual)

Badge	Purpose	Points
Ownership Champion	Proactive ownership beyond task	20

Badge	Purpose	Points
Documentation Pro	Clear, reproducible documentation	15
Team Enabler	Helped peers consistently	15
Critical Thinker	Asked sharp questions / trade-offs	10

Rules:

- Manual badge requires mandatory justification.
- Badge award is logged and visible.

8.3 Streak Logic

8.3.1 Tracker Streak

- Counts **consecutive calendar days** with tracker submission.
- Resets if:
 - Tracker missing
 - Submitted after deadline

8.3.2 Attendance Streak

- Counts **consecutive sessions attended**.
- Absence breaks streak unless marked “Approved Absence”.

8.4 Leaderboard Computation

- Leaderboard score = sum of badge points.
- Recomputed nightly.
- Tie-breakers:
 1. Attendance %
 2. Reflection streak length
 3. Earlier completion timestamp

8.5 Leaderboard Visibility & Ethics

- Visible to:
 - Students
 - Facilitator
 - Faculty
 - Exec
- No opt-out in v1 (by design).
- Leaderboard is **cohort-scoped** only.

8.6 Auditability

Every badge award stores:

- Badge key
 - Awarded by
 - Awarded to
 - Reason (mandatory for manual)
 - Timestamp
-

9) Notifications (Templates & Triggers)

Communication is **operational infrastructure**, not an afterthought.
WhatsApp is used because it matches **student behavior patterns**.

9.1 Notification Architecture

Channel	Purpose
Cohort WhatsApp Group	Announcements, schedules
Individual WhatsApp	Reminders, feedback, nudges

9.2 WhatsApp Integration Model

- Uses **WhatsApp Business API**.
- Message templates pre-approved.
- Messages are:
 - Plain text only
 - No file attachments (v1 constraint)

9.3 Notification Types & Triggers

9.3.1 System-Triggered Notifications

Trigger	Recipient	Message Type
Tracker missing	Student	Reminder
Review window open	Student / Faculty	Action prompt
Feedback added	Student	Informational
Badge awarded	Student	Recognition
Certificate ready	Student	Completion

9.3.2 Facilitator-Triggered Notifications

- Session announcements
- Schedule changes
- Special instructions
- Motivation notes

Sent to:

- Cohort group
- Selected individuals

9.4 Reminder Rules

- **One reminder only** for missing tracker.
- Sent at configurable time (default 20:30 IST).
- No escalation in v1.

9.5 Message Template Governance

Each template includes:

- Template key
- Variables
- Allowed roles to trigger

9.6 Logging & Reliability

Every notification logs:

- Recipient
- Template
- Payload
- Delivery status
- Timestamp

Failures:

- Logged
- Visible in facilitator dashboard

9.7 Compliance & Consent

- Student consent captured at first login.

- Opt-out **not allowed** (program participation implies consent).
- **Session Reminder** (group): “ Session at {{date}} {{day}} {{time}}: {{title}}. See assets: {{link}}” (T-60 / T-30 min)
- **Missing Tracker** (user): “ Tracker pending for {{date}}. Please submit before 21:00.”
- **Late Tracker Escalation** (user + optional facilitator): “ Tracker still pending for {{date}}.”
- **Selection Outcome** (user): “ You are {{status}} for Yzone—{{cohort}}.”
- **Badge Awarded** (user): “ Badge unlocked: {{badge_name}}.”
- **Certificate Ready** (user): “ Your certificate is ready: {{link}}.”
- **Schedule Change** (group): “ Update: {{session}} moved to {{new_time}}.”

Consent capture at first login (checkbox + timestamp).

10) Acceptance Criteria (MVP)

- **Tenancy & MoU**: create college, upload MoU PDF, set active, link to cohort.
- **Application**: form live; ≥1 student submits; ≥1 evaluator scores; shortlist generated; WhatsApp sent.
- **Curriculum**: create ≥10 sessions; clone once; students can view.
- **Attendance**: facilitator marks present/absent; student sees read-only; export CSV works.
- **Tracker**: student submits; facilitator remarks; missing-tracker reminder fires.
- **Badges**: Streak-5 and Attendance-Star award automatically; leaderboard updates; opt-out hides rank.
- **Notifications**: 3 templates live (reminder, selection outcome, badge); delivery <5 min median.
- **Certificates**: generate PDFs with digital signature; student can download; blob link valid.
- **Reports**: cohort summary + student scorecard export to CSV.
- **Security**: RBAC enforced; audit logs recorded for critical actions.
- **Data Import**: attendance CSV import succeeds with mapping wizard.
- **Reflection & Review Windows**: Weekly reflection submission possible
 - Facilitator can:
 - Enable review window
 - Select faculty/mentor
 - Select rubric dimensions
 - Faculty/Mentor:
 - Can submit feedback only during window
 - After window closes:
 - No edits allowed

- Data is read-only
- **Assessment & Rubric**
 - Rubric weights sum to 100
 - Multiple evaluators allowed
 - Advisory scores:
 - Stored separately
 - Not auto-aggregated
 - Final score:
 - Set by Facilitator / Exec only
 - Override requires justification

- **Badges & Leaderboard**
 - Badge rules trigger correctly
 - Manual badge requires justification
 - Leaderboard updates nightly
 - Ranking visible to all cohort members
- **Notifications (WhatsApp)**
 - Templates render correctly
 - Messages delivered to:
 - Group
 - Individuals
 - Missing tracker reminder fires once per day
 - Delivery logs visible

11) Non-Functional Requirements

- **Performance:** 25 concurrent users per cohort; up to 5 cohorts active; p95 API < 800 ms.
- **Reliability:** 99.5% uptime; notification queue retry (3 attempts, backoff).
- **Security:** JWT auth; server-side RBAC; tenant_id scoping; PII encrypted at rest where applicable.
- **Compliance:** IST time zone; retention 3 years.
- **Branding:** Standard TYN (no white-label).
- **A11y:** Keyboard navigable; minimum contrast; responsive UI.

12) Delivery Plan

13) Risks & Mitigation

Risk	Mitigation
WhatsApp delivery failures	Retry & log; optional email fallback; daily delivery report
Facilitator overload	Bulk review UI, quick-replies, batch attendance
Data leakage across tenants	Strict tenant filters; integration tests; code review gates
Low student compliance	Gamification + automated nudges; weekly summary to faculty
Import inconsistencies	Mapping wizard + preview; reject with error details

14) Open Questions (please confirm)

1. **WhatsApp Bot & Groups:** Will we use **individual DMs + one cohort group** per Yzone? Yes. create & manage official cohort groups.
2. **Eligibility Thresholds for Certificates:** Exact rules? e.g., Attendance $\geq 75\%$ **and** Tracker days $\geq 70\%$ **and** at least 1 project deliverable uploaded. Ok. Facilitator can over-ride.
3. **Evaluator Identities:** Should student see evaluator names in selection feedback? (Default: **hidden**.) No.
4. **Project Artefacts:** Do we need a per-team **deliverables repository** (file uploads, version notes) in v1? No.

5. **Faculty Remarks Visibility:** Should faculty remarks be visible to students by default? **yes**, with “private remark” option for faculty.
 6. **Data Migration:** Do we need to migrate **historic** Yzone spreadsheets (prior years) or only from current cohorts? Only current cohort.
 7. **Certificate Co-signers:** Single signer (TYN) now; should we support optional **college co-signer** image in v1 or Phase-2? Single signer now.
-

API Contract + Database Schema (DDL)

1. **API Design Principles & Conventions**
2. **API Contracts (Endpoints) – module-wise**
3. **Database Schema (DDL – PostgreSQL)**

If you want, next we can convert this into:

- OpenAPI (Swagger) YAML
 - Prisma / TypeORM schema
 - Jira epics & stories
-

1. API DESIGN PRINCIPLES

Base

```
Base URL:/api/v1
Auth:JWT(NiFoAuth)
Timezone:ISTenforcedatbackend
```

Headers

```
Authorization: Bearer <token>
X-Tenant-Id: <college_id>
X-Cohort-Id: <cohort_id>
```

Role Enforcement

- All authorization enforced **server-side**
 - Role checks happen per endpoint
 - Tenant & cohort isolation is mandatory
-

2. API CONTRACTS (MODULE-WISE)

2.1 Authentication & User Context

Get Logged-in User Context

```
GET /auth/me
```

Response

```
{  
  "userId": "uuid",  
  "name": "string",  
  "roles": ["FACILITATOR"],  
  "tenantIds": ["uuid"],  
  "cohortIds": ["uuid"]  
}
```

2.2 Tenant & MoU Management (TYN Exec)

Create Tenant (College)

```
POST /tenants  
{  
  "name": "GCT",  
  "contactPerson": "Principal Name"  
}
```

Upload MoU

```
POST /tenants/{tenantId}/mou  
{  
  "startDate": "2025-01-01",  
  "endDate": "2026-03-31",  
  "thirdParty": "Alumni Association"  
}
```

(File uploaded via multipart → stored in GCS)

2.3 Cohort Management

Create Cohort

```
POST /cohorts  
{  
  "tenantId": "uuid",  
  "name": "Yzone GCT Batch-2",  
  "startDate": "2025-11-01",  
  "endDate": "2026-02-15"  
}
```

Assign Facilitator / Faculty / Mentor

```
POST /cohorts/{cohortId}/roles  
{
```

```
"userId":"uuid",
"role":"FACULTY"
}
```

2.4 Student Admission

Wildcard Student Entry (Facilitator only)

```
POST /cohorts/{cohortId}/students/wildcard
{
  "name": "Student Name",
  "email": "email@gct.edu",
  "phone": "9XXXXXXXXX",
  "department": "CIVIL",
  "justification": "Late nominee by department"
}
```

2.5 Team & Project Management

Create Team

```
POST /cohorts/{cohortId}/teams
{
  "name": "Team Alpha"
}
```

Assign Student to Team

```
POST /teams/{teamId}/members
{
  "studentId": "uuid"
}
```

Create Project

```
POST /projects
{
  "cohortId": "uuid",
  "type": "MAJOR",
  "title": "Invoice Processing AI",
  "teamId": "uuid",
  "milestones": ["M1", "M2", "M3"]
}
```

2.6 Attendance (Facilitator Only)

Mark Attendance

```
POST /sessions/{sessionId}/attendance
{
  "studentId": "uuid",
  "status": "PRESENT",
  "reason": null
}
```

Get Attendance Matrix

```
GET /cohorts/{cohortId}/attendance
```

2.7 Daily Progress Tracker

Submit Daily Tracker

```
POST /tracker
{
  "date": "2026-01-15",
  "learned": "Studied invoice OCR pipeline",
  "issue": "Low OCR accuracy",
  "plan": "Try Tesseract vs PaddleOCR"
}
```

Facilitator Remark

```
POST /tracker/{trackerId}/remark
{
  "remark": "Good direction. Compare precision metrics."
}
```

2.8 Weekly Reflection & Review Window

Enable Review Window

```
POST /reviews/windows
{
  "cohortId": "uuid",
  "start": "2026-01-18T10:00",
  "end": "2026-01-18T18:00",
  "allowedRoles": ["FACULTY", "MENTOR"],
  "rubricDimensions": [3, 6, 7]
}
```

Submit Advisory Feedback

```
POST /reviews/{windowId}/feedback
{
  "studentId": "uuid",
  "comments": "Good analytical thinking",
  "scores": {
    "criticalThinking": 4
  }
}
```

2.9 Assessment & Rubric

Submit Assessment

```
POST /assessments
{
  "studentId": "uuid",
  "type": "INTERIM",
  "scores": {
    "attendance": 12,
    "projectOutput": 15
  },
  "comments": "Consistent progress"
}
```

2.10 Badges & Leaderboard

Award Manual Badge

```
POST /badges/award
{
  "studentId": "uuid",
  "badgeKey": "OWNERSHIP_CHAMPION",
  "reason": "Handled deployment independently"
}
```

Get Leaderboard

```
GET /cohorts/{cohortId}/leaderboard
```

2.11 Notifications (WhatsApp)

Trigger Notification

```
POST /notifications/send
{
  "type": "TRACKER_Reminder",
```

```
"userId":"uuid",
"payload":{
"date":"2026-01-15"
}
}
```

2.12 Certificates

Generate Certificates

```
POST /cohorts/{cohortId}/certificates/generate
```

Download Certificate

```
GET /certificates/{certificateId}/download
```

3. DATABASE SCHEMA (DDL – POSTGRESQL)

Below is **core DDL** (simplified but complete).

3.1 Users & Roles

```
CREATE TABLE users (
    id UUID PRIMARY KEY,
    name TEXT NOT NULL,
    email TEXT UNIQUE,
    phone TEXT,
    created_at TIMESTAMPDEFAULT NOW()
);

CREATE TABLE user_roles (
    user_id UUID REFERENCES users(id),
    tenant_id UUID,
    cohort_id UUID,
    role TEXT CHECK (role IN
    ('EXEC','FACILITATOR','FACULTY','MENTOR','STUDENT')),
    PRIMARY KEY (user_id, cohort_id, role)
);
```

3.2 Tenants & Cohorts

```
CREATE TABLE tenants (
    id UUID PRIMARY KEY,
    name TEXT,
```

```
    created_at TIMESTAMP  
);  
  
CREATE TABLE cohorts (  
    id UUID PRIMARY KEY,  
    tenant_id UUID REFERENCES tenants(id),  
    name TEXT,  
    start_date DATE,  
    end_date DATE  
);
```

3.3 Teams & Projects

```
CREATE TABLE teams (  
    id UUID PRIMARY KEY,  
    cohort_id UUID,  
    name TEXT  
);  
  
CREATE TABLE team_members (  
    team_id UUID,  
    student_id UUID,  
PRIMARY KEY (team_id, student_id)  
);  
  
CREATE TABLE projects (  
    id UUID PRIMARY KEY,  
    cohort_id UUID,  
    team_id UUID,  
    type TEXT CHECK (type IN ('MINI', 'MAJOR')),  
    title TEXT,  
    status TEXT  
);
```

3.4 Attendance

```
CREATE TABLE attendance (  
    id UUID PRIMARY KEY,  
    session_id UUID,  
    student_id UUID,  
    status TEXT CHECK (status IN ('PRESENT', 'ABSENT')),  
    reason TEXT,  
    marked_by UUID,  
    marked_at TIMESTAMP  
);
```

3.5 Daily Tracker

```
CREATE TABLE daily_tracker (  
    id UUID PRIMARY KEY,  
    student_id UUID,
```

```

cohort_id UUID,
dateDATE,
learned TEXT,
issue TEXT,
plan TEXT,
submitted_at TIMESTAMP,
UNIQUE(student_id, date)
);

CREATE TABLE tracker_remarks (
    tracker_id UUID,
    remark TEXT,
    remarked_by UUID,
    remarked_at TIMESTAMP
);

```

3.6 Reviews & Assessments

```

CREATE TABLE review_windows (
    id UUID PRIMARY KEY,
    cohort_id UUID,
    start_time TIMESTAMP,
    end_time TIMESTAMP
);

CREATE TABLE assessments (
    id UUID PRIMARY KEY,
    student_id UUID,
    type TEXT,
    scores JSONB,
    comments TEXT,
    evaluator_id UUID
);

```

3.7 Badges & Certificates

```

CREATE TABLE badge_awards (
    id UUID PRIMARY KEY,
    student_id UUID,
    badge_key TEXT,
    reason TEXT,
    awarded_at TIMESTAMP
);

CREATE TABLE certificates (
    id UUID PRIMARY KEY,
    student_id UUID,
    cohort_id UUID,
    file_url TEXT,
    issued_at TIMESTAMP
);

```


Yzone – Phase-wise Build Plan

Sustainable velocity: **18–22 story points / week, 2 week sprints**

TEAM STRUCTURE & OWNERSHIP

Role	Responsibility
Architect / Tech Lead (PT)	System design, PR reviews, schema/API sign-off
Dev-1 (Backend)	Core APIs, DB schema, RBAC
Dev-2 (Backend)	Tracker, attendance, assessments
Dev-3 (Frontend)	Student + Facilitator UI
Dev-4 (Frontend)	Reports, dashboards, admin UI

All devs use ChatGPT for:

- Boilerplate generation
 - Test case scaffolding
 - SQL/query validation
 - UI component drafts
-

PHASE 0 — FOUNDATION & ALIGNMENT

Objective

Avoid architectural debt before line-1 code.

Activities

- Finalize BRD/FSD (done)
- Lock:
 - API conventions
 - DB naming conventions
 - Role → permission matrix
- Setup:
 - GCP project
 - CI/CD pipeline
 - Base repo (frontend + backend)
- WhatsApp Business API sandbox setup

Deliverables

- Repo initialized
- Empty app deployable
- Auth + JWT working
- Developer playbook (how to run locally)

PHASE 1 — CORE PLATFORM SKELETON

Objective

Make the system **usable end-to-end for a single cohort**.

Features Built

Backend

- Tenant, cohort, user, role models
- Wild-card student entry
- Cohort assignment
- Audit logging (critical)

Frontend

- Login + cohort selection
- Student dashboard (basic)
- Facilitator dashboard (basic)

Infrastructure

- Tenant isolation enforced
- RBAC middleware complete

Deliverables

- A facilitator can:
 - Create cohort
 - Add students
 - Assign roles
- A student can:
 - Login
 - See dashboard

PHASE 2 — DAILY TRACKER & ATTENDANCE

Objective

Enable discipline, visibility, and reflection.

Features Built

Attendance

- Session creation
- Attendance matrix (facilitator only)
- Student read-only view

Daily Tracker

- Submit daily tracker
- Lock after 23:59 IST
- Facilitator remarks
- Missing-tracker reminder (WhatsApp)

Frontend

- Tracker UX (student)
- Bulk review UX (facilitator)

Deliverables

- Daily usage possible
- WhatsApp reminders firing
- Tracker data usable for assessment

PHASE 3 — PROJECTS, TEAMS & REVIEWS

Objective

Enable real project execution with audits.

Features Built

Projects & Teams

- Team creation (4–5)
- Mini project + major project
- Milestones

Review Windows

- Facilitator enables review
- Faculty / mentor feedback
- Advisory scores

Reflection

- Weekly reflection submission
- Review aggregation

Deliverables

- Project reviews operational
 - Faculty participation possible
 - Reflection data captured
-

PHASE 4 — ASSESSMENTS, BADGES & REPORTS (Week 12–14)

Objective

Convert activity into **measurable outcomes**.

Features Built

Assessment Engine

- Interim + final assessments
- Rubric-based scoring
- Advisory vs binding scores

Badges & Leaderboard

- Auto + manual badges
- Leaderboard computation

Reports

- Student scorecard
- Cohort performance
- Department-wise views

Deliverables

- Students see progress clearly
- Facilitators can evaluate objectively
- Exec dashboards live

PHASE 5 — CERTIFICATES, HARDENING & GO-LIVE

Objective

Make it **production-ready** and auditable.

Features Built

- Certificate generation (PDF + signature)
- Export reports (CSV)
- Performance tuning
- Security audit
- Bug fixes
- UX polish

Deliverables

- Ready for live Yzone cohorts
- Certificate-backed internship
- Clean audit trail

KEY EXECUTION DISCIPLINE (Non-negotiable)

1. Weekly demo to stakeholders
2. No feature without acceptance criteria
3. Architect signs off on DB & APIs
4. ChatGPT assists, humans decide
5. Do not optimize prematurely