

Rapport de Projet

Mira

Interface Angular pour un Chatbot Darija



Sujet : Conception d'une interface frontend Angular pour un chatbot intelligent en Darija marocaine

Réalisé par :

1er groupe : Chaimae Sayahi & Aya Saidi

2e groupe : Mohamed Amine Tahri & Khalid Aboulfaraj

3e groupe : Mouad El Abbassi & Mohamed Agoulzi

Encadré par :

Pr. OURDOU Amal

Année universitaire : 2025–2026

Table des matières

1	Introduction	3
1.1	Contexte du projet	3
1.2	Objectif général	3
2	Organisation du travail	4
2.1	Répartition générale	4
2.2	Méthodologie	4
3	Présentation du projet Mira	5
3.1	Présentation de Mira	5
3.2	Rôle de l'interface	5
3.3	Modèle de langage	5
4	Architecture générale	6
4.1	Architecture prévue	6
4.2	Description des couches	6
4.2.1	Frontend Angular	6
4.2.2	Backend Next.js	6
4.2.3	Base de données PostgreSQL	6
4.2.4	Modèle Hugging Face	6
4.3	État actuel de l'intégration	7
5	Technologies utilisées	8
5.1	Angular	8
5.2	TypeScript	8
5.3	SCSS	8
5.4	Next.js	8
5.5	PostgreSQL	9
5.6	Hugging Face	9
6	Réalisation de l'interface Angular	10
6.1	Structure du projet Angular	10

6.2	Composants principaux	10
6.3	Interface d'accueil	10
6.4	Interface après l'envoi d'un message	11
6.5	Sidebar	12
6.6	Identité visuelle	12
7	Fonctionnalités réalisées	13
7.1	Fonctionnalités frontend	13
7.2	Fonctionnalités simulées	13
8	Tests et validation	14
8.1	Lancement du projet	14
8.2	Compilation	14
8.3	Tests fonctionnels	14
9	Limites de la version actuelle	15
10	Travaux futurs	16
10.1	Connexion avec le backend	16
10.2	Connexion à PostgreSQL	16
10.3	Intégration du modèle Darija	16
10.4	Authentification	16
10.5	Amélioration de l'expérience utilisateur	16
11	Conclusion	18

1 Introduction

Dans le cadre de notre projet, nous avons travaillé sur la conception d'un chatbot intelligent capable de communiquer avec les utilisateurs en Darija marocaine. L'objectif global du projet est de proposer une solution conversationnelle simple, moderne et adaptée au contexte linguistique marocain.

Notre travail est organisé autour de deux parties complémentaires. La première partie concerne le modèle de langage, que nous avons déjà entraîné afin de produire des réponses en Darija. La deuxième partie concerne l'interface utilisateur, qui permet à l'utilisateur final d'interagir avec ce modèle d'une manière claire, fluide et professionnelle.

Dans ce rapport, nous nous concentrons principalement sur la réalisation de l'interface frontend avec Angular. Cette interface porte le nom **Mira**. Elle représente la couche visible du projet et prépare l'intégration future avec le backend et le modèle déjà entraîné.

1.1 Contexte du projet

La Darija marocaine est une langue largement utilisée dans la communication quotidienne au Maroc, mais elle reste moins représentée dans les systèmes conversationnels modernes. Notre projet vise donc à rapprocher les technologies d'intelligence artificielle des utilisateurs marocains en proposant une interface simple et naturelle.

Nous avons choisi de construire une interface moderne inspirée des assistants conversationnels actuels, tout en gardant une identité visuelle propre au projet. L'interface a été développée avec Angular afin de respecter les exigences techniques du projet.

1.2 Objectif général

L'objectif général du projet est de construire un chatbot capable de dialoguer avec l'utilisateur en Darija. Pour cela, nous avons déjà travaillé sur l'entraînement d'un modèle de langage spécialisé. L'interface actuelle constitue la partie frontend du projet. Elle permet de préparer l'expérience utilisateur avant la connexion finale avec le backend et le modèle.

2 Organisation du travail

Notre équipe est composée de six membres répartis en trois groupes de deux personnes. Cette organisation nous a permis de diviser le travail en plusieurs parties tout en gardant une cohérence globale dans le projet.

2.1 Répartition générale

Le travail a été réparti selon les axes suivants :

- préparation et entraînement du modèle de langage en Darija ;
- conception de l'architecture globale du chatbot ;
- développement de l'interface utilisateur avec Angular ;
- préparation du backend et de la base de données ;
- préparation de l'intégration future avec le modèle hébergé sur Hugging Face ;
- tests, validation et rédaction du rapport.

2.2 Méthodologie

Nous avons adopté une approche progressive. Dans un premier temps, nous avons travaillé sur le modèle et la structure backend. Ensuite, nous avons développé une interface frontend indépendante afin de valider l'expérience utilisateur avant l'intégration finale.

Cette méthode permet de séparer clairement la partie présentation, la partie logique backend et la partie intelligence artificielle.

3 Présentation du projet Mira

3.1 Présentation de Mira

Mira est une interface de chatbot conçue pour permettre aux utilisateurs de poser des questions de manière simple et naturelle. L'objectif est d'offrir une expérience proche des assistants modernes comme ChatGPT ou Claude, tout en préparant une intégration avec un modèle spécialisé en Darija.

L'interface actuelle est volontairement minimaliste. Elle met l'accent sur le champ de saisie, la lisibilité des messages et la simplicité de l'interaction.

3.2 Rôle de l'interface

L'interface développée dans cette version a pour rôle de :

- présenter une interface moderne pour le chatbot ;
- permettre à l'utilisateur de saisir un message ;
- afficher les messages utilisateur et les réponses du chatbot ;
- simuler localement une réponse de Mira ;
- préparer la future connexion au backend ;
- respecter l'exigence d'utilisation d'Angular pour la partie frontend.

3.3 Modèle de langage

Le modèle de langage destiné au chatbot a déjà été entraîné pour répondre en Darija. Ce modèle est prévu pour être hébergé et utilisé via Hugging Face. Dans cette version du projet, l'interface Angular n'est pas encore connectée directement au modèle. Elle représente la couche frontend prête à être reliée au backend.

Cette séparation est volontaire afin de sécuriser le développement. Elle nous permet de valider l'interface utilisateur avant d'ajouter les communications réseau, l'authentification et les appels au modèle.

4 Architecture générale

4.1 Architecture prévue

L'architecture globale du projet est organisée en plusieurs couches :

Frontend Angular $\longrightarrow$ **Backend Next.js** $\longrightarrow$ **PostgreSQL** $\longrightarrow$ **Modèle Hugging Face**

4.2 Description des couches

4.2.1 Frontend Angular

La partie frontend est développée avec Angular. Elle contient l'interface utilisateur, les composants visuels, la logique locale d'affichage des messages et la structure responsive de l'application.

4.2.2 Backend Next.js

Le backend existant est prévu pour gérer les fonctionnalités serveur, notamment l'authentification, les routes API, la communication avec la base de données et l'intégration avec le modèle de langage.

4.2.3 Base de données PostgreSQL

PostgreSQL est prévu pour stocker les utilisateurs, les conversations, les messages et les historiques. Dans la version actuelle de l'interface Angular, la persistance n'est pas encore activée directement depuis le frontend.

4.2.4 Modèle Hugging Face

Le modèle entraîné pour la Darija sera utilisé dans une étape suivante via le backend. Le frontend Angular enverra les messages au backend, qui se chargera ensuite de transmettre la requête au modèle et de renvoyer la réponse à l'utilisateur.

4.3 État actuel de l'intégration

Dans l'état actuel du projet, l'interface Angular fonctionne comme une démonstration frontend. Elle utilise un état local et une réponse simulée pour représenter le comportement attendu du chatbot. La connexion réelle avec le backend et le modèle constitue la prochaine étape.

5 Technologies utilisées

5.1 Angular

Angular est utilisé pour construire l'interface frontend. Il permet de structurer l'application en composants réutilisables et de gérer l'état de l'interface de manière claire.

Les principaux avantages d'Angular dans notre projet sont :

- architecture basée sur des composants ;
- utilisation de TypeScript ;
- bonne organisation du code ;
- facilité de maintenance ;
- adaptation aux applications frontend complexes.

5.2 TypeScript

TypeScript permet de renforcer la fiabilité du code grâce au typage statique. Dans notre interface, il est utilisé pour définir les messages, les conversations et les services de gestion d'état.

5.3 SCSS

SCSS est utilisé pour styliser l'interface. Il permet d'organiser les styles de manière plus propre et de construire une interface moderne avec des couleurs, des bordures, des animations et des effets visuels.

5.4 Next.js

Next.js reste prévu comme couche backend/API. Même si l'interface utilisateur finale est développée avec Angular, le backend existant pourra être conservé pour gérer les fonctionnalités serveur.

5.5 PostgreSQL

PostgreSQL est prévu pour la persistance des données. Il permettra de stocker les conversations, les utilisateurs et les messages.

5.6 Hugging Face

Hugging Face est prévu pour héberger ou servir le modèle de langage entraîné. L'intégration sera faite dans une prochaine étape à travers le backend.

6 Réalisation de l'interface Angular

6.1 Structure du projet Angular

La partie frontend est située dans le dossier :

```
frontend-angular/
```

La structure générale est la suivante :

```
frontend-angular/  
|-- angular.json  
|-- package.json  
|-- public/  
| |-- branding/  
| | |-- mira-logo.png  
|-- src/  
| |-- app/  
| |-- styles.scss  
| |-- index.html
```

6.2 Composants principaux

L'interface est organisée autour de plusieurs composants :

- **Sidebar** : contient le logo, le bouton nouveau chat et les actions utilisateur ;
- **Greeting** : affiche le message d'accueil ;
- **Composer** : représente le champ de saisie ;
- **Message List** : affiche les messages de l'utilisateur et les réponses de Mira ;
- **Chat Shell** : organise l'ensemble de la page.

6.3 Interface d'accueil

L'écran d'accueil contient un message simple :

Salam, ana Mira

Le champ de saisie est placé au centre afin d’orienter directement l’utilisateur vers l’action principale : poser une question. Cette approche est inspirée des interfaces modernes de chatbot, où la simplicité et la concentration sur la conversation sont prioritaires.

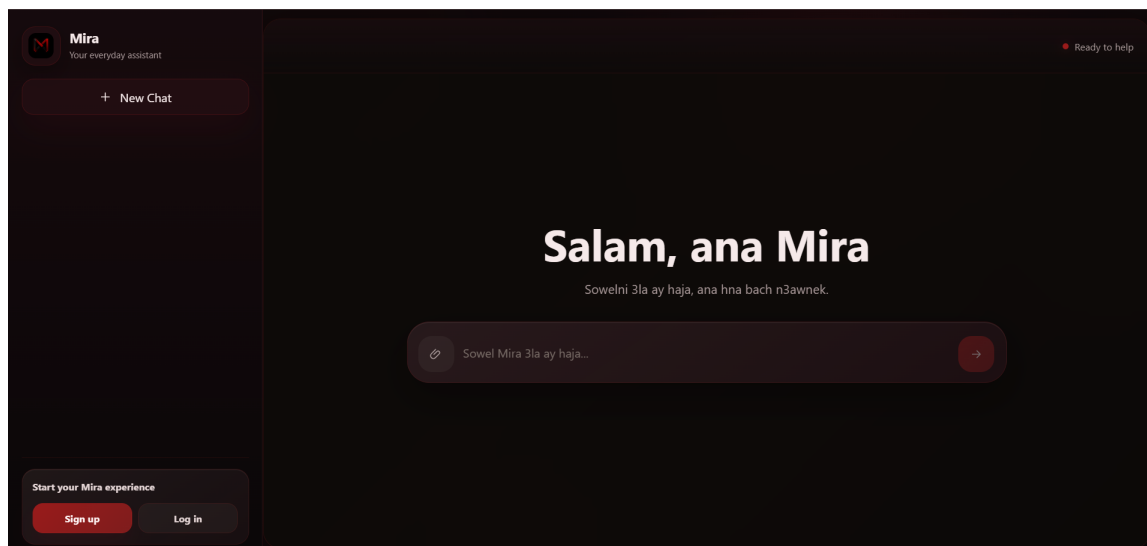


FIGURE 6.1 – Interface d’accueil de Mira

6.4 Interface après l’envoi d’un message

Après l’envoi du premier message, l’interface passe à un mode conversationnel. Les messages sont affichés dans la zone centrale et le champ de saisie se déplace vers le bas, comme dans les assistants modernes.

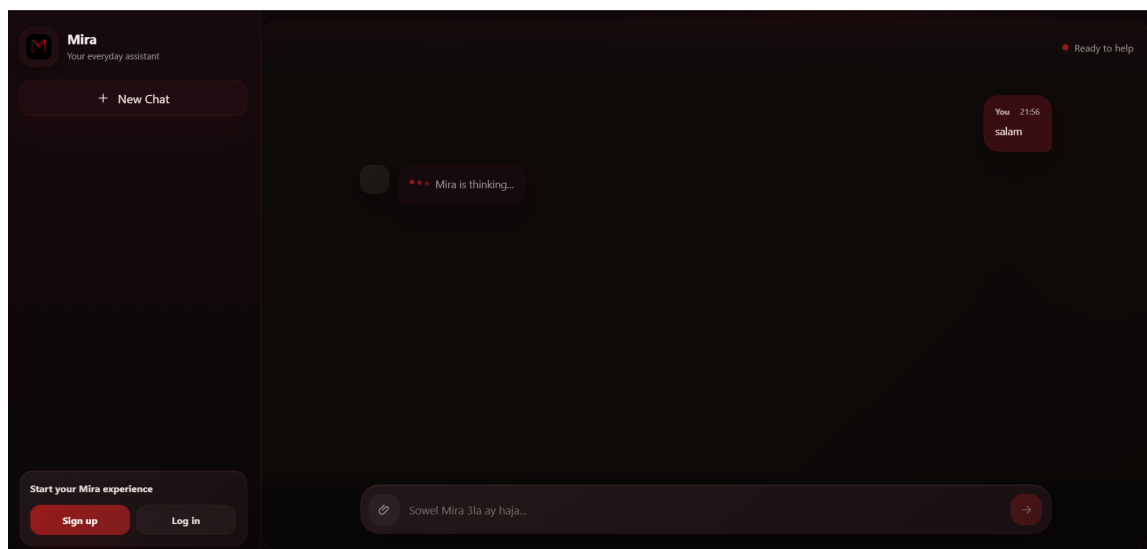


FIGURE 6.2 – Interface de conversation

6.5 Sidebar

La sidebar contient l'identité visuelle de Mira et les actions principales. Nous avons supprimé les historiques fictifs afin d'éviter de présenter une fonctionnalité non connectée à la base de données dans cette version.

La sidebar contient principalement :

- le logo Mira ;
- le nom de l'application ;
- le bouton *New Chat* ;
- les boutons *Sign up* et *Log in* comme éléments frontend.

6.6 Identité visuelle

Nous avons choisi un thème sombre professionnel basé sur :

- un fond noir ou charcoal ;
- des surfaces sombres ;
- des accents rouge foncé ;
- une typographie claire ;
- des bordures discrètes.

Ce choix permet de donner à l'interface une apparence plus professionnelle et moins artificielle.

7 Fonctionnalités réalisées

7.1 Fonctionnalités frontend

Les fonctionnalités suivantes ont été réalisées dans la version Angular :

- affichage d'une interface chatbot moderne ;
- affichage du logo et de l'identité Mira ;
- champ de saisie centré sur l'écran d'accueil ;
- envoi local d'un message ;
- affichage des messages utilisateur ;
- génération d'une réponse simulée de Mira ;
- bouton nouveau chat ;
- design responsive ;
- boutons frontend *Sign up* et *Log in*.

7.2 Fonctionnalités simulées

Certaines fonctionnalités sont simulées localement afin de préparer le comportement final :

- réponse du chatbot ;
- historique de conversation ;
- état de connexion utilisateur.

Ces éléments seront remplacés par des appels réels au backend dans la prochaine phase.

8 Tests et validation

8.1 Lancement du projet

Pour lancer le projet Angular, nous utilisons les commandes suivantes :

```
cd frontend-angular  
pnpm install  
ng serve
```

L'application est ensuite disponible à l'adresse :

```
http://localhost:4200
```

8.2 Compilation

Nous avons testé la compilation du projet avec la commande :

```
pnpm build
```

La compilation a été effectuée avec succès, ce qui confirme que l'interface Angular ne contient pas d'erreurs bloquantes au niveau TypeScript, template ou SCSS.

8.3 Tests fonctionnels

Nous avons vérifié les points suivants :

- l'interface s'affiche correctement ;
- le logo est visible ;
- le champ de saisie fonctionne ;
- l'envoi d'un message ajoute le message dans la conversation ;
- Mira affiche une réponse simulée ;
- le bouton *New Chat* réinitialise la conversation ;
- l'interface reste lisible sur différentes tailles d'écran.

9 Limites de la version actuelle

La version actuelle représente la partie frontend Angular. Elle ne constitue pas encore une version finale entièrement connectée au backend.

Les principales limites sont :

- les messages ne sont pas encore stockés dans PostgreSQL ;
- l'authentification n'est pas encore connectée depuis Angular ;
- les boutons *Sign up* et *Log in* sont encore des éléments frontend ;
- les réponses de Mira sont simulées localement ;
- le modèle Hugging Face n'est pas encore appelé depuis l'interface Angular.

Ces limites sont normales à cette étape, car l'objectif principal de cette version est de valider l'interface Angular.

10 Travaux futurs

10.1 Connexion avec le backend

La prochaine étape consiste à connecter l'interface Angular avec le backend Next.js déjà préparé. Cette connexion permettra d'utiliser les routes API existantes pour gérer les messages, les utilisateurs et les conversations.

10.2 Connexion à PostgreSQL

Une fois le backend connecté, les conversations pourront être sauvegardées dans PostgreSQL. Cela permettra aux utilisateurs de retrouver leur historique et de gérer leurs anciennes discussions.

10.3 Intégration du modèle Darija

Le modèle de langage entraîné pour répondre en Darija sera connecté via le backend. L'utilisateur enverra un message depuis Angular, le backend le transmettra au modèle, puis la réponse sera renvoyée à l'interface.

10.4 Authentification

Les boutons *Sign up* et *Log in* seront reliés au système d'authentification. Cela permettra de distinguer les utilisateurs anonymes des utilisateurs connectés.

10.5 Amélioration de l'expérience utilisateur

D'autres améliorations pourront être ajoutées :

- historique réel des conversations ;
- gestion des profils ;
- mode clair/sombre ;
- meilleure gestion des erreurs ;

- streaming des réponses ;
- amélioration de l'accessibilité.

11 Conclusion

Dans ce projet, nous avons réalisé une interface frontend Angular pour le chatbot Mira. Cette interface constitue une étape importante dans la construction d'un chatbot intelligent capable de communiquer avec les utilisateurs en Darija.

Nous avons déjà travaillé sur l'entraînement du modèle de langage destiné à la Darija. L'interface développée dans ce rapport prépare l'intégration de ce modèle dans une application complète. Elle propose une expérience utilisateur moderne, minimaliste et professionnelle.

La version actuelle se concentre volontairement sur la partie frontend afin de répondre à l'exigence Angular. Les prochaines étapes concernent la connexion avec le backend existant, l'utilisation de PostgreSQL pour la persistance des données et l'intégration du modèle hébergé sur Hugging Face.

Ce travail nous a permis de séparer clairement les responsabilités du projet : Angular pour l'interface utilisateur, Next.js pour le backend, PostgreSQL pour la base de données et Hugging Face pour l'intelligence artificielle.