उपभोक्ता मामले विभाग
DEPARTMENT OF CONSUMER AFFAIRS

Dark Patterns Buster
Hackathon 2023

Dark Patterns Buster Hackathon (DPBH-2023)

Round 3 Grand Finale @ IIT(BHU), Varanasi

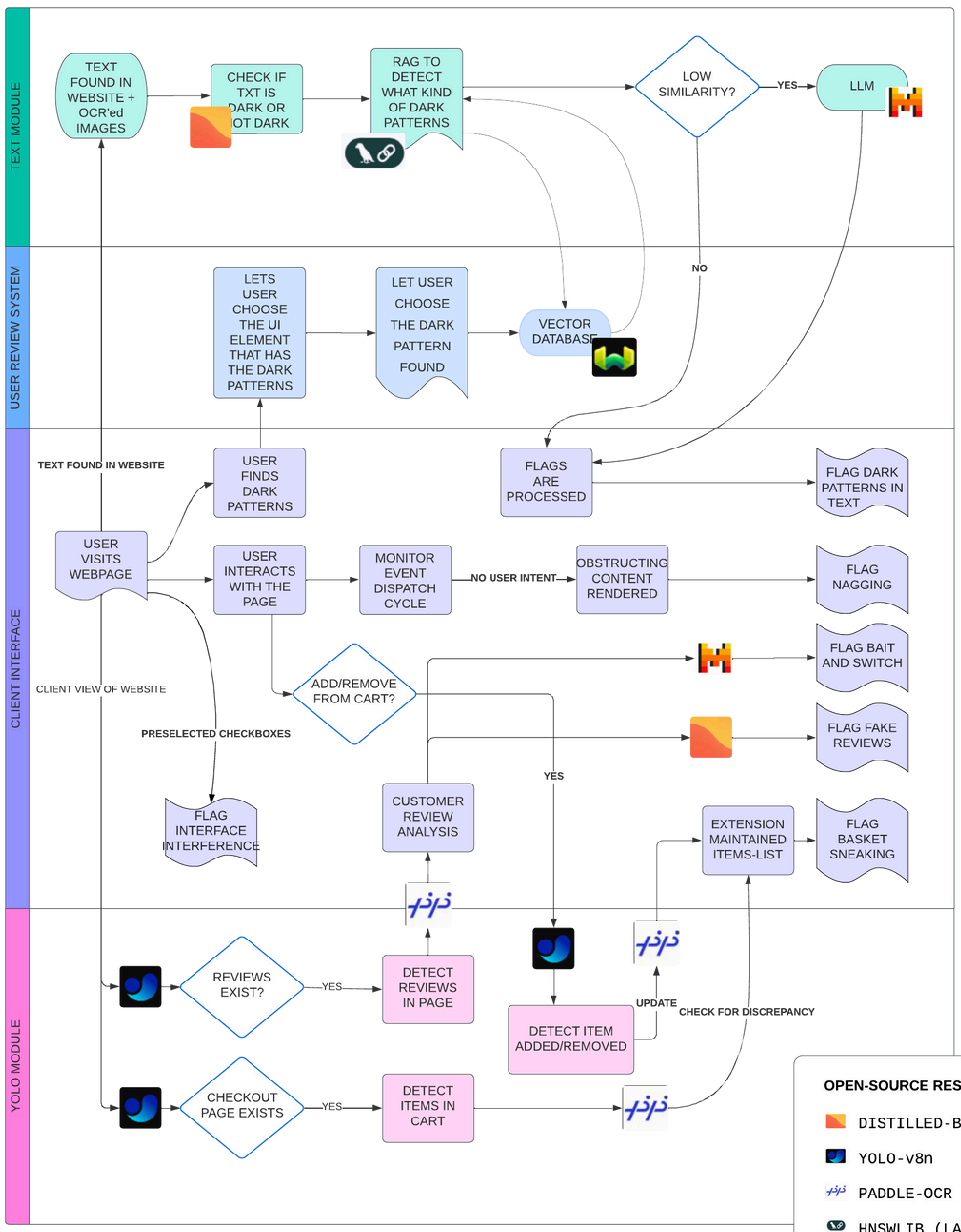
(Feb.17, 2024)

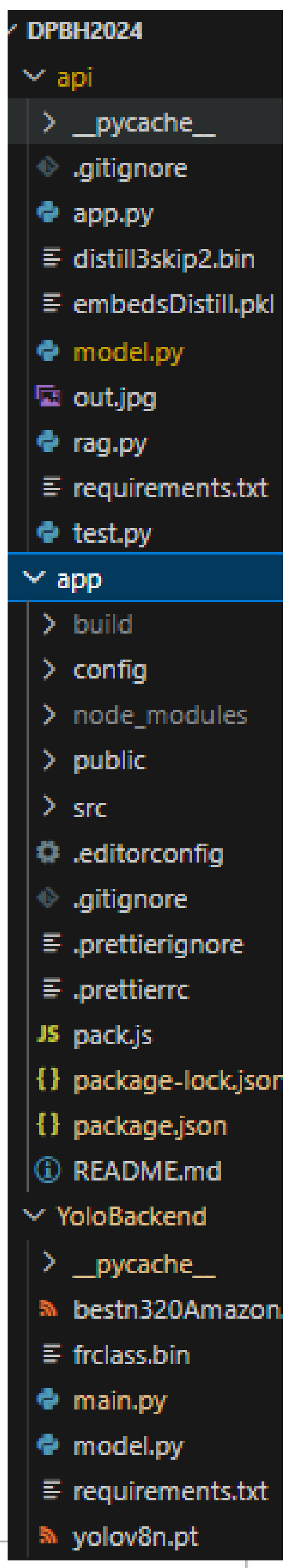
IIT Indore

Team #3@IIT Indore
(Deception Detectors)

Vishnu V Jaddipal
Ashutosh Samantaray
Gaurav Mahendraker
Rupal Shah
Samip Shah

How The Browser Extension Works:

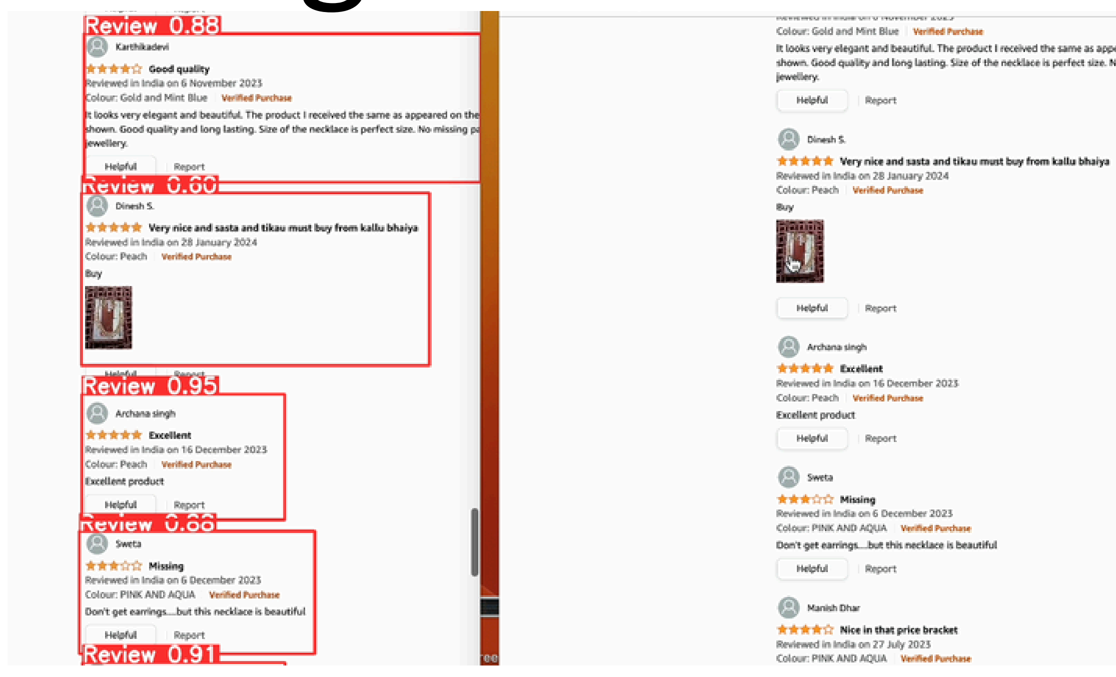



```
DPBH2024
  api
    > __pycache__
    .gitignore
    app.py
    distill3skip2.bin
    embedsDistill.pkl
    model.py
    out.jpg
    rag.py
    requirements.txt
    test.py
  app
    > build
    > config
    > node_modules
    > public
    > src
    .editorconfig
    .gitignore
    .prettierignore
    .prettierrc
    pack.js
    package-lock.json
    package.json
    README.md
  YoloBackend
    > __pycache__
    bestn320Amazon
    frclass.bin
    main.py
    model.py
    requirements.txt
    yolov8n.pt
```

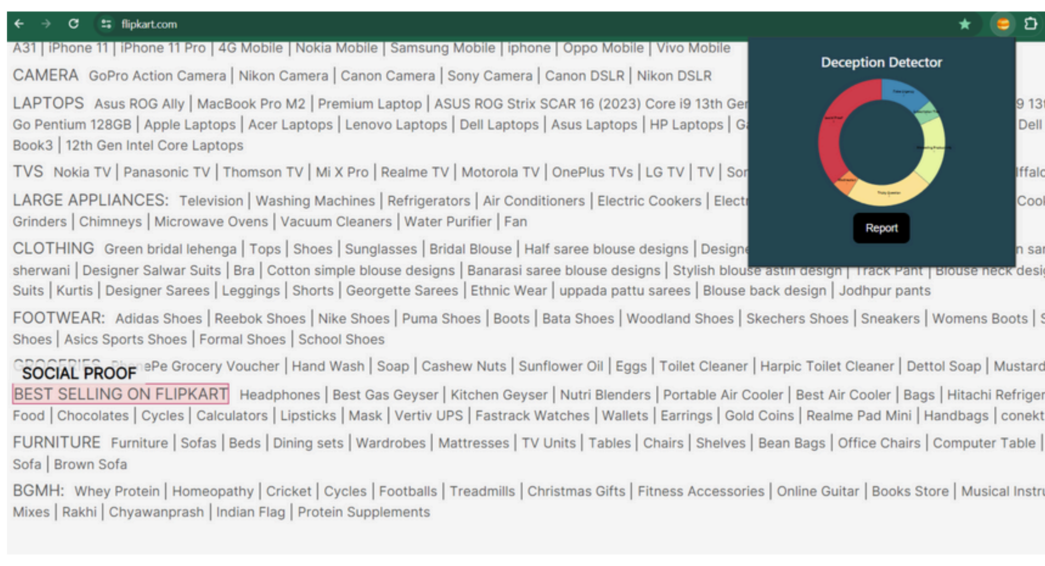
OPEN-SOURCE RESOURCES USED

- 🟧 DISTILLED-BERT
- YOLO-v8n
- PADDLE-OCR
- HNSWLIB (LANGCHAIN)
- MISTRAL
- WEAVIATE

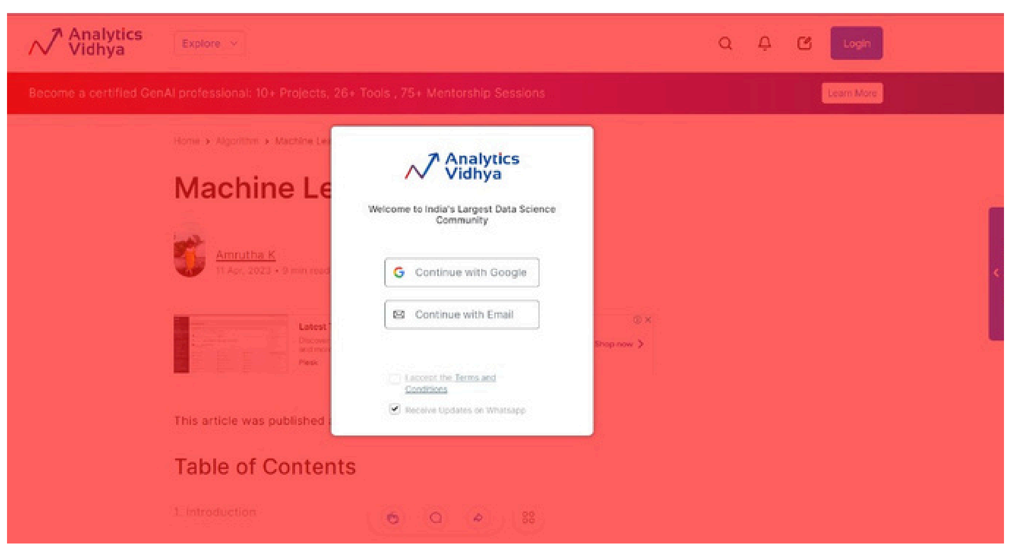
Plugin in action



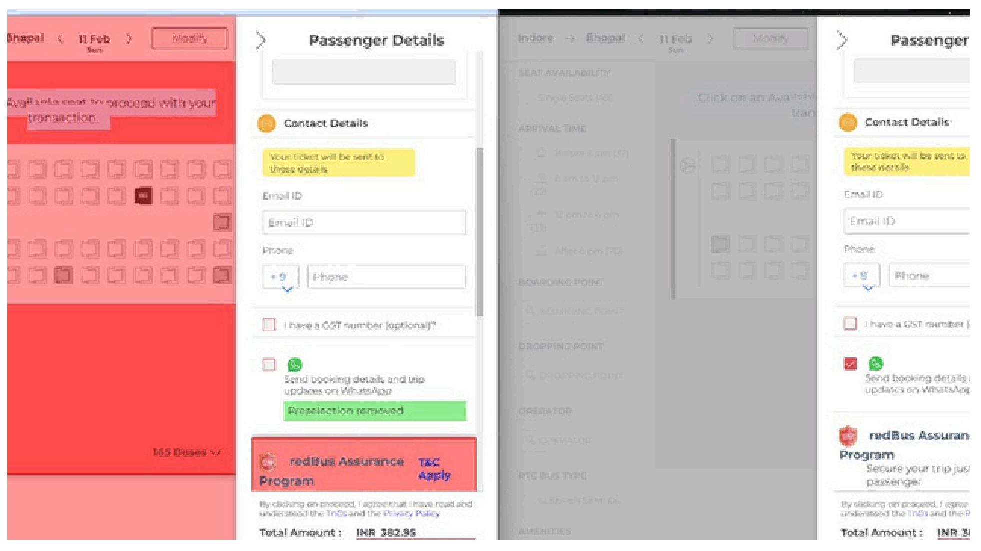
Review Detection



Dark patterns flagged on Flipkart with type breakdown



Extension flagging Forced Action



Interface Interference being solved

Handled dark patterns

- Confirm Shaming
- Subscription Trap
- False Urgency
- Forced Action
- Social Proof
- Scarcity
- Fake Review Detection
- Interface Interference
- Nagging
- Obstruction
- Basket Sneaking
- Misdirection

Scan this for Documentation

Contact:
Vishnu V Jaddipal
cse210001078@iiti.ac.in

Mentor: Dr. Nagendra Kumar
Asst. Professor, Dept. of CSE
IIT INDORE

Dr. N. S Rajput
Convener, DPBH-2023
Email: dpbh2023@iitbhu.ac.in

उपभोक्ता मामले विभाग
DEPARTMENT OF
CONSUMER AFFAIR

Dark Patterns Buster
Hackathon 2023

ABSTRACT

Explanation for different components of our architecture:

1. **YOLO Module:**
   a. used for detecting reviews, carts, and determining items that are being added/removed by the user.
   b. These items and reviews are stored in varying ways across different websites we use a YOLOv8n model for robustly detecting them, and use paddleOCR for extracting the text from them,
   c. the nano version of YOLO is used as it is lightweight. For **scalability**, we have deployed the model using **TF-Lite on the client side.** This allows for **realtime processing and less processing load on the server.**
2. **Text Module:** the text module uses a 3 layer architecture,
   a. we have trained a **distill-roberta model** for determining whether th**e text is dark or not-dark**, this is to weed out possible false positive alaems in our classification layer.
   b. Our next layer is a RAG-based system that queries our database to determine similarity between the target text, and possibly similar samples in our database from **(subscription trickery, false urgency, confirm shaming, misdirection and scarcity).**
   c. we store the embeddings of our text data and their corresponding labels (taken from mathur et al, and augmented using LLMs for different classes) in a **vector database (weaviate)**. **This is done for faster processing of requests for realtime response.**
   d. We use **RAG** for finding similarity because often common dark patterns have similarity across many different websites. and since it is able to **accomodate changing dark patterns**
   e. If similarity between the text, and samples in our database is low, we use LLM (mistral) for querying and classifying the text as dark pattern, note that we only use **LLMs in the worst case**, since querying **LLMs can be computationally expensive and not scalable.**
   f. **To reduce false negatives, the dataset** maintained for RAG is kept exceptionally **accurate**. We implement a **thresholding technique**. If the similarity **crosses** a certain threshold value, then only the respective text would be **labelled** and **flagged**.
3. **Client interface:** whenever the client visits a page and chooses to analyze it with our extension,
   a. the client sends the text to the text module, and the images to the YOLO module. for **realtime processing and flagging**, we have decided to use the **websockets** provided by fastAPI.
   b. We also provided a **report of found dark patterns** as a pie chart that is updated in realtime.
   c. We have a review analysis module, this module
      i. queries reviews into an LLM (mistral) to check if the **review is a complaint about bait and switch**.
      ii. queries to the fake review classification model to check if the **review is fake or real.**
   d. We listen for mutations to the DOM which are visible to the user, and to the **event dispatch cycle, for tracking user interactions**
      i. based on our rule based system, if the content is obstructing user interaction, and is likely not caused by user actions, we flag it as **nagging/forced action**
      ii. if the user has clicked on elements that are synonymous to adding/removing from cart, then we query our YOLO module, to track extensions' list, this list is checked against the cart items found on checkout (again by using YOLO) for **potential basket sneaking**
   e. We query to find out possible preselction of checkboxes by webpages, and remove and flag them.
4. **User review system:**
   a. We allow the user to report found dark patterns in websites with different labels, which we will then store in our vector database, this allows us to keep in touch with evolving dark patterns, which in-turn creates a robust model.
   b. The user can select a particular UI element, and then report it under a given label

Impact assessment

Our solution detects several dark patterns that are used to maliciously alter consumer behaviour, like confirmshaming, urgency, and scarcity etc. These dark patterns cost consumers their hard earned money and have negative impacts on consumers' mental health. The extension mitigates these harms. Moreover, the extension enables legal action to be taken against highly-reported websites.

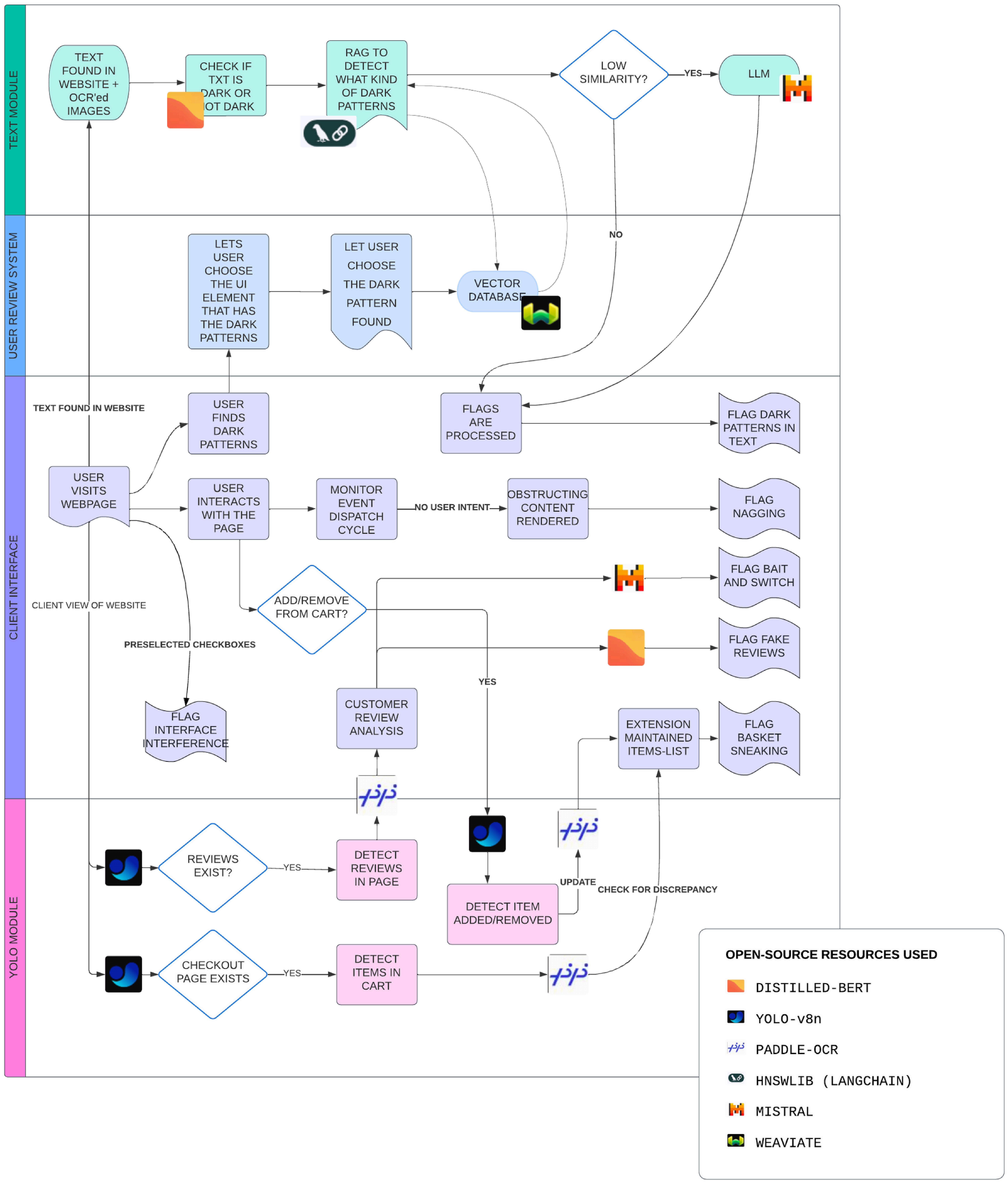
We have modularized our components, so that future solutions can easily integrate our components promoting further development in this field. Our database of dark patterns will be available for future use.

Highlights & Novelty:

- **3-level structure allows security computation balance.**
- **Retrieval block allows for learning in a dynamic manner with data reported by users.**
- **Image feature usage**
- **web sockets**
- **Forward compatibility for any evolving patterns**
- **Dataset Augmentation with LLMs**
- **Has a very low inference time and reflects in real time.**
- **Prevents the other dark patterns creating companies, and blocks the dark patterns sourced by them on the ecommerce websites.**
- **Amortized latency on the order of ms per sample.**

Team Members

Vishnu Jaddipal
Ashutosh Samantaray
Gaurav Mahendrakar
Samip Shah
Rupal Shah


TEXT MODULE
USER REVIEW SYSTEM
CLIENT INTERFACE
YOLO MODULE
TEXT FOUND IN WEBSITE + OCR'ed IMAGES
CHECK IF TXT IS DARK OR NOT DARK
RAG TO DETECT WHAT KIND OF DARK PATTERNS
LOW SIMILARITY?
YES
LLM
NO
LETS USER CHOOSE THE UI ELEMENT THAT HAS THE DARK PATTERNS
LET USER CHOOSE THE DARK PATTERN FOUND
VECTOR DATABASE
TEXT FOUND IN WEBSITE
USER FINDS DARK PATTERNS
FLAGS ARE PROCESSED
FLAG DARK PATTERNS IN TEXT
USER VISITS WEBPAGE
USER INTERACTS WITH THE PAGE
MONITOR EVENT DISPATCH CYCLE
NO USER INTENT
OBSTRUCTING CONTENT RENDERED
FLAG NAGGING
FLAG BAIT AND SWITCH
ADD/REMOVE FROM CART?
CLIENT VIEW OF WEBSITE
PRESELECTED CHECKBOXES
FLAG FAKE REVIEWS
YES
FLAG INTERFACE INTERFERENCE
CUSTOMER REVIEW ANALYSIS
EXTENSION MAINTAINED ITEMS-LIST
FLAG BASKET SNEAKING
REVIEWS EXIST?
YES
DETECT REVIEWS IN PAGE
UPDATE
CHECK FOR DISCREPANCY
DETECT ITEM ADDED/REMOVED
CHECKOUT PAGE EXISTS
YES
DETECT ITEMS IN CART
OPEN-SOURCE RESOURCES USED
DISTILLED-BERT
YOLO-v8n
PADDLE-OCR
HNSWLIB (LANGCHAIN)
MISTRAL
WEAVIATE