**📄 Project Documentation – AI Summarizer**

**1. Introduction**

The **AI Summarizer** project is designed to take long text input from users and generate concise summaries. It leverages modern **frontend frameworks, backend APIs, and AI services** to deliver a seamless user experience. Additionally, users can send summaries directly to their email for future reference.

**2. Approach**

1. **Problem Identification**
   * Long texts such as articles, research papers, or emails are time-consuming to read.
   * Users need quick and accurate summaries without losing the context.
2. **Solution**
   * Build a **web-based summarization tool** that takes text input and generates AI-powered summaries.
   * Provide an option for users to **receive the summary via email** for convenience.
3. **Architecture Overview**
   * **Frontend (UI):** Users paste or type their text in a form and click "Summarize".
   * **Backend API:** The text is sent to custom API routes for summarization.
   * **AI Summarizer Engine:** Processes the text and returns a concise summary.
   * **Email Service:** If selected, the generated summary is sent to the user’s email.

**3. Process**

**Step 1: Input**

* User enters or pastes text in the input box.
* User can optionally provide an email address.

**Step 2: API Handling**

* The request is sent to **Next.js API routes**:
  + summarizeroute.ts → Handles text summarization.
  + sendemail/route.ts → Handles sending summary to email.

**Step 3: Summarization**

* The summarizer route interacts with the AI model (or an external API if configured).
* A concise summary is generated and sent back to the frontend.

**Step 4: Email Delivery**

* If the user provided an email, the backend uses the sendemail API to send the summary.

**Step 5: Output**

* The frontend displays the summarized text immediately.
* The user also receives a copy in their inbox (if email was provided).

**4. Tech Stack**

**Frontend**

* **React (with Next.js)** → Interactive UI with server-side rendering.
* **TypeScript (.tsx)** → Type safety and scalability.

**Backend**

* **Next.js API Routes** → Lightweight serverless backend to handle summarization and email.
* **TypeScript (.ts)** → Enforces type safety for backend code.

**Environment & Configuration**

* **.env.local** → Stores private keys (API keys, SMTP credentials).
* **tsconfig.json** → Configures TypeScript compiler.
* **.gitignore** → Excludes sensitive files and node\_modules from version control.

**Dependencies & Tools**

* **Node.js + npm** → Package manager and runtime.
* **AI Summarization Service** → Core engine for generating summaries.
* **SMTP / Email Service** → Sends summaries to user emails.

**5. Project Structure**

AI SUMMARIZER

│

├── app

│ └── api

│ ├── sendemail

│ │ └── route.ts

│ └── summarize

│ └── summarizeroute.ts

│

├── page.tsx

├── .env.local

├── .gitignore

├── package.json

└── tsconfig.json

**6. Future Enhancements**

* Add support for **multiple languages**.
* Implement **voice-to-summary** functionality.
* Provide **download as PDF/Docx** option for summaries.
* Add a **history panel** for users to view past summaries.