Research Project Document

📌 Phase 1: Data Collection, Cleaning & Idea Generation (Week 1)

Collecting Research Papers (Free Sources)

🗓️ **Day 1-2**
✅ **Why?**
To train your models, you need a **large dataset of research papers**.

✅ **How?**

- **Free APIs for Research Papers**:
  - arXiv API → **Best for AI/ML papers**
  - Semantic Scholar API → **Best for citations & abstracts**
  - CORE API → **Full-text papers**

✅ **Where to Store?**

- Use **MongoDB Atlas (free tier)** for NoSQL storage.
- Store structured data in **PostgreSQL (free on Render.com)**.

Cleaning & Preprocessing

🗓️ **Day 3-4**
✅ **Why?**
Cleaning helps remove **irrelevant text**, improving **model accuracy**.

✅ **How?**

- Convert text to **lowercase**
- Remove **citations** (e.g., [1], (Smith, 2020)) using **regex**
- Tokenize text into **sentences**
- Remove **stopwords** (the, is, and, etc.)

✅ **Tools (Free)**

- **NLTK** (pip install nltk)
- **spaCy** (pip install spacy)

✅ **Where to Store Clean Data?**

- Store in **PostgreSQL (hosted on free-tier Supabase)**

Idea Generation Model (Using OpenAI Free API or Local Model)

🗓️ **Day 5**
✅ **Why?**
To generate research ideas based on topics users are interested in.

✅ **How?**

- **Option 1 (Free API-based):**
  - Use **OpenAI's Free API Tier** (GPT-3.5 Turbo) → free 5M tokens per month
  - Call API with {"topic": "Artificial Intelligence"} → Get 3-5 research ideas
- **Option 2 (Local Model - Free)**
  - Install GPT4All (runs locally, no cost)
  - Train on a **small research dataset**

✅ **Where?**

- Implement in **FastAPI (/generate-idea route)**
- Store **user ideas in PostgreSQL**

📌 **Phase 2: Model Training (Week 2)**

Training Summarization Model (T5/BART)

🗓️ **Day 6-7**
✅ **Why?**
To generate **concise summaries** of research papers.

✅ **How?**

- Use **pre-trained facebook/bart-large-cnn model** (free on Hugging Face)
- Fine-tune on **arXiv abstracts dataset**
- Train using **Google Colab's Free GPU**

✅ **Free Resources**

- **Google Colab Free Tier** (uses Tesla T4 GPU)
- **Hugging Face Pre-trained Model** (transformers library)

✅ **Where?**

- Deploy on **FastAPI (/summarize route)**

Training Topic Classification Model (BERT)

🗓️ **Day 8-9**
✅ **Why?**
To classify papers into topics like **AI, Cybersecurity, NLP, etc.**

✅ **How?**

- Use **bert-base-uncased (free Hugging Face model)**
- Fine-tune on **labeled dataset from arXiv**
- Train using **Google Colab (Free GPU)**

✅ **Free Resources**

- **Google Colab Free Tier**
- **arXiv Open Dataset**

✅ **Where?**

- Deploy on **FastAPI (/classify route)**

Training Methodology Extraction Model (NER)

🗓️ **Day 10-11**
✅ **Why?**
To extract methodologies like **CNN, LSTM, SVM, PCA** from research papers.

✅ **How?**

- Train **spaCy NER** on a **dataset of methodology terms**
- Use **bert-base-NER** with **BIO tagging**

✅ **Free Resources**

- **Google Colab Free Tier**
- **Hugging Face Model Hub**

✅ **Where?**

- Deploy on **FastAPI (/methods route)**

📌 **Phase 3: Backend Development (Week 3)**

Creating FastAPI Backend

🗓️ **Day 12-13**
✅ **Why?**
To provide a **REST API** for the frontend.

✅ **How?**

- Define routes:
    - /generate-idea → Research Idea Generator
    - /summarize → Research Paper Summarization
    - /classify → Topic Classification
    - /methods → Methodology Extraction

✅ **Free Hosting**

- Deploy **FastAPI on Render (Free Tier)**

Implementing User Authentication

🗓️ **Day 14-15**
✅ **Why?**
Users need authentication to **save & retrieve their research ideas and papers**.

✅ **How?**

- Use **JWT Authentication**
- Store users in **PostgreSQL (users table)**

✅ **Free Resources**

- **FastAPI + OAuth2**
- **Supabase (Free PostgreSQL)**

📌 **Phase 4: Frontend Development (Week 4)**

1️⃣ Setting Up React Frontend

🗓️ **Day 16-17**
✅ **Why?**
To allow users to **upload research papers and get results**.

✅ **How?**

- Use **React + Vite**
- Install **Tailwind CSS** for styling

✅ **Free Hosting**

- **Vercel / Netlify Free Tier**

🔟 **Building UI Components**

🗓️ **Day 18-19**
✅ **Why?**
Users need an **interactive UI** to view results.

✅ **How?**

- Create **UploadForm**
- Create **SummaryCard**
- Create **TopicBadge**
- Create **MethodologyList**
- Create **IdeaGenerator**

✅ **Where?**

- Folder: /frontend/components

🔗 **Connecting Backend & Frontend**

🗓️ **Day 20-21**
✅ **Why?**
To send user input to the backend and fetch results.

✅ **How?**

- Use **Axios** for API calls
- Manage state with **React useState**

✅ **Free Resources**

- **Axios (Free NPM package)**
- **React Hooks (Built-in)**

📌 **Phase 5: Deployment & Testing (Final Week)**

🚀 **Deploying Backend to Cloud**

🗓️ **Day 22-23**
✅ **Why?**
To make the API accessible.

✅ **How?**

- Deploy FastAPI on **Render Free Tier**

✅ **Free Hosting**

- **Render (Free for FastAPI)**

3️⃣Deploying Frontend

🗓️ **Day 24-25**
✅ **Why?**
To make the UI accessible.

✅ **How?**

- Deploy React on **Vercel (Free Tier)**

✅ **Free Hosting**

- **Vercel / Netlify (Free)**

4️⃣Final Testing & Debugging

🗓️ **Day 26-28**
✅ **Why?**
To ensure the app works perfectly.

✅ **How?**

- Test API with **Postman**
- Test UI with **Jest**

✅ **Free Tools**

- **Postman (Free Plan)**
- **Jest (Free for React)**

🚀 **Final Outcome**

✅ **Users can upload papers, generate ideas, get summaries, classify topics, and extract methodologies—completely free!**

📌 **Need code for any step? Let me know!** 🚀🔥

Phase	Task	Days	Why?	How?
Phase 1: Data Collection & Preprocessing (Week 1)	**Collecting Research Papers**	Day 1-2	Needed for summarization, classification, and methodology extraction.	Scrape from arXiv API, Semantic Scholar API, or Google Scholar (manual).
	Cleaning & Preprocessing	Day 3-4	Remove irrelevant data, improve accuracy.	Convert text to lowercase, remove stopwords & citations, tokenize sentences.
Future Project Idea Generation	**Analyzing Trends for New Research Areas**	Day 5-6	Recommend new research topics based on trends.	NLP-based topic modeling on research papers.
Phase 2: Model Training & Evaluation (Week 2)	**Training Summarization Model (T5/BART)**	Day 5-6	Generate concise research paper summaries.	Fine-tune T5/BART models, train with Cross-Entropy Loss.
	Training Topic Classification Model (BERT/GPT)	Day 7-8	Classify papers into AI, Cybersecurity, NLP, etc.	Fine-tune bert-base-uncased with a Softmax layer on labeled data.
	Training Methodology Extraction Model (NER)	Day 9-10	Extract methodologies (CNN, LSTM, SVM, PCA).	Train spaCy NER and BERT-based NER with BIO tagging.
Phase 3: Backend Development & API Creation (Week 3)	**Creating the FastAPI Backend**	Day 11-12	Provide REST API for summaries, topics, methodologies.	Define API routes (/summarize, /classify, /methods).
	Implementing User Authentication	Day 13-14	Secure user data, manage research history.	Use JWT authentication, store users in PostgreSQL.
Phase 4: Frontend Development (Week 4)	**Setting Up React Frontend**	Day 15-16	Build a UI for uploading papers & displaying results.	Use React + Vite, Tailwind CSS for styling.
	Building UI Components	Day 17-18	Make the UI interactive for users.	Create UploadForm, SummaryCard,

				TopicBadge, MethodologyList.
	Connecting Backend & Frontend	Day 19-20	Fetch and display results from backend.	Use Axios to call FastAPI, handle responses with useState.
Phase 5: Deployment & Testing (Final Week)	**Deploying Backend to Cloud**	Day 21-22	Make API publicly accessible.	Deploy FastAPI on AWS EC2/DigitalOcean with Gunicorn & Nginx.
	Deploying Frontend	Day 23-24	Provide users with a web interface.	Build React app (npm run build), deploy on Vercel/Netlify.
	Final Testing & Bug Fixing	Day 25-28	Ensure the app runs smoothly.	Test API with Postman, UI with Jest + React Testing Library.
Final Outcome	☑ Users can upload papers & get automated summaries.			
	☑ Topics & methodologies are extracted.			