

Создание локального чат-бота с подключением искусственного интеллекта с помощью Ollama

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс разработки локального чат-бота, использующего технологии искусственного интеллекта на основе платформы Ollama. Основное внимание уделяется этапам настройки среды, интеграции модели ИИ и обеспечению автономной работы без постоянного подключения к внешним сервисам. Приведены примеры реализации, описаны архитектура решения и использованные инструменты. Также анализируются преимущества и ограничения подхода локального размещения моделей, в том числе с точки зрения конфиденциальности данных и производительности. Работа может быть полезна разработчикам и исследователям, интересующимся созданием автономных интеллектуальных систем.

Ключевые слова: чат-бот, искусственный интеллект, локальная модель, Ollama, автономная система, конфиденциальность данных, производительность, интеграция ИИ, архитектура решения, разработка.

Creating a local chatbot with artificial intelligence using Ollama

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the process of developing a local chatbot using artificial intelligence technologies based on the Ollama platform. The main focus is on the steps of setting up the environment, integrating the AI model, and ensuring autonomous operation without constant connection to external services. The article provides examples of implementation, describes the solution architecture, and outlines the tools used. It also analyzes the advantages and limitations of the local model deployment approach, including data privacy and performance considerations. This work can be useful for developers and researchers interested in creating autonomous intelligent systems.

Keywords: chatbot, artificial intelligence, local model, Ollama, autonomous system, data privacy, performance, AI integration, solution architecture, development.

1 Введение

1.1 Актуальность

В последние годы стало заметно, что разговор с компьютером или программой уже никого не удивляет — чат-боты активно внедряются в повседневную жизнь. Тем не менее, большинство из них работают через интернет и завязаны на сторонние сервисы, что вызывает опасения по поводу приватности и надежности. Особенно это важно, если речь идет о конфиденциальной информации или необходимости круглосуточной доступности без подключения к сети. Поэтому всё больше разработчиков задумываются над тем, как запустить умного помощника прямо на своём компьютере или сервере, без участия облаков. Инструмент Ollama как раз дает такую возможность — запускать модели искусственного интеллекта локально. Это направление заслуживает внимания, поскольку сочетает в себе и удобство, и безопасность, и гибкость в использовании.

1.2 Обзор исследований

И. Н. Куринин, В. И. Нардюжев, И. В. Нардюжев в своей работе описали интеграцию Ollama в систему «Умный дом» описали метод использования, инструменты, а также предложили варианты улучшения модели [1]. С.В.Снимщиков, И. П. Саврасов, Е. В. Сумароков в статье «Информационная модель алгоритма применения машинопонимаемых нормативных документов при проектировании строительных конструкций» описали метод использования Ollama для распознавания документальной части в более понятной обычному человеку форме [2]. Авторы статьи Е. А. Копылов, Е.А.Дзюба создали нейроконсультанта для помощи в ответе на тяжелые вопросы с применением Ollama и реализацией продукта на локальной машине [3]. А. И. Хренников в своей магистерской диссертации описал способ написания инструкция для человека при помощи искусственного интеллекта, взяв за основу одну из моделей Ollama [4]. G. Kosherbay рассмотрел способ написания помощника в колл-центра, а именно голосового оператора, добавив ему искусственный интеллект для ответа на вопросы и более обширного словарного запаса и правильности речи [5].

1.3 Цель исследования

Целью данной работы является создание чат-бота, способного работать локально с использованием искусственного интеллекта через платформу Ollama.

2 Описание разработки чата

Для начала необходимо перейти на сайт <https://ollama.com/download/windows> и скачать программу, которая будет в проекте выступать API-сервером (рис. 1).

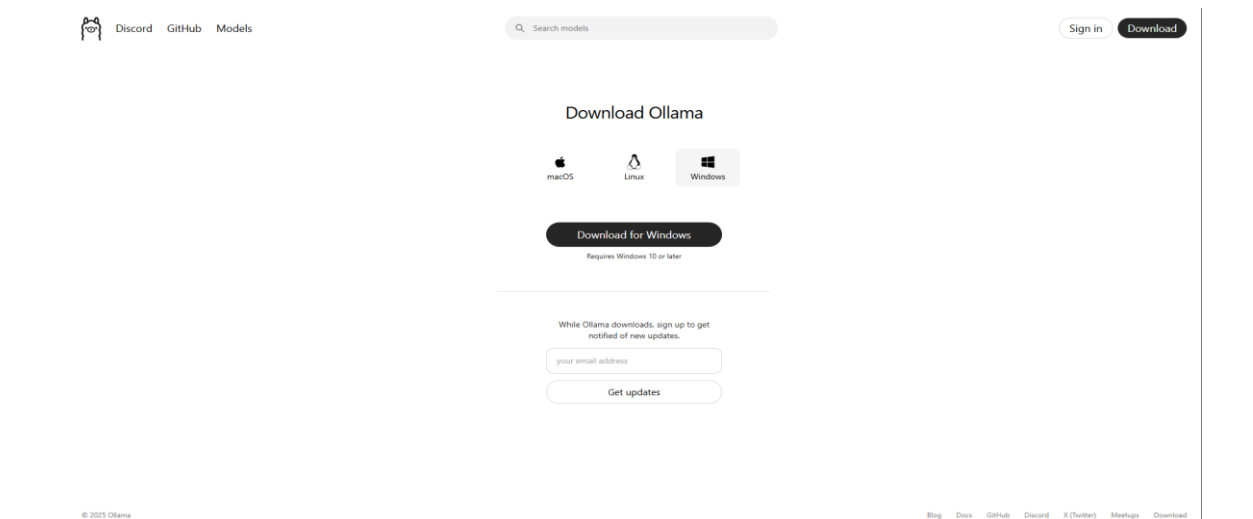


Рисунок 1 –Страница загрузки на сайте ollama.com

Далее необходимо установить приложение (рис. 2).

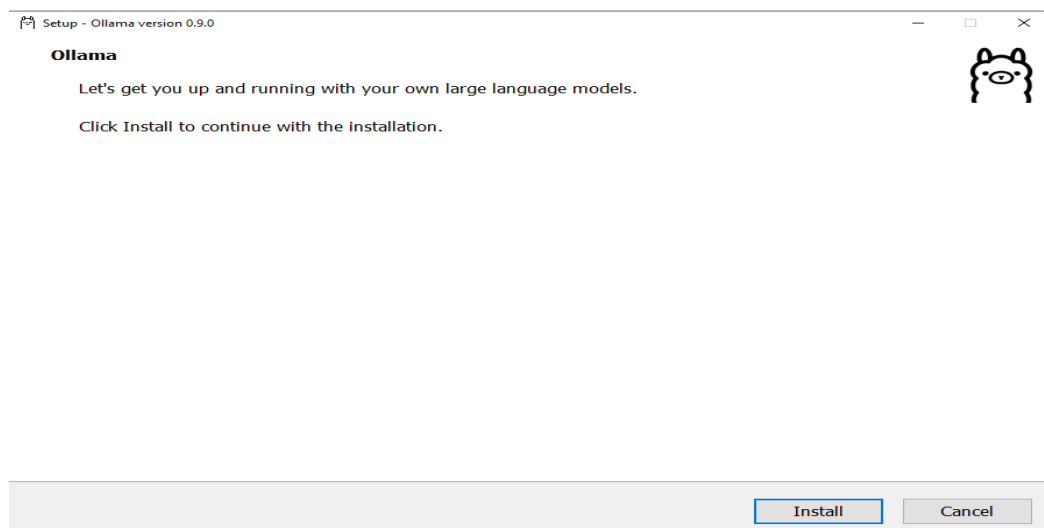


Рисунок 2 – Установка приложения Ollama

После установки, необходимо перейти в папку localhost приложения openserver panel, и создать в корне папке следующие файлы: index.html, chat.php. Также следует создать папку assets, в которой нужно создать файл style.css (рис. 3).

Имя	Дата изменения	Тип	Размер
assets	08.06.2025 0:05	Папка с файлами	
chat.php	08.06.2025 0:05	PHPfile	0 КБ
index.html	08.06.2025 0:04	Opera GX Web Do...	0 КБ

Рисунок 3 – Структура файлов в папке localhost.

Теперь настало время написать код для файла index.html. В данном коде необходимо расписать внешний вид сайта (заголовки, названия, поле ввода и т.д.) (рис. 4).

```

1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4 <meta charset="UTF-8">
5 <title>Чат-бот</title>
6 <link rel="stylesheet" href="assets/style.css">
7 </head>
8 <body>
9 <div id="chat">
10 <h2>Чат-бот, но с искусственным интеллектом</h2>
11 <div id="messages"></div>
12 <form id="chat-form">
13 <input type="text" id="message" placeholder="Введите сообщение... required">
14 <button type="submit">Отправить</button>
15 </form>
16 </div>
17
18 <script>
19 const form = document.getElementById('chat-form');
20 const input = document.getElementById('message');
21 const messages = document.getElementById('messages');
22
23 form.addEventListener('submit', async (e) => {
24   e.preventDefault();
25   const userMessage = input.value.trim();
26   if (!userMessage) return;
27   messages.innerHTML += `<div class="message user"><strong>Вы:</strong> ${userMessage}</div>`;
28   input.value = '';
29
30   const response = await fetch('chat.php', {
31     method: 'POST',
32     headers: {'Content-Type': 'application/x-www-form-urlencoded'},
33     body: `message=${encodeURIComponent(userMessage)}`
34   });
35
36   const data = await response.json();
37   const botMessage = data.message?.content || 'Ошибка получения ответа';
38   messages.innerHTML += `<div class="message bot"><strong>Бот:</strong> ${botMessage}</div>`;
39 });
40 </script>
41 </body>
42 </html>

```

Рисунок 4 – Код для файла index.html

После, следует перейти к файлу chat.php, который будет отвечать за серверную часть, а именно обработку запросов и подключения к API по localhost (рис. 5).

```

1 <?php
2 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
3   $userInput = $_POST['message'] ?? '';
4
5   $payload = [
6     'model' => 'mistral',
7     'messages' => [
8       ['role' => 'user', 'content' => $userInput]
9     ],
10    'stream' => false
11  ];
12
13  $ch = curl_init('http://127.0.0.1:11434/api/chat');
14  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
15  curl_setopt($ch, CURLOPT_HTTPHEADER, ['Content-Type: application/json']);
16  curl_setopt($ch, CURLOPT_POST, true);
17  curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
18
19  $result = curl_exec($ch);
20
21  if ($result === false) {
22    echo json_encode(['message' => ['content' => 'Ошибка запроса: ' . curl_error($ch)]]);
23  } else {
24    echo $result;
25  }
26
27  curl_close($ch);
28  exit;
29 }
30
31

```

Рисунок 5 – Код файла chat.php

На данном этапе по-своему усмотрению можно добавить немного стилей сайту в файле style.css, стили коснутся в основном поле ввода сообщения и текста (рис. 6).

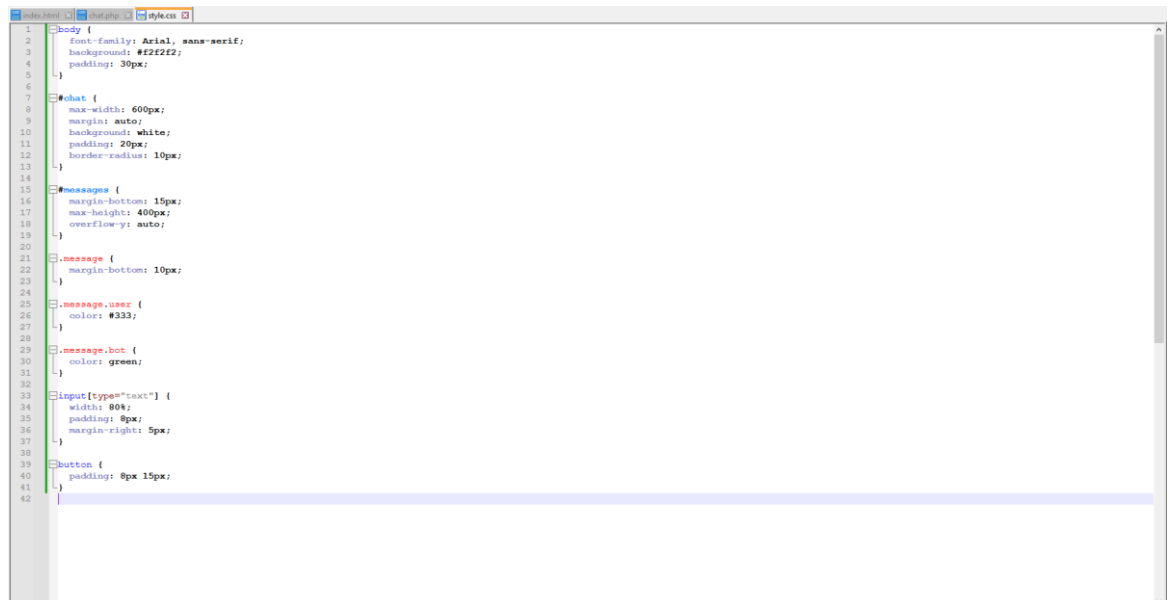


Рисунок 6 – Код файла style.css

Теперь проект необходимо запустить. Сначала нужно запустить openserver panel.

Следующим шагом необходимо выбрать модель, которая будет генерировать ответ на запрос пользователя, поэтому рекомендовано попробовать модель deepseek-coder:6.7b-instruct. Она является не очень требовательной (8-16 гб оперативной памяти)

Для того чтобы загрузить данную модель нужно зайти в командную строку и ввести: `ollama pull deepseek-coder:6.7b-instruct`. После чего начнется загрузка модели (рис. 7).

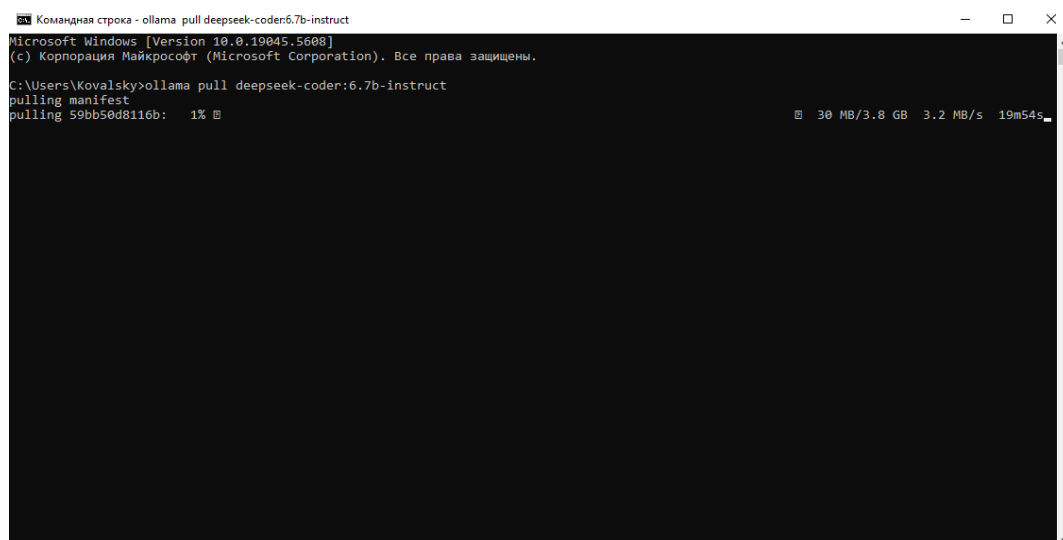


Рисунок 7 – Загрузка модели в cmd

Для данной модели потребуется примерно 16 гб оперативной памяти.

После того, как загрузится модель, ее необходимо запустить следующей командой: `ollama run deepseek-coder:6.7b-instruct` (рис. 8).

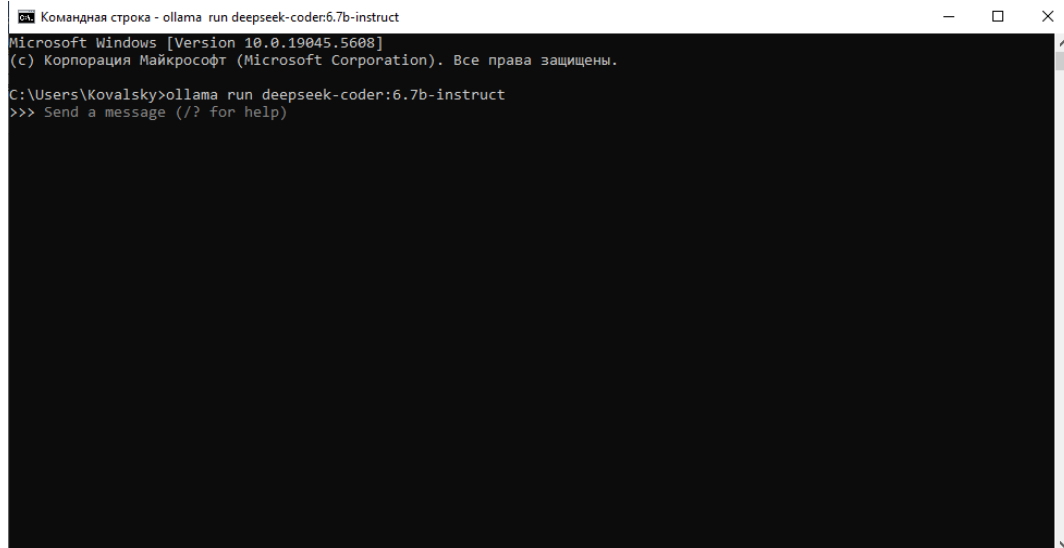


Рисунок 8 – Запуск модели в cmd

После запуска модели необходимо запустить браузер и ввести localhost (рис. 9).



Рисунок 9 – запуск локальной WEB-страницы

Главная страница содержит поле ввода, в которой пользователь вводит вопрос (рис. 10).

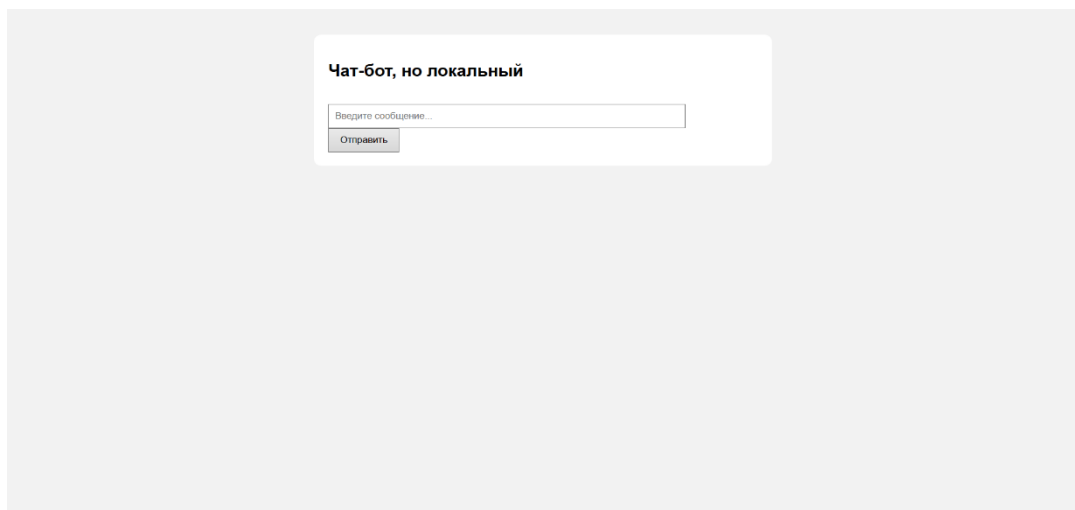


Рисунок 10 – Главная страница сайта с чат-ботом

Теперь боту можно задать любой вопрос (рис. 11).

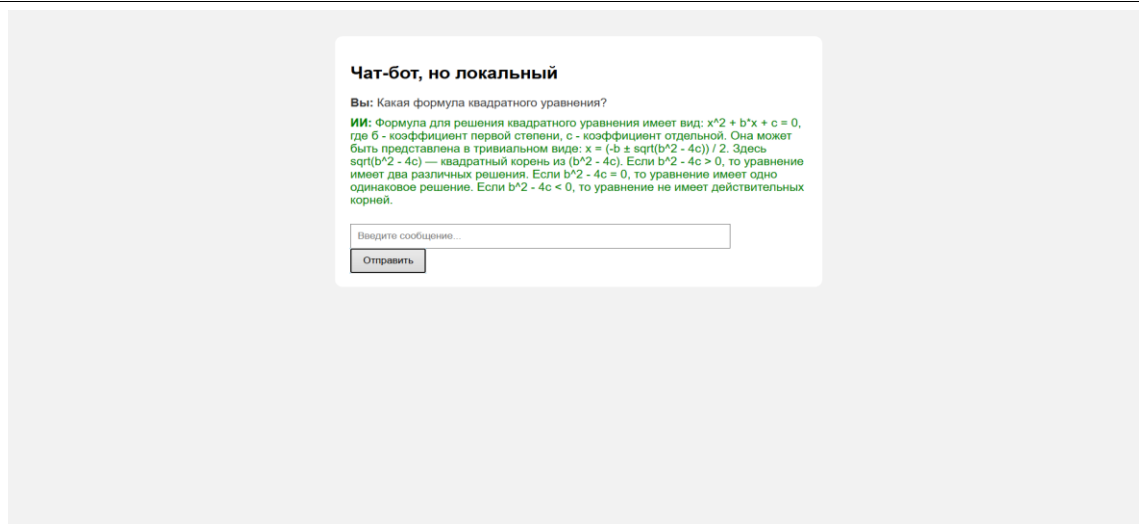


Рисунок 11 – Пример вопроса и ответ чат-бота

3 Выводы

В ходе работы удалось продемонстрировать, что использование платформы Ollama позволяет эффективно реализовать локального чат-бота с поддержкой искусственного интеллекта без зависимости от внешних облачных сервисов. Такой подход не только повышает уровень безопасности и конфиденциальности данных, но и дает больше контроля над самим процессом взаимодействия с моделью.

Несмотря на некоторые технические сложности при настройке, локальные решения открывают новые возможности для самостоятельной разработки интеллектуальных систем. Полученные результаты подтверждают, что автономный ИИ может успешно применяться в практических задачах и является достойной альтернативой онлайн-сервисам.

Библиографический список

1. Костин К. А., Бурукина И. П. Конфиденциальный «умный дом» // Вестник Пензенского государственного университета. 2025. №. 2 (50). С. 72-74.
2. Снимщиков С. В., Саврасов И. П., Сумароков Е. В. Информационная модель алгоритма применения машинопонимаемых нормативных документов при проектировании строительных конструкций //Бетон и железобетон. 2025. Т. 627. №. 2. С. 16-26.
3. Копылов Е. А., Дзюба Е. А. Личные нейроконсультанты с применением больших языковых моделей // Вестник Пензенского государственного университета. 2025. №. 1 (49). С. 66-72.
4. Хренников А. И. Применение системы LLM-агентов для решения задач, требующих рассуждений: магистерская диссертация: дис. – 2024.
5. Kosherbay G., Arbaz N. AI-Based IVR // arXiv preprint arXiv:2408.10549. 2024.