
📰 Project Report: Daily Dev Digest

Powered by Gemini 1.5 Flash and Streamlit

📌 Project Title

Daily Dev Digest: An AI-Powered Personalized Digest for Software Developers

---

📚 Abstract

Daily Dev Digest is a mobile-optimized web application designed to deliver personalized, concise, and informative daily technical summaries to software developers. By leveraging modern APIs and large language models, the application aggregates content from three major sources — **GitHub**, **arXiv**, and **Medium** — based on user-defined keywords. It then uses **Google Gemini 1.5 Flash** to generate rich, structured notes, enhancing the user's technical knowledge efficiently and interactively.

🎯 Objectives

- Allow users to explore daily technical content by **keyword/topic**
- Fetch top repositories from **GitHub**
- Find academic papers from **arXiv (cs.SE)** category
- Retrieve blog posts from **Medium** based on tags
- Generate **detailed technical summaries** using LLMs
- Make the app **mobile-friendly** for daily use on-the-go
- Enable **offline-like access** via "Add to Home Screen" functionality

---

🛠️ Technology Stack

Component	Tool / Library
Language	Python
Web Framework	Streamlit

Component	Tool / Library
LLM	Gemini 1.5 Flash (Google AI)
APIs	GitHub REST API, arXiv RSS, Medium RSS
Summarization	google-generativeai SDK
UI Deployment	Streamlit Local / Cloud
Env Management	python-dotenv
Parser	feedparser (for RSS feeds)

🧩 System Architecture

1. **Input**:
   User enters a topic keyword (e.g., "code generation").
2. **Data Collection**:
   - GitHub: Searches top-starred repositories with that keyword
   - arXiv: Fetches recent research papers from the *cs.SE* category
   - Medium: Retrieves blog posts tagged with the keyword
3. **Summarization with Gemini**:
   The collected content (README, abstracts, summaries) is passed to **Gemini 1.5 Flash** with a well-crafted prompt to extract:
   - The problem being solved
   - Techniques/implementation details
   - Real-world relevance or impact
4. **Display**:
   Results are shown in sections (GitHub, Research Paper, Medium) with formatted titles, links, and detailed Gemini-generated notes.
5. **Mobile Optimization**:
   The app can be pinned to the home screen on Android/iOS for a native app-like experience.

🧠 Gemini Prompt Design

To generate insightful summaries, a structured prompt was used:

Carefully read the following [type] and produce a comprehensive, well-structured set of notes...

Focus on:

- What the work is about

- What was done (techniques, tools)

- Why it matters

- Limitations (if any)

Avoid author names and metadata...

✅ **Key Features**

- 🔍 **Keyword-based content search**
- 🤖 **LLM-generated developer-focused summaries**
- 📱 **Mobile-friendly UI** with "Add to Home Screen"
- 📘 **Educational focus**: summaries are structured like study notes
- 📄 Support for **research, tools, and blogs** in one interface

---

💡 **Use Cases**

- Developers seeking daily technical learning
- Students exploring academic or open-source content
- Researchers comparing projects and papers
- Professionals staying updated on trends

🧪 **Challenges Faced**

- Handling missing/empty GitHub READMEs
- Parsing inconsistent Medium RSS summaries
- Ensuring URL-safe keyword encoding

- Optimizing prompt engineering for relevant output
- API rate-limiting (GitHub unauthenticated requests)

🛠️ Enhancements / Future Work

- Add user history & daily log
- Export digest to PDF or Email
- Integrate Google Scholar for richer research indexing
- Add multiple GitHub/paper results per keyword
- Personalize feed by interests (ML, Web, Testing, etc.)
- Schedule digests with push notifications

📸 Screenshots / UI Overview

(*Add screenshots showing GitHub repo, research paper, Medium post with summaries on mobile or desktop view*)

📁 Project Structure

📦 daily-dev-digest/

```
├── app.py              # Main Streamlit application
├── .env                # Environment file for Gemini API key
├── requirements.txt    # Python dependencies
└── README.md           # Project description and usage
```

📜 Conclusion

Daily Dev Digest simplifies the process of staying technically informed by automatically collecting and summarizing relevant open-source, academic, and community-driven content. It empowers developers to learn faster and smarter, making AI-powered technical briefings part of their everyday routine.

📎 Appendix

🔑 **.env file (example):**

GOOGLE_API_KEY=your_gemini_api_key_here

📥 **Install Requirements:**

pip install -r requirements.txt

▶️ **Run App:**

streamlit run app.py
