Cohere Chat & Moderation Integration

This README explains the server-side PHP integration for the Cohere Chat + Moderation endpoints used by the front-end chat widget.

Important environment variables and configuration

- COHERE_API_KEY: Required. Server environment variable that contains your Cohere API key.
  - Can be provided as standard OS env or in `config/.env` by creating `COHERE_API_KEY=sk-...` (no quotes).
- COHERE_MODEL: Optional. Default: `command-a-03-2025`.
- COHERE_SYSTEM_PROMPT: Optional. Defaults to a mediator-focused system prompt. If you want 'optique' behavior, set this string in your environment.
- COHERE_DEBUG: Optional (true/false). When enabled, more detailed error messages appear in JSON responses.

Endpoints

- POST /TasnimCRUD/api/chatbot.php

  - Body: JSON. Accepts either a `message` (string) or `messages` (array of `{role, content}`) to match the C++ client pattern.
  - Optional: `model` (string) to override the default per request.
  - Optional: `context` (object) contains metadata (e.g., event summaries)
  - Response: { ok: true, reply: '...' } or { ok: false, error: '...' }
    - The endpoint also returns a Cohere-ish chat-completion shape for compatibility with C++ clients:
      - `id`, `object`, `created`, `model` (top-level)
      - `message`: object, `message.content[]` contains assistant content (e.g., `message.content[0].text`) similar to Cohere `/v1/chat` responses
      - `text`: top-level condensed assistant reply (string) for quick use
      - `choices`: array; `choices[0].message.content[0].text` contains the assistant reply (if present)
      - `cohere.body`: raw JSON returned by Cohere (useful for debugging; safe; does not contain the API key)
  - Error handling: If the server lacks configuration (COHERE_API_KEY missing), response returns 500 {ok:false, error:'Configuration serveur manquante'}; if COHERE_DEBUG=1 then the exception message is appended.

- POST /TasnimCRUD/api/moderation.php

  - Body: { text: '...' }
  - Response: {ok:true, flag: true|false, scores: { hate: 0.2, insult: 0.1 } }.

Config & Files

- `api/cohere_config.php`: central helper for reading environment, a lightweight `.env` loader (`config/.env`), log rotation, cohere_post wrapper and helper utilities.

- `config/cohere.key`: Optional fallback file (plain key content). If both `COHERE_API_KEY` and `config/cohere.key` are missing, the server will throw an error.

Logging

- Logs are written to `logs/cohere.log` with rotation when >1MB.
- DO NOT put secrets or API keys in logs.

Front-end Notes

- The widget code is at `assets/js/events.js`. Use `sendToChatbot(messageOrMessages, onReply, onError, model)` to call the server.
  - `messageOrMessages` can be a string (message param) or an array (messages array) as in the C++ client.
- For moderation call use `checkModeration(text)` which returns a promise resolving to `{allowed: boolean, scores: {...}}`.

Example .env
COHERE_API_KEY=sk-...YOUR_KEY_HERE...
COHERE_MODEL=command-a-03-2025
COHERE_SYSTEM_PROMPT=optique
COHERE_DEBUG=1

Security

- Do not commit or paste the API KEY anywhere in the project or version control.
- Keep `config/cohere.key` outside of version control and restrict file permissions.

Testing

- Use `curl` or Postman to test endpoints:
  - curl -X POST -H 'Content-Type: application/json' -d '{"message":"Bonjour"}' http://localhost/TasnimCRUD/api/chatbot.php
  - curl -X POST -H 'Content-Type: application/json' -d '{"text":"message potentiel offensant"}' http://localhost/TasnimCRUD/api/moderation.php

Questions or Changes

- If you'd like the default system prompt to be changed (e.g., to the exact 'optique' prompt from your C++ sample), update the `COHERE_SYSTEM_PROMPT` environment variable or the `cohere_get_system_prompt` helper.