**# Project Overview and Onboarding — 360-feedback branch**

Last updated: 2025-09-01

This document helps new collaborators quickly understand, run, and contribute to the project. It reflects the current 360-feedback branch.

**## 1) Functional overview**

- Purpose: 360° feedback platform with a single public survey URL per survey, gated by email + auth code.

- Roles:

  - Admin/Creator (internal): Create surveys, assign participants (self/manager/peer/reportee), start/end surveys, view responses.

  - Participant (public): Authenticate with email + auth code, select assigned role, submit responses.

- Question types: Mixed Multiple Choice (MCQ) and Text questions across multiple categories.

- Lifecycle: Draft → Active (single URL generated) → Ended/Completed (assignments lock; no further edits).

**## 2) User journeys**

- Admin/Creator

  - Create a survey in Draft.

  - Assign participants: select employees per role in one grid (self, manager, peer, reportee).

  - Start survey: generates a single public URL token.

  - End survey when done; responses remain viewable.

  - View Survey: shows participants list and summary counts (total participants, total submissions, per-role totals).

  - View Responses: grouped by employee + feedback role; filter by role; shows categories and answers.

- Participant (public)

  - Open survey URL.

  - Enter company email + personal auth code.

  - Pick an allowed role (based on assignment).

  - Answer role-specific questions (MCQ/Text) and submit. Duplicates are prevented per participant-per-role.

**## 3) Key rules and validations**

- Flat participant model by role: self, manager, peer, reportee.

- If an employee is assigned to any non-self role, they must be assigned to self as well (enforced by backend and UI logic).

- Only assigned participants can submit for that role.

- Duplicate prevention: one submission per participant per role.

- Participant edits/assignment changes are blocked after survey end.

- Employees shown in assignment UI are scoped to the survey’s company.

**## 4) Tech stack and architecture**

- Frontend: React (Create React App), axios, served by nginx with SPA fallback.

- Backend: Node.js + Express; SQLite; express-validator; JWT; bcrypt; CORS; dotenv.

- Data model highlights: companies, employees (auth\_code, manager hierarchy), categories, question bank, survey\_questions copied from bank, participants (flat), responses.

- Containers: Docker Compose services for backend (5000) and frontend (3000). Environment variables include FRONTEND\_BASE\_URL and CORS origins to support single-URL flow.

**## 5) Project structure (key paths)**

- backend/

  - index.js — Express app, seeders (question bank, demo survey, participants, responses), public endpoints.

  - package.json — Dependencies and scripts.

- frontend/

  - src/components/SurveyCreator.js — Survey cards, lifecycle actions, Assign Participants, View Survey/Responses.

  - src/components/SurveyTaking.js — Public single-URL flow, email+auth, role selection, answering.

- docs/

  - functional-coverage.md — Detailed functional coverage and seed data details for this branch.

  - Project-Overview-and-Onboarding.md — This document.

- scripts/

  - export-docs.ps1 — Optional helper to export docs to PDF/DOCX via pandoc if installed.

- docker-compose.yml, Dockerfile.backend, Dockerfile.frontend — Container setup.

**## 6) Local setup**

- With Docker (recommended)

  - Build and run backend and frontend containers (backend on 5000, frontend on 3000).

  - Seeders run on first start; single-URL flow will be available for the demo survey.

- Without Docker

  - Install backend deps, run backend.

  - Install frontend deps, run frontend.

  - Ensure CORS and FRONTEND\_BASE\_URL align in env.

See README.md for step-by-step commands.

**## 7) Seed data and demo**

- Companies and employees with a manager hierarchy; unique auth codes generated.

- Question bank with mixed MCQ/Text across categories: Purpose, Personal Development, Presence, People Skills, Practicalities, Principles.

- Demo survey: "Demo 360 Survey (Self/Manager/Peer)" for company 1; active with single URL token.

- Enabled feedback roles: self, manager, peer, reportee.

- Survey questions: per role, small subset copied from the bank (e.g., 2 MCQ + 1 Text) with MCQ options replicated.

- Participants: at least one per role; non-self implies self.

- Responses: pre-seeded for all assigned participants/roles (MCQ first option; text placeholders) so View Responses is populated.

**## 8) API overview (selected endpoints)**

- Public

  - GET /api/validate-survey-token/:token — Validate public URL access.

  - POST /api/authenticate-survey-participant — Email+auth → allowed roles + questions.

  - POST /api/survey-responses — Submit answers; validates assignment; prevents duplicates.

- Admin

  - GET /api/surveys/:id/participants — Retrieve participants by role.

  - PUT /api/surveys/:id/participants — Replace assignments; blocked after end.

  - POST /api/surveys/:id/start — Generate url\_token and activate.

  - PUT /api/surveys/:id/status — Update lifecycle (e.g., end).

  - GET /api/surveys/:id/urls — Return single public URL.

  - GET /api/surveys/:id/responses — List responses grouped by role.

**## 9) Collaboration workflow**

- Branching

  - main: stable baseline; merge via PR after review.

  - feature branches (e.g., 360-feedback): active development.

- Commits/PRs

  - Keep commits scoped and messages descriptive.

  - Reference docs/functional-coverage.md when updating behavior.

- Code style

  - JavaScript/React/Node conventions; prefer small, testable modules.

- Security

  - Do not commit secrets; use .env and sample .env.example.

**## 10) Build, test, and run**

- Build (Docker): use docker-compose to build with --no-cache when needed.

- Lint/tests: add as needed; keep endpoints and UI minimal but robust.

- Smoke test

  - Admin UI: View Survey should show participants and summary counts.

  - Public: Use single URL, authenticate with a seeded employee, pick role, submit.

**## 11) Exporting docs to Word/PDF**

Options:

- Quick (manual)

  - Open the Markdown in Microsoft Word and "Save As" .docx or PDF.

  - Or print to PDF from your Markdown preview.

- Automated (optional, requires pandoc)

  - Use scripts/export-docs.ps1 to generate .pdf and .docx files for this doc and functional-coverage.md.

**## 12) Sharing the project**

- To share code/data as of this branch, create an archive of the project subtree. A sample ZIP was created: SurveyProject-360-feedback-YYYYMMDD.zip in the project root.

- Alternatively, use GitHub to invite collaborators and open PRs to main.

**## 13) Troubleshooting**

- SPA routing: nginx fallback must be enabled for client routes.

- CORS/URLs: FRONTEND\_BASE\_URL must match the deployed frontend URL.

- Seeds not visible: ensure backend container initialized with a clean DB or run seed logic.

References

- docs/functional-coverage.md — deeper functional coverage and seed specifics.