**💡 Project: AI Email Assistant & Auto-Responder**

Users paste incoming emails → AI generates suggested replies, summarizes threads, and tags intent (e.g. complaint, inquiry, task).

**🔧 STEP 1: Set Up the Project Structure**

**1. Create Project & Backend Folder**

mkdir EmailAI && cd EmailAI

mkdir backend && cd backend

python -m venv venv

PS C:\Users\pc\OneDrive\Desktop\EmailAI> mkdir backend

Directory: C:\Users\pc\OneDrive\Desktop\EmailAI

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 6/23/2025 1:09 PM backend

PS C:\Users\pc\OneDrive\Desktop\EmailAI> cd backend

PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> python -m venv venv

PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend>

**2. Activate the virtual environment:**

* **Windows**

venv\Scripts\activate

PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> venv\Scripts\activate

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend>

**STEP 2: Create requirements.txt**

Create backend/requirements.txt and paste this:

fastapi

uvicorn

python-multipart

transformers

torch

spacy

nltk

scikit-learn

textblob

fastapi-cors

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> pip install -r requirements.txt

Requirement already satisfied: fastapi in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 1)) (0.115.13)

Requirement already satisfied: uvicorn in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 2)) (0.34.3)

Requirement already satisfied: python-multipart in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 3)) (0.0.20)

Requirement already satisfied: transformers in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 5)) (4.52.4)

Requirement already satisfied: torch in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 6)) (2.7.1)

Requirement already satisfied: spacy in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 7)) (3.8.7)

Requirement already satisfied: nltk in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 8)) (3.9.1)

Requirement already satisfied: scikit-learn in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 9)) (1.7.0)

Requirement already satisfied: textblob in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from -r requirements.txt (line 10)) (0.19.0)

Requirement already satisfied: starlette<0.47.0,>=0.40.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from fastapi->-r requirements.txt (line 1)) (0.46.2)

Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from fastapi->-r requirements.txt (line 1)) (2.11.7)

Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from fastapi->-r requirements.txt (line 1)) (4.14.0)

Requirement already satisfied: click>=7.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from uvicorn->-r requirements.txt (line 2)) (8.2.1)

Requirement already satisfied: h11>=0.8 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from uvicorn->-r requirements.txt (line 2)) (0.16.0)

Requirement already satisfied: filelock in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (3.18.0)

Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (0.33.0)

Requirement already satisfied: numpy>=1.17 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (2.3.1)

Requirement already satisfied: packaging>=20.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (25.0)

Requirement already satisfied: pyyaml>=5.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (6.0.2)

Requirement already satisfied: regex!=2019.12.17 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (2024.11.6)

Requirement already satisfied: requests in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (2.32.4)

Requirement already satisfied: tokenizers<0.22,>=0.21 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (0.21.1)

Requirement already satisfied: safetensors>=0.4.3 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (0.5.3)

Requirement already satisfied: tqdm>=4.27 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from transformers->-r requirements.txt (line 5)) (4.67.1)

Requirement already satisfied: sympy>=1.13.3 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from torch->-r requirements.txt (line 6)) (1.14.0)

Requirement already satisfied: networkx in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from torch->-r requirements.txt (line 6)) (3.5)

Requirement already satisfied: jinja2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from torch->-r requirements.txt (line 6)) (3.1.6)

Requirement already satisfied: fsspec in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from torch->-r requirements.txt (line 6)) (2025.5.1)

Requirement already satisfied: setuptools in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from torch->-r requirements.txt (line 6)) (80.9.0)

Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (3.0.12)

Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (1.0.5)

Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (1.0.13)

Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (2.0.11)

Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (3.0.10)

Requirement already satisfied: thinc<8.4.0,>=8.3.4 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (8.3.6)

Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (1.1.3)

Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (2.5.1)

Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (2.0.10)

Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (0.4.1)

Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (0.16.0)

Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from spacy->-r requirements.txt (line 7)) (3.5.0)

Requirement already satisfied: joblib in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from nltk->-r requirements.txt (line 8)) (1.5.1)

Requirement already satisfied: scipy>=1.8.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from scikit-learn->-r requirements.txt (line 9)) (1.16.0)

Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from scikit-learn->-r requirements.txt (line 9)) (3.6.0)

Requirement already satisfied: colorama in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from click>=7.0->uvicorn->-r requirements.txt (line 2)) (0.4.6)

Requirement already satisfied: language-data>=1.2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from langcodes<4.0.0,>=3.2.0->spacy->-r requirements.txt (line 7)) (1.3.0)

Requirement already satisfied: annotated-types>=0.6.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (0.7.0)

Requirement already satisfied: pydantic-core==2.33.2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (2.33.2)

Requirement already satisfied: typing-inspection>=0.4.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1)) (0.4.1)

Requirement already satisfied: charset\_normalizer<4,>=2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from requests->transformers->-r requirements.txt (line 5)) (3.4.2)

Requirement already satisfied: idna<4,>=2.5 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from requests->transformers->-r requirements.txt (line 5)) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from requests->transformers->-r requirements.txt (line 5)) (2.5.0)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from requests->transformers->-r requirements.txt (line 5)) (2025.6.15)

Requirement already satisfied: anyio<5,>=3.6.2 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from starlette<0.47.0,>=0.40.0->fastapi->-r requirements.txt (line 1)) (4.9.0)

Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from sympy>=1.13.3->torch->-r requirements.txt (line 6)) (1.3.0)

Requirement already satisfied: blis<1.4.0,>=1.3.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy->-r requirements.txt (line 7)) (1.3.0)

Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy->-r requirements.txt (line 7)) (0.1.5)

Requirement already satisfied: shellingham>=1.3.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy->-r requirements.txt (line 7)) (1.5.4)

Requirement already satisfied: rich>=10.11.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy->-r requirements.txt (line 7)) (14.0.0)

Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy->-r requirements.txt (line 7)) (0.21.1)

Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy->-r requirements.txt (line 7)) (7.1.0)

Requirement already satisfied: MarkupSafe>=2.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from jinja2->torch->-r requirements.txt (line 6)) (3.0.2)

Requirement already satisfied: sniffio>=1.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from anyio<5,>=3.6.2->starlette<0.47.0,>=0.40.0->fastapi->-r requirements.txt (line 1)) (1.3.1)

Requirement already satisfied: marisa-trie>=1.1.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy->-r requirements.txt (line 7)) (1.2.1)

Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy->-r requirements.txt (line 7)) (3.0.0)

Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy->-r requirements.txt (line 7)) (2.19.2)

Requirement already satisfied: wrapt in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy->-r requirements.txt (line 7)) (1.17.2)

Requirement already satisfied: mdurl~=0.1 in c:\users\pc\onedrive\desktop\emailai\backend\venv\lib\site-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy->-r requirements.txt (line 7)) (0.1.2)

[notice] A new release of pip is available: 25.0.1 -> 25.1.1

[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend>

**📁 STEP 3: Folder & File Structure**

Inside backend/, run:

mkdir data

touch main.py summarizer.py intent\_classifier.py reply\_generator.py utils.py

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> mkdir data

Directory: C:\Users\pc\OneDrive\Desktop\EmailAI\backend

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 6/23/2025 1:23 PM data

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend>

EmailAI/

└── backend/

├── main.py ← FastAPI app & API routes

├── summarizer.py ← Email summarization logic

├── intent\_classifier.py ← Intent detection logic

├── reply\_generator.py ← Smart reply generator

├── utils.py ← Basic text cleaning

└── data/ ← Store labeled intent examples (optional)

All files are made.

**🧠 STEP 4: Download spaCy Model**

python -m spacy download en\_core\_web\_sm

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> python -m spacy download en\_core\_web\_sm

Collecting en-core-web-sm==3.8.0

Downloading https://github.com/explosion/spacy-models/releases/download/en\_core\_web\_sm-3.8.0/en\_core\_web\_sm-3.8.0-py3-none-any.whl (12.8 MB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 1.7 MB/s eta 0:00:00

Installing collected packages: en-core-web-sm

Successfully installed en-core-web-sm-3.8.0

[notice] A new release of pip is available: 25.0.1 -> 25.1.1

[notice] To update, run: python.exe -m pip install --upgrade pip

✔ Download and installation successful

You can now load the package via spacy.load('en\_core\_web\_sm')

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend>

backend\intent\_classifier.py

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

# Expanded example dataset

training\_data = [

    ("Can you send me the contract?", "request"),

    ("Please schedule the meeting.", "task"),

    ("We're unhappy with the service.", "complaint"),

    ("How much does this cost?", "inquiry"),

    ("Thanks for your help!", "gratitude"),

    ("When will this be delivered?", "follow-up"),

    ("Here's the report you asked for.", "response"),

]

X\_texts, y\_labels = zip(\*training\_data)

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(X\_texts)

classifier = LogisticRegression()

classifier.fit(X, y\_labels)

def classify\_intent(text: str) -> str:

    try:

        vec = vectorizer.transform([text])

        pred = classifier.predict(vec)

        return pred[0]

    except:

        return "unknown"

backend\main.py

from fastapi import FastAPI

from fastapi.middleware.cors import CORSMiddleware  # Add this import

from pydantic import BaseModel

from summarizer import summarize\_email

from intent\_classifier import classify\_intent

from reply\_generator import generate\_reply

from utils import clean\_text

app = FastAPI(title="AI Email Assistant")

# Add CORS middleware

app.add\_middleware(

    CORSMiddleware,

    allow\_origins=["\*"],  # Allows all origins (for development only)

    allow\_credentials=True,

    allow\_methods=["\*"],  # Allows all methods

    allow\_headers=["\*"],  # Allows all headers

)

class EmailInput(BaseModel):

    email\_text: str

    tone: str = "formal"  # formal, friendly, short

@app.post("/summarize/")

def summarize(payload: EmailInput):

    text = clean\_text(payload.email\_text)

    if len(text) < 20:

        return {"summary": "Email too short to summarize."}

    return {"summary": summarize\_email(text)}

@app.post("/classify/")

def classify(payload: EmailInput):

    text = clean\_text(payload.email\_text)

    return {"intent": classify\_intent(text)}

@app.post("/generate\_reply/")

def reply(payload: EmailInput):

    text = clean\_text(payload.email\_text)

    return {"reply": generate\_reply(text, tone=payload.tone)}

backend\reply\_generator.py

from transformers import pipeline

generator = pipeline("text-generation", model="gpt2")

def generate\_reply(email\_text: str, tone: str = "formal") -> str:

    prompt\_style = {

        "formal": "Write a formal reply to this email:",

        "friendly": "Reply in a friendly and warm tone:",

        "short": "Give a short and to-the-point response:",

    }

    base\_prompt = prompt\_style.get(tone.lower(), prompt\_style["formal"])

    prompt = f"{base\_prompt}\n\nEmail: {email\_text}\n\nReply:"

    result = generator(prompt, max\_length=150, num\_return\_sequences=1, temperature=0.7)[0]["generated\_text"]

    return result.split("Reply:")[-1].strip()

backend\utils.py

import re

def clean\_text(text: str) -> str:

    # Remove common signatures and excess whitespace

    text = re.sub(r"\s+", " ", text)

    text = re.sub(r"(?i)(regards|thanks|sincerely|best|cheers)[^$]\*$", "", text)

    return text.strip()

(venv) PS C:\Users\pc\OneDrive\Desktop\EmailAI\backend> uvicorn main:app --reload

INFO: Will watch for changes in these directories: ['C:\\Users\\pc\\OneDrive\\Desktop\\EmailAI\\backend']

INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)

INFO: Started reloader process [48452] using StatReload

Device set to use cpu

Device set to use cpu

INFO: Started server process [29752]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: 127.0.0.1:54656 - "GET /docs HTTP/1.1" 200 OK

INFO: 127.0.0.1:54656 - "GET /openapi.json HTTP/1.1" 200 OK

The backend is working now correctly.

**✅ STEP 1: Create Frontend Project**

From inside your EmailAI folder:

npx create-react-app frontend

cd frontend

npm install axios

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

PS C:\Users\pc\OneDrive\Desktop\EmailAI> npx create-react-app frontend

Creating a new React app in C:\Users\pc\OneDrive\Desktop\EmailAI\frontend.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 2m

269 packages are looking for funding

run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 18 packages, and changed 1 package in 12s

269 packages are looking for funding

run `npm fund` for details

Removing template package using npm...

removed 1 package, and audited 1342 packages in 8s

269 packages are looking for funding

run `npm fund` for details

9 vulnerabilities (3 moderate, 6 high)

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

Created git commit.

Success! Created frontend at C:\Users\pc\OneDrive\Desktop\EmailAI\frontend

Inside that directory, you can run several commands:

npm start

Starts the development server.

npm run build

Bundles the app into static files for production.

npm test

Starts the test runner.

npm run eject

Removes this tool and copies build dependencies, configuration files

and scripts into the app directory. If you do this, you can’t go back!

We suggest that you begin by typing:

cd frontend

npm start

Happy hacking!

PS C:\Users\pc\OneDrive\Desktop\EmailAI> cd frontend

PS C:\Users\pc\OneDrive\Desktop\EmailAI\frontend> npm install axios

added 3 packages, and audited 1345 packages in 5s

269 packages are looking for funding

run `npm fund` for details

9 vulnerabilities (3 moderate, 6 high)

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

PS C:\Users\pc\OneDrive\Desktop\EmailAI\frontend> npm install -D tailwindcss@3.4.17 postcss@8.5.4 autoprefixer

changed 1 package, and audited 1345 packages in 5s

269 packages are looking for funding

run `npm fund` for details

9 vulnerabilities (3 moderate, 6 high)

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

PS C:\Users\pc\OneDrive\Desktop\EmailAI\frontend> npx tailwindcss init -p

Created Tailwind CSS config file: tailwind.config.js

Created PostCSS config file: postcss.config.js

PS C:\Users\pc\OneDrive\Desktop\EmailAI\frontend>

frontend/

├── node\_modules/ # Installed dependencies (auto-generated)

├── public/ # Static assets (HTML, images, etc.)

├── src/ # Source code

│ ├── App.css # Component-specific styles (optional with Tailwind)

│ ├── App.js # Main React component

│ ├── App.test.js # Unit tests for App component

│ ├── index.css # Global styles, usually includes Tailwind directives

│ ├── index.js # Entry point of React application

│ ├── logo.svg # Logo asset (used in App.js typically)

│ ├── reportWebVitals.js # Web performance reporting (optional)

│ └── setupTests.js # Test environment setup (e.g., Jest, React Testing Library)

├── .gitignore # Specifies files/folders Git should ignore

├── package-lock.json # Dependency tree lockfile (auto-managed)

├── package.json # Project metadata & dependency definitions

├── postcss.config.js # PostCSS configuration (used with Tailwind)

├── README.md # Project description & instructions

└── tailwind.config.js # Tailwind CSS custom configuration

frontend\tailwind.config.js

/\*\* @type {import('tailwindcss').Config} \*/

module.exports = {

  content: [

    "./src/\*\*/\*.{js,jsx,ts,tsx}",

  ],

  theme: {

    extend: {

      colors: {

        indigo: {

          50: '#f0f4ff',

          100: '#e0e8ff',

          200: '#c7d2fe',

          300: '#a5b4fc',

          400: '#818cf8',

          500: '#6366f1',

          600: '#4f46e5',

          700: '#4338ca',

          800: '#3730a3',

          900: '#312e81',

        },

      },

    },

  },

  plugins: [

    require('@tailwindcss/line-clamp'),

  ],

}

frontend\src\index.css

@tailwind base;

@tailwind components;

@tailwind utilities;

/\* Custom styles \*/

.prose {

  max-width: 100%;

}

.prose p {

  margin-bottom: 1rem;

  line-height: 1.6;

}

.line-clamp-2 {

  display: -webkit-box;

  -webkit-line-clamp: 2;

  -webkit-box-orient: vertical;

  overflow: hidden;

}

/\* Animation for the copy button \*/

@keyframes fadeIn {

  from { opacity: 0; }

  to { opacity: 1; }

}

.animate-fade-in {

  animation: fadeIn 0.3s ease-in-out;

}

frontend\src\App.js

import React, { useState, useEffect } from 'react';

import axios from 'axios';

import { FiMail, FiSend, FiRefreshCw, FiCopy, FiCheck } from 'react-icons/fi';

function App() {

  const [emailText, setEmailText] = useState('');

  const [tone, setTone] = useState('formal');

  const [summary, setSummary] = useState('');

  const [intent, setIntent] = useState('');

  const [reply, setReply] = useState('');

  const [isLoading, setIsLoading] = useState(false);

  const [copied, setCopied] = useState(false);

  const [history, setHistory] = useState([]);

  const [activeTab, setActiveTab] = useState('compose');

  const API\_URL = 'http://localhost:8000';

  const processEmail = async () => {

    if (!emailText.trim()) return;

    setIsLoading(true);

    try {

      // Get all responses in parallel

      const [summaryRes, intentRes, replyRes] = await Promise.all([

        axios.post(`${API\_URL}/summarize/`, { email\_text: emailText }),

        axios.post(`${API\_URL}/classify/`, { email\_text: emailText }),

        axios.post(`${API\_URL}/generate\_reply/`, { email\_text: emailText, tone })

      ]);

      setSummary(summaryRes.data.summary);

      setIntent(intentRes.data.intent);

      setReply(replyRes.data.reply);

      // Add to history

      setHistory(prev => [{

        id: Date.now(),

        email: emailText,

        summary: summaryRes.data.summary,

        intent: intentRes.data.intent,

        reply: replyRes.data.reply,

        timestamp: new Date().toISOString()

      }, ...prev.slice(0, 9)]);

    } catch (error) {

      console.error('Error processing email:', error);

    } finally {

      setIsLoading(false);

    }

  };

  const copyToClipboard = () => {

    navigator.clipboard.writeText(reply);

    setCopied(true);

    setTimeout(() => setCopied(false), 2000);

  };

  const loadFromHistory = (item) => {

    setEmailText(item.email);

    setSummary(item.summary);

    setIntent(item.intent);

    setReply(item.reply);

  };

  const clearAll = () => {

    setEmailText('');

    setSummary('');

    setIntent('');

    setReply('');

  };

  return (

    <div className="min-h-screen bg-gray-50 text-gray-900">

      <header className="bg-indigo-600 text-white p-4 shadow-md">

        <div className="container mx-auto flex justify-between items-center">

          <h1 className="text-2xl font-bold flex items-center">

            <FiMail className="mr-2" /> AI Email Assistant

          </h1>

          <div className="flex space-x-4">

            <button

              onClick={() => setActiveTab('compose')}

              className={`px-4 py-2 rounded-lg ${activeTab === 'compose' ? 'bg-indigo-800' : 'bg-indigo-700 hover:bg-indigo-800'}`}

            >

              Compose

            </button>

            <button

              onClick={() => setActiveTab('history')}

              className={`px-4 py-2 rounded-lg ${activeTab === 'history' ? 'bg-indigo-800' : 'bg-indigo-700 hover:bg-indigo-800'}`}

            >

              History

            </button>

          </div>

        </div>

      </header>

      <main className="container mx-auto p-4 md:p-6">

        {activeTab === 'compose' ? (

          <div className="grid grid-cols-1 lg:grid-cols-3 gap-6">

            {/\* Email Input \*/}

            <div className="lg:col-span-2 space-y-6">

              <div className="bg-white rounded-xl shadow-md overflow-hidden">

                <div className="p-4 bg-indigo-50 border-b border-indigo-100">

                  <h2 className="text-lg font-semibold text-indigo-800">Incoming Email</h2>

                </div>

                <div className="p-4">

                  <textarea

                    className="w-full h-64 p-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-indigo-500 focus:border-indigo-500"

                    placeholder="Paste the email you received here..."

                    value={emailText}

                    onChange={(e) => setEmailText(e.target.value)}

                  />

                </div>

                <div className="p-4 bg-gray-50 border-t border-gray-200 flex justify-between items-center">

                  <div className="flex items-center space-x-4">

                    <label className="text-sm font-medium text-gray-700">Tone:</label>

                    <select

                      className="border border-gray-300 rounded-md px-3 py-1 text-sm focus:ring-indigo-500 focus:border-indigo-500"

                      value={tone}

                      onChange={(e) => setTone(e.target.value)}

                    >

                      <option value="formal">Formal</option>

                      <option value="friendly">Friendly</option>

                      <option value="short">Short</option>

                    </select>

                  </div>

                  <div className="flex space-x-2">

                    <button

                      onClick={clearAll}

                      className="px-4 py-2 bg-gray-200 text-gray-800 rounded-md hover:bg-gray-300 transition-colors"

                    >

                      Clear

                    </button>

                    <button

                      onClick={processEmail}

                      disabled={isLoading || !emailText.trim()}

                      className={`px-4 py-2 bg-indigo-600 text-white rounded-md hover:bg-indigo-700 transition-colors flex items-center ${(isLoading || !emailText.trim()) ? 'opacity-70 cursor-not-allowed' : ''}`}

                    >

                      {isLoading ? (

                        <>

                          <FiRefreshCw className="animate-spin mr-2" />

                          Processing...

                        </>

                      ) : (

                        <>

                          <FiSend className="mr-2" />

                          Analyze

                        </>

                      )}

                    </button>

                  </div>

                </div>

              </div>

              {/\* Summary & Intent \*/}

              <div className="grid grid-cols-1 md:grid-cols-2 gap-6">

                <div className="bg-white rounded-xl shadow-md overflow-hidden">

                  <div className="p-4 bg-indigo-50 border-b border-indigo-100">

                    <h2 className="text-lg font-semibold text-indigo-800">Summary</h2>

                  </div>

                  <div className="p-4">

                    {summary ? (

                      <p className="text-gray-700">{summary}</p>

                    ) : (

                      <p className="text-gray-400 italic">Summary will appear here</p>

                    )}

                  </div>

                </div>

                <div className="bg-white rounded-xl shadow-md overflow-hidden">

                  <div className="p-4 bg-indigo-50 border-b border-indigo-100">

                    <h2 className="text-lg font-semibold text-indigo-800">Intent</h2>

                  </div>

                  <div className="p-4">

                    {intent ? (

                      <span className={`inline-block px-3 py-1 rounded-full text-sm font-medium ${

                        intent === 'complaint' ? 'bg-red-100 text-red-800' :

                        intent === 'inquiry' ? 'bg-blue-100 text-blue-800' :

                        intent === 'task' ? 'bg-yellow-100 text-yellow-800' :

                        intent === 'request' ? 'bg-purple-100 text-purple-800' :

                        intent === 'gratitude' ? 'bg-green-100 text-green-800' :

                        'bg-gray-100 text-gray-800'

                      }`}>

                        {intent.charAt(0).toUpperCase() + intent.slice(1)}

                      </span>

                    ) : (

                      <p className="text-gray-400 italic">Intent will appear here</p>

                    )}

                  </div>

                </div>

              </div>

            </div>

            {/\* Generated Reply \*/}

            <div className="space-y-6">

              <div className="bg-white rounded-xl shadow-md overflow-hidden h-full flex flex-col">

                <div className="p-4 bg-indigo-50 border-b border-indigo-100 flex justify-between items-center">

                  <h2 className="text-lg font-semibold text-indigo-800">Suggested Reply</h2>

                  {reply && (

                    <button

                      onClick={copyToClipboard}

                      className="flex items-center text-sm text-indigo-600 hover:text-indigo-800"

                    >

                      {copied ? (

                        <>

                          <FiCheck className="mr-1" /> Copied!

                        </>

                      ) : (

                        <>

                          <FiCopy className="mr-1" /> Copy

                        </>

                      )}

                    </button>

                  )}

                </div>

                <div className="p-4 flex-grow">

                  {reply ? (

                    <div className="prose max-w-none text-gray-700">

                      <p>{reply}</p>

                    </div>

                  ) : (

                    <p className="text-gray-400 italic h-full flex items-center justify-center">

                      Your AI-generated reply will appear here

                    </p>

                  )}

                </div>

              </div>

            </div>

          </div>

        ) : (

          <div className="bg-white rounded-xl shadow-md overflow-hidden">

            <div className="p-4 bg-indigo-50 border-b border-indigo-100">

              <h2 className="text-lg font-semibold text-indigo-800">History</h2>

            </div>

            <div className="p-4">

              {history.length > 0 ? (

                <div className="space-y-4">

                  {history.map((item) => (

                    <div

                      key={item.id}

                      className="border border-gray-200 rounded-lg p-4 hover:bg-gray-50 cursor-pointer transition-colors"

                      onClick={() => loadFromHistory(item)}

                    >

                      <div className="flex justify-between items-start">

                        <div>

                          <span className={`inline-block px-2 py-1 rounded-full text-xs font-medium mb-2 ${

                            item.intent === 'complaint' ? 'bg-red-100 text-red-800' :

                            item.intent === 'inquiry' ? 'bg-blue-100 text-blue-800' :

                            item.intent === 'task' ? 'bg-yellow-100 text-yellow-800' :

                            item.intent === 'request' ? 'bg-purple-100 text-purple-800' :

                            item.intent === 'gratitude' ? 'bg-green-100 text-green-800' :

                            'bg-gray-100 text-gray-800'

                          }`}>

                            {item.intent.charAt(0).toUpperCase() + item.intent.slice(1)}

                          </span>

                          <p className="text-sm text-gray-500 line-clamp-2">{item.summary}</p>

                        </div>

                        <span className="text-xs text-gray-400 whitespace-nowrap">

                          {new Date(item.timestamp).toLocaleTimeString()}

                        </span>

                      </div>

                    </div>

                  ))}

                </div>

              ) : (

                <p className="text-gray-400 italic text-center py-8">

                  No history yet. Process some emails to see them here.

                </p>

              )}

            </div>

          </div>

        )}

      </main>

      <footer className="bg-white border-t border-gray-200 py-4 mt-8">

        <div className="container mx-auto px-4 text-center text-gray-500 text-sm">

          <p>AI Email Assistant - Powered by FastAPI & React</p>

        </div>

      </footer>

    </div>

  );

}

export default App;