

# Documentation Interne

*Prototype d'agent  
conversationnel pour les  
séniors*

# Documentation Interne

*Prototype d'agent conversationnel pour les  
séniors*

Cette documentation vise à fournir une vision claire et structurée de l'architecture technique de l'application, son fonctionnement et ses modules, pour permettre une maintenance efficace par une tierce personne.

## Table des matières

1. Introduction générale.....	5
2. Architecture générale.....	5
3. Description des modules.....	5
Module : formulaire.....	5
Module : profil.....	5
Module : Principale.....	5
Module : Paramètres.....	6
Module : Contact.....	6
Module : Historique.....	6
Module : Commun.....	6
Module : Aide.....	7
4. Documentation connexe.....	7

# 1. Introduction générale

L'application est un prototype web destiné aux seniors. Cette application comporte une interface conversationnelle, des fonctions de personnalisation, une intégration de la météo et un historique de conversation.

## 2. Architecture générale

L'application est construite en HTML/CSS/JavaScript et utilise localStorage, une API météo externe (WeatherAPI) et le modèle Ollama qui utilise le LLM Llama 2 pour les réponses conversationnelles.

## 3. Description des modules

### Module : formulaire

**Objectif : Formulaire d'évaluation du profil utilisateur**

- Dépendances : formulaire.html, formulaire.css, formulaire.js
- Saisie des données de profil (santé, logement, activité animaux, contact et alimentation.)
- Variables : localStorage['formulaireData']

### Module : profil

**Objectif : Affichage du profil utilisateur**

- Dépendances : profil.html profil.css, profil.js, navbar.css, common.js
- Lit les données depuis localStorage
- Procédure : updateProfile(data)
- Variables : localStorage['formulaireData']

### Module : Principale

**Objectif : Page principale avec affichage météo, chatbot et aperçu de l'historique**

- Dépendances : index.html, index.css, index.js, common.js, script.js
- Variables : localStorage['mainAvatar'], localStorage['chatHistory'], localStorage['selectedBubble']

- Procédures : updateBubble(), wrapText(), displayHistory(), generatePreventionMessage(), displayWeather(), handleSendMessage(), envoyer(), manageHistory()

## Module : Parametres

### Objectif : Interface de personnalisation

- Dépendances : settings.html, settings.css, settings.js, navbar.css, common.js
- Permet de personnaliser le type de police ainsi que sa taille, le fond d'écran, le style de la bulle et enfin le style de l'avatar
- Variables : localStorage['selectedFont'], localStorage['selectedFontSize'], localStorage['selectedBubble'], localStorage['mainAvatar']
- Procédures : applySettings(), resetSettings()

## Module : Contact

### Objectif : Interface de contact

- Dépendances : contact.html, contact.css, contact.js, navbar.css, common.js
- Permet d'appeler et d'afficher les numéros d'urgence, permet aussi d'ajouter des numéros manuellement.
- Procédure : loadContacts()
- Variables : localStorage['customContacts']

## Module : Historique

### Objectif : Consultation et gestion des conversations

- Dépendances : historique.html, historique.css, historique.js, navbar.css, common.js
- Affiche et permet de modifier les conversations
- Variables : localStorage['messageHistory'], localStorage['historyLastId']
- Procédures : renderHistoryTable(), editHistory()

## Module : Commun

- Dépendances : common.js
- Gère les paramètres d’affichage globaux (navbar, fond écran police etc ...)
- Procédures : updateNavbarAvatar(), applySavedBackground(), applySavedFont()
- Variables : localStorage['selectedFontSize'], localStorage['selectedFont'], localStorage['selectedBackground'], localStorage['navbarAvatar']

## Module : Aide

**Objectif : Aider les utilisateurs dans leur navigation sur le site**

- Dépendances : help.html, help.css, help.js, navbar.css, common.js
- Affichage textuelle explicatif concernant les clefs navigations
- Variables : localStorage['navbarAvatar']

## 4. Documentation connexe

- Cahier des charges – Définition des besoins fonctionnels
- Cahier de recette – Tests fonctionnels détaillés
- Documentation WeatherAPI et Llama 2
- Maquettes Figma – Référentiel visuel
- Visuel du Site – Captures d’écrans