AirGap-Chat

Browser-based AI chat interface for air-gapped environments. No server. No data transmission. Model-agnostic.

Repository: https://github.com/ChadsCode/AirGap-Chat

What This Is

Glue code. A model-agnostic UI that connects to whatever language model you choose. Currently configured for Microsoft Phi-3 (MIT licensed), but swap one line and it runs any WebLLM-compatible model.

The interface isn't the point. The architecture is.

Why It Matters

If you're in a regulated industry (finance, healthcare, government, legal, defense, insurance, pharmaceuticals), you already know the problem. You want AI capabilities, but certain data like PII and PHI can't leave your environment.

This runs entirely in the browser. Nothing is sent to a server. No API logs. Your data stays yours.

Quick Start

1. Clone this repository
2. Serve files via localhost (ES modules required)
3. Open index.html in Chrome
4. Click "Load Phi-3 Model"
5. Chat

Requirements

- Google Chrome with WebGPU support (only browser tested)
- Local server environment
- ~2.3GB disk space for model cache
- Internet connection for initial model download only

Changing Models

Edit one line in `app.js`:

```
javascript
```

```
engine = await CreateMLCEngine(
  "Phi-3-mini-4k-instruct-q4f16_1-MLC", // Change this line
  { ... }
);
```

Compatible models include:

- Llama-3-8B-Instruct-q4f16_1-MLC
- Mistral-7B-Instruct-v0.2-q4f16_1-MLC
- Gemma-2B-it-q4f16_1-MLC
- Any other WebLLM-supported model (see https://github.com/mlc-ai/web-llm)

One line. New model. Everything else stays the same.

Project Structure

```
AirGap-Chat/
├── index.html         # Main application
├── style.css          # UI styling
├── app.js             # Core logic
├── check_models.html  # Model testing utility
└── README.md
```

Less than 650 lines of vanilla code. No framework dependencies beyond WebLLM. Easy to audit, easy to extend.

Use Cases

- Healthcare: HIPAA-compliant AI without PHI transmission
- Finance: Regulatory compliance where external APIs are prohibited
- Government: Classified or sensitive environments
- Enterprise: Internal AI tools without cloud dependencies

Technical Details

- Framework: Vanilla JavaScript (no dependencies, easier to audit)
- AI Engine: WebLLM (loaded via CDN import)
- Acceleration: WebGPU required
- Architecture: Fully client-side, static deployment

Performance Notes

Response times depend on hardware. On older hardware with integrated graphics, expect 30 seconds to 2 minutes per response. Production deployment would benefit from local GPU infrastructure.

Known Issues

Chrome on Windows may show a console warning about `powerPreference`. This is a known Chrome bug (https://crbug.com/369219127) and does not affect functionality.

Browser Compatibility

Only tested with Google Chrome. Other browsers have not been tested.

License

MIT License - Free for commercial and personal use

Author

Chad Wigington LinkedIn: https://www.linkedin.com/in/chadwigington/ GitHub: https://github.com/ChadsCode

Disclosures

1. Personal hobby project created prior to employment.
2. Not associated with or endorsed by any employer.
3. This project is independently developed and is not affiliated with, endorsed by, or sponsored by Microsoft, Hugging Face, or MLC AI.
4. Microsoft, Phi-3, and GitHub are trademarks of Microsoft Corporation. Hugging Face is a trademark of Hugging Face, Inc. WebLLM is a project of MLC AI. Use of these names is for identification purposes only and does not imply endorsement, sponsorship, or affiliation. All trademarks remain the property of their respective owners.
5. Have all code professionally verified before use.
6. Views are my own.

Questions? Open an issue or reach out via LinkedIn: https://www.linkedin.com/in/chadwigington/