What We're Building!

Gimsoi Project Tracker – Full Context for Teams

Project Purpose

We are not building a generic SaaS or selling software upfront.

We are building:

- An internal delivery system that doubles as a client transparency layer.

Key points:

- Clients don't install or configure anything.
- Value is experienced first — they see progress and receive reports.
- Our system becomes sticky because clients trust it and adopt it naturally.
- Internally, it enforces discipline, accountability, and structured delivery.

The Real Problem we're Solving:

- Clients resist new tools unless they see immediate, obvious value.
- Most tools require onboarding, licenses, or internal change.
- Gimsoi solves this by giving clients visibility without asking them to change anything.

The Value Flywheel:

- Deliver projects efficiently.
- Clients see live progress and download reports.
- Trust increases.
- Adoption and stakeholder confidence grow.

- ➢ Opportunity for future monetisation emerges.

Team Expectations & Roles

1. Product & Programme Management

Role: Define workflows, scope, and success criteria. Ensure the project tracker reflects real delivery.

Responsibilities:

- ➢ Define MVP scope and sprint goals.
- ➢ Write and approve user stories (Admin, PM, Intern, Client).
- ➢ Validate project workflows and dashboards.
- ➢ Set success metrics: task completion, sprint velocity, contribution tracking.
- ➢ Run sprint demos and track accountability.

Outcome:

- A well-structured, internal delivery system that aligns with client-facing transparency.

---

2. Backend Engineering

Role: Build the “brains” of the tracker — APIs, database, authentication, and task logic.

Responsibilities:

- ➢ Build authentication and role-based access control (RBAC).
- ➢ Implement APIs for clients, projects, sprints, and tasks.
- ➢ Ensure data isolation (internal vs client views).
- ➢ Log activities for audit and reporting.
- ➢ Ensure secure and scalable deployment.

Outcome:

- ➢ Interns and teams can reliably input and track work.
- ➢ Clients can view read-only dashboards without affecting internal operations.

---

3. Frontend Engineering

Role: Build the “face” of the tracker — dashboards, Kanban boards, and client interfaces.

Responsibilities:

- ➢ Build login/logout and role-based dashboards.
- ➢ Display internal progress for teams.
- ➢ Display client-facing read-only dashboards and reports.
- ➢ Implement Kanban boards for task management.
- ➢ Handle responsive UI, error handling, and usability.

Outcome:

- Teams see all project data internally.
- Clients access real-time progress without any effort.

---

4. UX & Data / Analytics

Role: Make the tracker intuitive and extract actionable insights.

Responsibilities:

- ➢ Create wireframes and UI designs for all dashboards.
- ➢ Define and track MVP metrics (velocity, contribution, task completion).
- ➢ Build reports for sprints, project progress, and team performance.

Outcome:

- ➢ Interns can track performance visually.
- ➢ Clients get professional, easy-to-read reports.

---

5. QA

Role: Ensure the tracker works reliably for both internal and client-facing views.

Responsibilities:

- Test all flows: authentication, CRUD operations, tasks, dashboards.
- Validate internal vs client read-only access.
- Conduct regression and usability testing.

Outcome:

- A polished, reliable system that functions as intended and builds client trust.

---

6. DevOps & Security

Role: Make the system deployable, secure, and monitored.

Responsibilities:

- Set up Git repos, CI/CD, and dev/staging environments.
- Secure environment variables and API keys.
- Monitor performance and conduct access audits.
- Deploy production-ready system.

Outcome:

- The tracker is stable, secure, and production-ready for internal use and client visibility.

Key Takeaways for Teams

- Internally Essential: Enables structured project delivery and accountability.
- Externally Valuable: Gives clients real-time visibility and reports without them changing processes.
- Commercially Optional: The tool can later evolve into a paid product, but the focus now is on trust and delivery.
- Execution-Focused: Every sprint, demo contributions, task tracking, and KPIs reinforce accountability.

Team & Sprint Focused

Gimsoi Project Tracker – Execution Guide (Team & Sprint Focused)

Overview:

- 12-week MVP roadmap, 6 sprints (2 weeks each).
- Goal: Build an internal delivery system that also provides clients read-only dashboards and reports.
- Focus: Intern execution, accountability, and leadership oversight.
- Each task includes purpose, dependencies, guidance, expected outcome, and client-value alignment.

SPRINT 0 (Weeks 1–2): FOUNDATION & SETUP

Objective: Prepare the environment, define scope, and set structures for internal delivery.

Team Responsibilities

Product & Programme Management

- Define MVP scope, success criteria, and Definition of Done.
- Write core user stories (Admin, PM, Intern, Client) with acceptance criteria.
- Create backlog and prioritisation.
- Client value: Lays the foundation so internal work can be tracked reliably and visible to clients later.

Backend Engineering

- Define architecture, service boundaries, and database schema.
- Set up a base project structure.
- Client value: Ensures client data is isolated and accessible in read-only dashboards.

Frontend Engineering

- ➢ Set up frontend framework, routing, and basic layout.
- ➢ Client value: Prepares client-facing dashboards for later sprints.

UX & Data

- ➢ Wireframes for internal and client dashboards.
- ➢ Define MVP metrics: velocity, task completion.
- ➢ Client value: Ensures client dashboards are clear and metrics are meaningful.

QA

- ➢ Define test strategy and bug-tracking process.
- ➢ Client value: Prevents errors in dashboards or reports clients will see.

DevOps & Security

- ➢ Set up Git repos, CI/CD, dev/staging environments.
- ➢ Client value: Ensures client dashboards and internal systems are stable and secure.

SPRINT 1 (Weeks 3–4): AUTHENTICATION & USER MANAGEMENT

Objective: Enable secure login and role-based access control for internal and client users.

Team Responsibilities
Product

- ➢ Finalise auth-related user stories.
- ➢ Approve role-based access flows.
- ➢ Client value: Clients will only see what's relevant; security is ensured.

Backend

- ➢ Implement Google OAuth.
- ➢ Implement RBAC and audit logging.

- Create user/role APIs.
- Client value: Clients can log in safely with read-only access.

Frontend

- Build login/logout flows.
- Render dashboards by role.
- Handle access errors.
- Client value: Clients get seamless access without setup.

QA

- Test auth and role-access flows.
- Client value: Reduces friction and ensures correct data visibility.

DevOps & Security

- Secure environment variables.
- Review access permissions.
- Client value: Clients' data is protected.

SPRINT 2 (Weeks 5–6): CLIENTS, PROJECTS & SPRINTS

Objective: Implement management of clients, projects, and internal sprints.

Team Responsibilities
Product

- Validate workflows for projects/sprints.
- Client value: Structured progress becomes visible to clients later.

Backend

- APIs for clients, projects, sprints.
- Enforce lifecycle rules.
- Client value: Clients can see accurate project progress.

Frontend

- UI for client/project management, sprint lists.
- Client value: Visual dashboards for clients to track work.

Database

- Migrate schemas, enforce relationships.
- Client value: Data integrity ensures client trust.

QA

- CRUD testing and regression.
- Client value: Guarantees internal and client dashboards are reliable.

SPRINT 3 (Weeks 7–8): TASKS & KANBAN BOARD

Objective: Implement internal task management and Kanban board for workflow tracking.

Team Responsibilities
Product

- Accept task lifecycle and prioritisation rules.
- Client value: Clients see realistic progress; tasks are tracked transparently.

Backend

- Task CRUD APIs, assignment logic, comments, effort tracking.
- Client value: Clients' read-only dashboards reflect real work.

Frontend

- Kanban board, drag-and-drop tasks, task details.
- Client value: Visual clarity for clients on progress.

QA

- Task flow and usability testing.
- Client value: Accurate display and smooth functionality.

Data

- ➢ Capture task-level metrics.
- ➢ Client value: Metrics form the basis of client reports.

SPRINT 4 (Weeks 9–10): CLIENT DASHBOARD, REPORTING & SECURITY

Objective: Enable clients to view dashboards, download reports, and maintain system security.

Team Responsibilities
Product

- ➢ Approve client read-only experience.
- ➢ Validate reporting usefulness.
- ➢ Client value: Clients receive actionable updates without intervention.

Backend

- ➢ Client APIs, data isolation, reporting endpoints.
- ➢ Client value: Secure client access to live progress and reports.

Frontend

- ➢ Client dashboard UI, progress visualisation, report views.
- ➢ Client value: Clients consume value without effort.

Data Analytics

- ➢ Sprint velocity and contribution metrics.
- ➢ Client value: Stakeholders can see performance in reports.

Security & DevOps

- ➢ Security hardening, access audits, performance testing.
- ➢ Client value: Clients trust the system is secure.

QA

- ➢ Security and performance testing.
- ➢ Client value: Ensures dashboards are accurate and functional.

SPRINT 5 (Weeks 11–12): STABILISATION, DOCUMENTATION & DEMO

Objective: Finalise MVP, polish UI, fix bugs, document usage, and prepare for internal and client demos.

Team Responsibilities
Product & Programme Management

- ➢ Final MVP acceptance and demo script.
- ➢ Define Phase 2 roadmap.
- ➢ Client value: Client transparency and reporting are production-ready.

Backend

- ➢ Bug fixes, refactoring, performance optimisation.
- ➢ Client value: Stable, secure, and efficient client-facing system.

Frontend

- ➢ UI polish and responsiveness.
- ➢ Client value: Clean, professional dashboards for clients.

QA

- ➢ Full regression testing and release sign-off.
- ➢ Client value: Ensures system reliability for clients.

DevOps

- ➢ Production deployment and monitoring.
- ➢ Client value: Client dashboards are accessible and monitored.

Documentation & Support

- User guides, intern contributions summary.
- Client value: Smooth onboarding and clear understanding of tracker outputs.

Tracking & Accountability (All Sprints)

- Each member assigned sprint tasks.
- At least one demo contribution per sprint.
- PRs/deliverables linked to tasks in the project tracker.
- Teams meet regularly to coordinate dependencies.
- Client-value alignment explained at each demo.

This guide ensures every team understands exactly what to build, why it matters for clients, and how their deliverables create trust and transparency.

Gimsoi MVP Chart

To understand your role, responsibilities, and why it matters.

Gimsoi MVP – Team & Sprint Client-Value Chart

Sprint	Team	Key Tasks	Client Value / Impact
0: Foundation & Setup	Product	Define MVP scope, user stories, backlog	Lays groundwork for client transparency later
	Backend	Setup architecture, DB schema, project structure	Ensures client data is safe and ready for dashboards
	Frontend	Setup framework, routing, basic layout	Prepares client dashboards for future use
	UX/Data	Wireframes, MVP metrics	Dashboards and metrics will be clear to clients
	QA	Define test strategy, bug tracking	Prevents errors in dashboards clients will see
	DevOps/Security	Git repos, CI/CD, dev/staging environments	Stable, secure internal system supports client views

1: Auth & User Management	Product	Finalise auth stories, role flows	Clients see only relevant data; secure access
	Backend	Google OAuth, RBAC, audit logs	Secure login; client visibility controlled
	Frontend	Login/logout flows, role-based dashboards	Clients get smooth, read-only access
	QA	Auth & access testing	Prevents access issues, ensures reliability
	DevOps/Security	Secure env vars, access review	Protects client data and system integrity
2: Clients, Projects & Sprints	Product	Validate workflows	Structured progress visible to clients
	Backend	Client/project/sprint APIs	Accurate project progress feeds client dashboards
	Frontend	UI for client/project management	Visual clarity for clients
	Database	Schema migration, relationships	Data integrity builds client trust
	QA	CRUD & regression testing	Dashboards and reports function reliably
3: Tasks & Kanban Board	Product	Accept task lifecycle rules	Clients see real-time, accurate progress
	Backend	Task APIs, assignment, comments	Tasks are tracked transparently for clients
	Frontend	Kanban board, task details	Visual clarity and interaction for clients

	QA	Task flow, usability testing	Accurate task display for client dashboards
	Data	Capture task-level metrics	Metrics form basis of client reports
4: Client Dashboard, Reporting & Security	Product	Approve read-only dashboards, reporting	Clients can track projects effortlessly
	Backend	Client APIs, data isolation, reporting endpoints	Secure, live dashboards and reports for clients
	Frontend	Client dashboard UI, reports	Clients consume value without effort
	Data Analytics	Sprint velocity & contribution metrics	Stakeholders see clear performance data
	Security/Dev Ops	Security hardening, audits, performance	Builds client trust in system integrity
	QA	Security & performance testing	Dashboards are accurate and reliable
5: Stabilisation, Documentation & Demo	Product	Final MVP acceptance, demo script	Clients see polished, professional delivery
	Backend	Bug fixes, optimisation	Stable, secure system for client use
	Frontend	UI polish, responsiveness	Professional dashboards for clients
	QA	Full regression, release sign-off	System reliability ensures client confidence
	DevOps	Production deployment & monitoring	Dashboards accessible, monitored

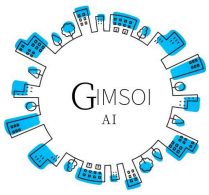
	Documentation	User guides, intern contribution summary	Clients and teams understand system and reporting

---

Use This Chart to:

- Employee: Check your team column for sprint tasks and understand client impact.
- Team Leads: Please use this to assign tasks and explain why they matter.
- All Teams: Align your work with client value — every task either delivers internal efficiency or creates client transparency.

Team Tools - MVP



Gimsoi MVP – Team + Sprint + Tools Map

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
0: Foundation & Setup	Product	Define MVP scope, user stories, backlog	Google Docs, Notion, Trello	Docs & backlog tracking; collaboration tools
	Backend	Setup architecture, DB schema, project structure	Firebase Firestore + Cloud Functions, Supabase	Free NoSQL DB + serverless backend
	Frontend	Setup framework, routing, basic layout	React.js + Tailwind CSS, Vercel Hosting	Open-source frontend framework; free hosting
	UX/Data	Wireframes, MVP metrics	Figma Free, Google Sheets, Google Data Studio	Wireframes for dashboards; metric tracking
	QA	Define test strategy, bug tracking	GitHub Issues, Trello	Track bugs & testing tasks
	DevOps/Security	Git repos, CI/CD, dev/staging environments	GitHub, GitHub Actions, Vercel	Free repos, CI/CD, hosting

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
1: Auth & User Management	Product	Finalise auth stories, role flows	Google Docs, Trello	Story tracking
	Backend	Google OAuth, RBAC, audit logs	Firebase Authentication, Supabase Auth	Free auth & role management
	Frontend	Login/logout flows, role-based dashboards	React.js, Tailwind CSS	Integrate with Firebase Auth
	QA	Auth & access testing	GitHub Issues, Trello	Track tests
	DevOps/Security	Secure env vars, access review	GitHub Actions, Vercel	CI/CD & environment secrets

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
2: Clients, Projects & Sprints	Product	Validate workflows	Google Docs, Notion	Track project workflow acceptance
	Backend	APIs for clients/projects/sprints	Firebase Cloud Functions, Supabase	Free serverless API endpoints
	Frontend	UI for client/project management	React.js + Tailwind CSS	Connect to backend APIs
	Database	Schema migration, relationships	Firestore, Supabase Postgres	Free tier databases
	QA	CRUD & regression testing	GitHub Issues, Trello	Testing internal CRUD flows

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
3: Tasks & Kanban Board	Product	Accept task lifecycle rules	Google Docs, Trello	Workflow tracking
	Backend	Task APIs, assignment, comments	Firebase Cloud Functions, Supabase	Free backend endpoints
	Frontend	Kanban board, task details	React.js + Tailwind CSS	Drag-and-drop UI
	QA	Task flow & usability testing	Trello, GitHub Issues	Track usability bugs
	Data	Capture task-level metrics	Google Sheets, Google Data Studio	Free analytics & dashboards

---

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
4: Client Dashboard, Reporting & Security	Product	Approve read-only dashboards	Google Docs, Figma	Wireframe & approval
	Backend	Client APIs, reporting endpoints	Firebase Cloud Functions, Supabase	Provide read-only data safely
	Frontend	Client dashboard UI, reports	React.js + Tailwind CSS	Visualise client dashboards
	Data Analytics	Sprint velocity & contribution metrics	Google Data Studio, Google Sheets	Client-ready reports
	Security/DevOps	Security hardening, audits, performance	GitHub Actions, Vercel	Ensure secure, monitored deployment

	QA	Security & performance testing	GitHub Issues, Trello	Test dashboards & reports

Sprint	Team	Key Tasks	Suggested Free Tools / Credits	Notes
5: Stabilisation, Documentation & Demo	Product	Final MVP acceptance, demo script	Google Docs, Notion	Track deliverables & demo notes
	Backend	Bug fixes, optimisation	Firebase / Supabase	Backend stability
	Frontend	UI polish, responsiveness	React.js + Tailwind CSS	Professional look & feel
	QA	Full regression, release sign-off	GitHub Issues, Trello	Ensure readiness
	DevOps	Production deployment & monitoring	Vercel Hosting, GitHub Actions	Free hosting & CI/CD monitoring
	Documentation	User guides, intern contribution summary	Notion, Google Docs	Internal + client documentation