**C. Realization & Technologies Used**

**C1. Backend Implementation (3 pts)**

The backend is a Node.js 20 application that combines Express routing, JWT authentication, session management, and GPT4All orchestration. Core capabilities include:

* **Authentication flow:** Registration captures email and password, hashes the credential with bcryptjs, derives a unique encryption salt, and initializes an encrypted vault. Login verifies the hash, issues an HTTP-only session cookie alongside a JWT containing role claims, and loads the decrypted vault for the client overlay. Logout tears down the session and invalidates the JWT.
* **Vault management:** Authenticated users can fetch the decrypted vault, persist updates, export their data as a JSON bundle, wipe conversation history, or invoke the right-to-be-forgotten workflow. Each write recalculates encryption using AES-256-GCM with per-user keys derived via PBKDF2, so the vault at rest always remains opaque.
* **Assessment and progress tracking:** Endpoints accept structured inputs describing motivations, possession counts, and milestone achievements. The server normalizes payloads, updates phase progression (initiation → reduction → refinement → optimization → maintenance), and persists both numeric telemetry and narrative insights in the vault.
* **Generative AI orchestration:** /api/chat consumes the latest message and vault snapshot, determines the appropriate coaching approach, injects emotional safety directives when needed, and streams GPT4All responses. Rate limiting protects the route from abuse, and Socket.IO mirrors stream updates to the UI.
* **Administrative analytics:** An admin-only summary endpoint aggregates totals (active users, phase distribution, cleared milestones) while deliberately excluding personal identifiers, enabling compliance-friendly reporting.

**C2. Frontend Implementation (3 pts)**

The frontend consists of three functional views built with vanilla JavaScript and a shared authentication overlay:

* **Index / Chat preview:** The entry page hosts the login overlay and a teaser of the chat experience. Once authenticated, it displays welcome messaging and links to the assessment and dashboard sections.
* **Assessment workflow:** A guided, multi-step form walks users through motivation discovery, inventory auditing, attachment reflections, and baseline goal setting. Each step persists directly through the Auth helper, guaranteeing that the encrypted vault mirrors user inputs in real time.
* **Progress dashboard:** The dashboard presents current item counts, phase summaries, milestone history, and (for admins) aggregated community metrics. It listens for vault:updated events so charts and summaries refresh whenever the vault changes from another page.

Shared modules power the UI:

* auth.js renders the overlay, coordinates login/register flows, stores session tokens, and exposes helper methods to read/write vault data while dispatching DOM events.
* dashboard.js, assessment.js, and enhanced-chat.js subscribe to those events, transform vault data into view models, and render accessible UI components styled by the minimalist design system.

Responsive CSS ensures the application remains usable on desktops and tablets, and token streaming in the chat view creates an engaging conversational feel without external dependencies beyond GPT4All.