

BÁO CÁO KỸ THUẬT & TIỀN ĐỘ PHÁT TRIỂN AI CHATBOT DU LỊCH

Phiên bản: 5.3 (Accent Preservation & Enhanced Triggers)

Ngày cập nhật: 26/11/2025

Trạng thái: Stable (Ôn định) - Đã khắc phục lỗi mất dấu tiếng Việt và thiếu từ khóa kích hoạt.

1. TỔNG QUAN NÂNG CẤP (v5.3)

Hệ thống đã được tinh chỉnh để xử lý ngôn ngữ tự nhiên tiếng Việt chính xác hơn, khắc phục hoàn toàn tình trạng AI "trả lời chung chung" khi người dùng hỏi chi tiết.

Các nâng cấp cốt lõi

1. Accent Preservation (Bảo toàn dấu tiếng Việt):

- Vấn đề cũ:* Hệ thống chuẩn hóa bỏ dấu ("Phở bò" -> "pho bo") khiến MongoDB không tìm thấy item gốc "Phở bò" (có dấu).
- Giải pháp v5.3:* Logic tìm kiếm giữ nguyên dấu của từ khóa quan trọng, chỉ loại bỏ ký tự rác.
- Ví dụ:* User hỏi "Phở bò Hà Nội" -> Hệ thống tìm "Phở bò" (có dấu) -> **Khớp chính xác**.

2. Extended Trigger Keywords (Mở rộng từ khóa kích hoạt):

- Vấn đề cũ:* Các từ như "ngon không", "đẹp không", "review" chưa được đưa vào danh sách kích hoạt chế độ "Chi tiết" (Specific Mode).
- Giải pháp v5.3:* Bổ sung regex mạnh mẽ để nhận diện ý định hỏi chi tiết: review, đánh giá, ngon, tốt, đẹp, hay, thế nào, ra sao, có không.

3. Word Boundary Logic (Ranh giới từ):

- Sử dụng Regex \b để tránh cắt nhầm từ.
- Ví dụ:* Từ khóa "bán" sẽ không cắt cụt chữ "Bánh" trong cụm "Bánh căn".

2. LUỒNG XỬ LÝ TÌM KIẾM (MỚI)

(Mô tả quy trình xử lý dữ liệu từ đầu vào đến đầu ra)

1. **User Input:** "Phở bò Hà Nội ngon không?"
 2. **Cleaner v5.3:** Làm sạch câu hỏi, giữ nguyên dấu tiếng Việt, loại bỏ từ khóa rác.
 3. **Search Variations:** Tạo biến thể tìm kiếm (Ví dụ: tách "Hà Nội" ra khỏi "Phở bò").
 4. **Scan DB (Case-Insensitive):** Quét Database Mongo với từ khóa có dấu.
 5. **Match Found:** Tìm thấy item "Phở bò".
 6. **Thinking Mode:** Kích hoạt chế độ suy luận AI.
 7. **AI Response:** Trả về bài review chi tiết về Phở bò.
-

3. CẤU TRÚC SYSTEM PROMPT

Để đảm bảo chất lượng câu trả lời, hệ thống tuân thủ các quy tắc sau:

- **NO EMOJI:** Tuyệt đối không sử dụng biểu tượng cảm xúc (icon/emoji) trong văn bản.
 - **CONTEXT AWARE (Nhận thức ngữ cảnh):** Nếu dữ liệu thuộc tinh gộp (merged data), AI phải giải thích rõ ràng. (Ví dụ: "Món này là đặc sản ở Quy Nhơn, thuộc dữ liệu quản lý của tỉnh Gia Lai").
 - **FORMAT (Cấu trúc):**
 - **Summary:** 3-4 câu văn xuôi, xác nhận địa điểm và đánh giá tổng quan.
 - **Tips:** Các mẹo thực tế (giờ ăn, quán ngon, lưu ý khi di chuyển).
-

4. BẢNG TEST CASE KIỂM CHỨNG (v5.3)

Nhóm A: Accent & Keyword Testing (Kiểm tra dấu câu & Từ khóa)

Mục tiêu: Đảm bảo các câu hỏi có dấu hoặc từ khóa đánh giá đều tìm ra kết quả chính xác.

STT	Câu hỏi (User)	Vấn đề phiên bản cũ	Kết quả v5.3	Trạng thái
A1	"Phở bò Hà Nội ngon không?"	Query "pho bo" (mất dấu) không khớp DB "Phở bò".	Query "Phở bò" (có dấu) khớp DB. Keyword "ngon" kích hoạt chế độ Chi tiết.	<input checked="" type="checkbox"/> PASS
A2	"Review Bánh căn Phan Thiết"	Bị cắt từ "Bánh" thành "h" do từ khóa "bán".	Dùng \b bảo vệ từ "Bánh". Tìm ra item "Bánh căn".	<input checked="" type="checkbox"/> PASS

STT	Câu hỏi (User)	Vân đề phiên bản cũ	Kết quả v5.3	Trạng thái
A3	"Tháp Đôi có đẹp không"	Keyword "đẹp" bị bỏ qua -> Trả về danh sách.	Keyword "đẹp" kích hoạt <code>isDetailQuery</code> -> Trả về review Tháp Đôi.	<input checked="" type="checkbox"/> PASS

Nhóm B: Data-Driven & Merged Context (Dữ liệu gộp)

Mục tiêu: Kiểm tra khả năng định vị địa lý của AI trong dữ liệu gộp.

STT	Câu hỏi	Logic xử lý	Mong đợi đầu ra
B1	"Mô tả Eo Gió"	Doc: <code>province_gia_lai</code> . Merged: ["Bình Định"].	AI: "Eo Gió nằm ở Quy Nhơn, Bình Định (hiện thuộc dữ liệu vùng Gia Lai mở rộng)..."
B2	"Tìm hiểu Mũi Né"	Doc: <code>province_lam_dong</code> . Merged: ["Bình Thuận"].	AI: "Mũi Né là thiên đường resort ở Bình Thuận..."

5. CÁC FILE ĐÃ CẬP NHẬT

1. File `chatbot.service.js`

- **Hàm `scanItemInDB`:**
 - Thêm logic \b (Word Boundary) để tránh cắt nhầm chữ.
 - **QUAN TRỌNG:** Loại bỏ hàm `normalize()` ở bước tạo biến thể tìm kiếm để **giữ nguyên dấu tiếng Việt**, giúp khớp chính xác với MongoDB.
- **Hàm `suggestHybrid`:**
 - Tích hợp luồng tìm kiếm mới, ưu tiên `scanItemInDB` để bắt các câu hỏi chi tiết ngay từ đầu.

2. File `composer.service.js`

- **Hàm `findSpecificItem`:**
 - Cập nhật Regex `isDetailQuery` bổ sung thêm các từ khóa: `ngon`, `tốt`, `đẹp`, `review`, `đánh giá`.
- **Hàm `composeSpecificItem`:**

- Cập nhật Prompt để AI nhận diện được trường `merged_from` và xử lý ngữ cảnh địa lý tốt hơn.