PROMPT NO CHATGPT

Quero que você atue como engenheiro de software sênior full stack e arquiteto de sistemas, especializado em Python, Flask, SQLite, APIs de IA (como Gemini) e desenvolvimento web moderno (HTML, TailwindCSS, JavaScript, UX/UI). Seu papel será me ajudar a construir um sistema completo de recomendação de florais de Bach, onde o nome do sistema será "FloralBot AI", com interface web, backend funcional, banco de dados e integração com IA. O projeto segue bons padrões técnicos com ênfase em simplicidade, empatia visual e boa arquitetura de software.

Contexto e Objetivo: O sistema tem como propósito criar um chatbot integrado via API Gemini de sugestões automáticas de florais de Bach (ou puxar do banco de dados dos florais), onde ajudará os usuários comuns a obter os florais mais adequados a partir de seus sintomas emocionais. Conforme o usuário vai descrevendo o que está sentindo, o chatbot que estará integrado a uma API irá interagindo com o usuário, fazendo uma ficha de anamnese para no final sugerir o melhor floral com base nos sintomas apresentados. Lembrando que esse chatbot não substitui um profissional da saúde.

Requisitos do Projeto

Crie um README.md sobre o projeto para incluir no GitHub.

Aplicação de Machine Learning e Deep Learning - Treinar pelo menos um modelo de aprendizado de máquina ou rede neural para resolver o problema escolhido. Implementar métricas de avaliação (ex: acurácia, precisão, recall, F1-score) com base nas conversas dos usuários com o chatbot para avaliar quantas vezes o chatbot acertou nas sugestões dos florais para com os usuários.

Natural Language Processing (NLP) - Implementar análise de sentimentos no chatbot.

AI-Driven Software Engineering - Documentar o projeto aplicando boas práticas de engenharia de software com apoio de ferramentas de IA.

Conceitos e Critérios Abordados
- Artificial Intelligence Fundamentals: fundamentos, tipos de IA e casos de uso.
- Machine Learning e Deep Learning: construção, treinamento e avaliação de modelos de ML/DL.
- Gen AI and Prompt Engineering: aplicação prática de IA generativa e design de prompts eficazes.
- Natural Language Processing (NLP): processamento e análise de linguagem natural.
- AI-Driven Software Engineering: integração de ferramentas de IA no ciclo de desenvolvimento e documentação.

Requisitos Técnicos do Sistema

1. Estrutura geral

O sistema deve conter:

Frontend web: interface moderna e intuitiva com HTML, TailwindCSS e JavaScript.
Backend: API REST em Python com Flask.
Banco de dados: SQLite integrado via SQLAlchemy.
Autenticação: sistema de login e senha com Flask-Login e Bcrypt.
Chatbot: integrado via API Gemini, com comunicação via endpoint /api/chatbot.
Design UX/UI: cores suaves (verde e branco), tipografia moderna e ícones relacionados à natureza e bem-estar.
Deploy local ou remoto.

2. Funcionalidades obrigatórias

Módulo	Funcionalidade	Descrição resumida
Usuários	Cadastro e login	Cria sua própria conta no FloralBot, com senha autenticada e criptografada.
Florais	Cadastro	Registrar nome dos florais, descrição e indicação.
Chatbot IA	Acolhimento e sugestão	Permite conversas em linguagem natural e sugere florais com base em emoções.
Interface Web	Layout moderno e responsivo	Páginas com cores suaves, ícones naturais, tipografia leve e menus intuitivos.
Usuário Administrativo	Cadastro e login	Cadastrado via terminal. O mesmo terá acesso a um dashboard com informações dos usuários cadastrados, conversas do chatbot e florais. O administrador pode adicionar, editar e excluir usuários e florais.

3. Fluxo de navegação

Tela de navegação: Tela Home com uma breve descrição, onde apresentará ao lado uma telinha para clicar em fazer login, criar conta e permanecer deslogado.
Tela Chatbot: Interação do usuário com o chatbot. Interface conversacional integrada à API Gemini — ex.: “Estou ansioso e sem foco.”
O chatbot responde, fazendo perguntas de anamnese e em seguida diz: “Os florais recomendados são Clematis, Impatiens e White Chestnut. Gostaria de saber mais sobre eles?”
Tela Florais: Apresentação dos 38 florais de Bach com descrição e indicação.
Botão sair: desloga o usuário, voltando para a tela inicial.
Cadastro: Usuário se cadastra para ter acesso ao FloralBot logado. (Campos: Nome, Data de Nascimento, Gênero (Feminino, Masculino, Outro, Prefiro não dizer), Email, Senha e Confirmar senha).
Login: O usuário acessa a conta com e-mail e senha.
Administrador: Faz login com e-mail e senha e tem acesso ao dashboard.
Dashboard: Mostra os usuários cadastrados, últimos chats e os florais cadastrados. O administrador pode adicionar, editar e excluir usuários e florais.

4. Modelo de Banco de Dados

Tabela Campos principais
user: id, name, email, password_hash, created_at, birthdate, gender, is_admin (1 para sim e 0 para não)
florais: id, name, descriptions, indications
chat_history: id, user_id, user_message, bot_message, created_at

5. Estrutura de Pastas Recomendada

```
floralbot_ai/
│
├── app/
│   ├── auth/
│   ├── static/
│   │   ├── css/
│   │   ├── js/
│   │   └── img/
│   ├── templates/
│   ├── models.py
│   ├── routes.py
│   ├── chatbot.py
│   ├── utils.py
│   └── ml_pipeline.py
│
├── migrations/
├── database/
├── seeds/
├── tests/
│
├── run.py
├── requirements.txt
├── README.md
└── tailwind.config.js
```

6. Backend (Flask) – Rotas esperadas

Rota	Método	Função
/login	GET/POST	Login de usuários e administrador
/logout	GET	Encerrar sessão
/register	POST	Criar novo usuário
/api/user	GET/POST/DELETE	CRUD de usuários
/api/florais	GET/POST/DELETE	CRUD de florais
/api/chatbot	POST	Integração com API Gemini
/	GET	Página inicial com boas-vindas e atalhos

7. Frontend e UX/UI

Layout esperado:

Página inicial:

“Desperte seu
Equilíbrio Emocional.

O FloralBot AI é seu assistente virtual para o bem-estar. Através de uma conversa amigável, ele ajuda você a identificar suas emoções e sugere as essências florais de Bach mais indicadas para harmonizar seu estado de espírito.

Um Sistema de Cura Natural
A Terapia Floral de Bach é um sistema de cura natural, desenvolvido pelo Dr. Edward Bach. Baseia-se no uso de 38 essências extraídas de flores silvestres para equilibrar as emoções.

Equilíbrio para Suas Emoções
Dr. Bach acreditava que as emoções em desordem podiam causar doenças. As essências florais ajudam a harmonizar nosso interior, transformando sentimentos ruins em bons.”

“FloralBot AI"

"Sua jornada de bem-estar começa aqui”autoconhecimento e equilíbrio emocional com a ajuda dos florais de Bach começa aqui.

Botões: Home | Chat | Florais | Entrar | Login | Criar Conta | Permanecer desconectado

Cores: tons de degradê de verde e branco.

Tipografia: “Poppins” (título), “Open Sans” (texto).

Ícones: Folhas.

Feedback visual: alertas positivos com frases empáticas como:

“Florais adicionados com sucesso ”
“Usuário cadastrado!”

8. Chatbot com API Gemini

Objetivo:
Permitir que o usuário converse com uma IA empática sobre seu estado emocional e receba sugestões de florais. A IA deve começar a frase com: "Olá! Sou o FloralBot. Como você está se sentindo hoje?"

Exemplo de fluxo:

Usuário: “Estou triste e com falta de energia.”

Chatbot: “Sinto muito por isso. Florais como Mustard, Olive e Gentian podem ajudar a restaurar o equilíbrio emocional.”

Endpoint: /api/chatbot

API: Google Gemini

Mecanismo: Flask recebe POST → envia requisição à API → retorna resposta JSON

Registra logs no banco (chat_history)

9. Segurança e autenticação

Login obrigatório para acesso.

Senhas criptografadas com Flask-Bcrypt.

Sessões gerenciadas com Flask-Login.

Prevenção de injeção SQL (ORM SQLAlchemy).

CORS habilitado para integração com front-end.

10. Entregáveis esperados

Código completo em Python/Flask + HTML/TailwindCSS/JS

Banco SQLite funcional

Integração com API Gemini

Página inicial intuitiva e bonita

CRUD completo de florais e usuários

Autenticação funcional

12. Tecnologias sugeridas

Backend	Python, Flask, SQLAlchemy
Frontend	HTML, TailwindCSS, JS, Bootstrap
Banco	SQLite
Chatbot	API Gemini (Google)

Autenticação Flask-Login, Bcrypt
Hospedagem Localhost e/ou Render

13. Resultado esperado

Um sistema web funcional, empático e útil, onde:

Os usuários podem se cadastrar incluindo "Nome Completo", "Data de Nascimento" (o sistema apresentará a idade de forma automática ao incluir a data de nascimento) (utilize o calendário gregoriano), Gênero (inclua "Feminino", "Masculino", “Outro”, "Prefiro dizer", "E-mail", "Senha" e “Confirmar senha”;

O sistema recomenda florais de Bach automaticamente com base em sintomas após anamnese feita pelo próprio chatbot;

Usuários podem conversar com uma IA amigável para receber suporte emocional e orientações de florais;

Tudo funciona com segurança, boa aparência e performance estável.

Instrução final

Gere o projeto completo, com todos os arquivos base (estrutura Flask, templates HTML, TailwindCSS, JS, banco SQLite, endpoints REST e integração com Gemini), seguindo o padrão de boas práticas de desenvolvimento e documentação.

Use nomes de variáveis e classes em português e siga a convenção PEP8.

O resultado deve ser um sistema simples de instalar, rodar e demonstrar.

OBSERVAÇÕES:

Quero que você ajuste tudo o que precisar, com base nessas informações. Lembrando também que vou usar o VS CODE para rodar os códigos. Me dê o passo a passo do que fazer e como fazer, como criar as pastas dentro do VS CODE e o que eu vou rodar no terminal e o que vou jogar em cada pasta. Preciso de tudo muito bem explicado, quais são as extensões que eu preciso instalar no VS CODE para usar o SQLITE.