CONSENTCANVAS:AI NEGOTIATED TOOL

This tool can we made into a chrome extension tool which addresses the issue where we mostly accept the terms and conditions and cookies without going through it in details, So this tool will go through in detail in behalf of you and ensure only the necessary and safe permission and consent is granted.

Tech Stack:

| **Type** | **Stack** |
| --- | --- |
| **Frontend** | JavaScript, HTML, CSS, localStorage |
| **AI Layer** | HuggingFace models / Llama.cpp |
| **Backend (optional)** | FastAPI, SQLite, Python |
| **DevOps** | GitHub, GitHub Pages, Render, Railway |
| **Browser Extension (optional)** | JS + Manifest v3 |

***Step1:Website Frontend(React)***

can be loaded as a Chrome Extension which would detect the presence of cookie policy.

* Insert a javascript file to detect the consent and cookie popups.
* The raw html and text is extracted.

**Step2:Data Transfer to backend**

Extract the details of the policy in the form of text using Axios call . using DOM

* The extracted raw html post to backend using axios

**Step3:Data Analysis**

Now the input would be processed by a hugging face model where we identifies the type of cookies asking for permission. Its third party policies and all sorts of terms and conditions. Tracking permissions

* Label the cookies using NER tagging the parts which mention ads,analytics
* Split the text into clauses using spaCy and classify it into the manipulative clauses or clear clauses and check risk score by assigning scores to each clauses .using scikit

**Step 4: Preference match**

User must already load the user preferences before hand in a db.

For example: If the user does not want to give location permissions would deny.

Match the tagged clauses with preference

It will give output as the json object.

**Step 5:Ai generated consent preference**

* It will generate a text summary for user convenience using a hugging io model.
* Json object schema is generated using a json schema builder.

**Step 6: Set consent decisions on behalf of the user.**

* It will auto click reject cookie banners. After the page load, query SelectorAll will find button/ links where reject decline such words would have been mentioned to check them.
* To prevent any trackers after loading the declarativeNetRequests block the trackers as per the preference output json schema.

We will have set of block rules based on the user preference.

* Mutation Observer to check if further new buttons or terms pop up. And re run the rejectcookies to find and reject the new buttons.
* Convert the user json schema into actual blocking rules using chrome.declarativeNetRequest
* Save all the rules json
* Chrome loads the rules json and starts blocking the file/links.

Step 7:Optional feedback form to understand how to make the model better.

* Basic issue that user faced, its stored in DB and prefer analysis of the feedbacks selecting the most common ones and most issues faced. Using Promethus and Grafana to measure and show the feedbacks data

**PIPELINE DIAGRAM**

**[Browser Extension]**

├── JS Script: Scan DOM for Consent Banners

├── POST Extracted Text to API

**[FastAPI Backend]**

├── NLP Pipeline → HuggingFace Transformers, LLaMA.cpp

├── Compare with User Privacy Rules (IndexedDB/TinyDB)

├── Generate AI Summary + Consent JSON

└── Return Action Instructions to Browser

**[Extension Again]**

├── Inject Consent Actions to DOM

├── Block Tracking Requests (declarativeNetRequest)

├── Show AI Summary to User

└── Log Feedback (Optional)