

## Practical Exercise: Building an AI Chatbot with Next.js and Django

### Objective:

The goal of this exercise is to assess your skills in backend development (Django) and frontend development (Next.js), as well as your ability to integrate a generative AI solution (OpenAI API or an open-source model). This project allows you to demonstrate your technical expertise and creativity in an AI-driven application.

### General Guidelines:

#### 1. Required Technologies:

- Backend: Django (Django Rest Framework)
- Frontend: Next.js (React)
- Database: PostgreSQL
- AI API: OpenAI API (or an Open-Source Model on Hugging Face like Llama or Mistral )

#### 2. Mandatory Features

##### A. Frontend (Next.js) – User Interface:

- Chat interface that allows users to send messages to the AI and display responses.
- Conversation history stored in the database and displayed on the UI.
- Dark mode and responsive design for a better user experience.

##### B. Backend (Django) – REST API:

- Endpoint to send a request to the AI and return a response.
- Store conversation history in the database (Django model with user, prompt, and response).

### 3. Optional Features (Bonus Points):

- Candidates can implement additional improvements or extra features. Some ideas include:
- Dark mode and responsive design for a better user experience.
- User profiles with personalized AI settings.
- Multi-language support (detect language and respond accordingly).
- File upload feature (allow users to send documents for AI analysis).
- ...

### 4. Constraints & Expectations

- The user interface should be functional and intuitive.
- The code should be well-structured, readable, and follow best practices.
- The database should be properly modeled to support the required features.
- You should also provide a minimal documentation (README) explaining:
- How to install and run the project locally.
- Any additional features you implemented.

### Deliverables

- Project source code hosted on a Git repository (GitHub).
- A README.md file containing:
- Instructions for setting up and running the project.
- Technical and architectural decisions.
- Explanation of any additional features implemented.

### Exercise Duration:

You have until the deadline communicated during your technical interview to complete this exercise. If you encounter any difficulties, document your decisions and explain your limitations in the README.

### Evaluation Criteria:

- Code quality (clean, maintainable, and well-structured).
- Readability and clarity of the implementation.
- Adherence to best practices (for Django, Next.js, and API development).
- Functionality and completeness of the solution.
- Database design and data management.
- Innovation and additional features beyond the basic requirements.
- Clarity and completeness of the README file.