# Dark Pattern Detector Documentation

## Table of Contents

1. Introduction

2. Project Overview

3. Tech Stack

4. Installation

5. Usage

6. Features

7. Implementation Details

8. Challenges and Future Improvements

9. Contributing

---

## 1. Introduction

Welcome to the Dark Pattern Detector project documentation! This project is designed to identify and analyze dark patterns on websites. Dark patterns are user interface design choices that intentionally trick users into taking actions they might not otherwise choose. Our Dark Pattern Detector aims to promote transparency and ethical design practices on the web.

## 2. Project Overview

The Dark Pattern Detector is built to analyze websites and highlight potential dark patterns using machine learning algorithms. The project is implemented using Python for backend development, Flask as the web framework, and JavaScript, HTML, and CSS for the frontend.

## 3. Tech Stack

- \*\*Backend:\*\*

- Python

- Flask

- \*\*Frontend:\*\*

- HTML

- CSS

- JavaScript

## 4. Installation

To set up the Dark Pattern Detector on your local machine, follow these steps:

## 5. Usage

Once the application is running, users can input a website URL into the provided form. The Dark Pattern Detector will then analyze the website and provide a report highlighting any potential dark patterns found.

## 6. Features

- \*\*Website Analysis:\*\* Enter a website URL and receive a comprehensive analysis of potential dark patterns.

- \*\*User-Friendly Interface:\*\* The web interface is designed to be intuitive and easy to use.

- \*\*Transparency Reports:\*\* View detailed reports on the detected dark patterns with explanations and suggested improvements.

## 7. Implementation Details

The Dark Pattern Detector utilizes machine learning algorithms to analyze website interfaces. The frontend is built using HTML, CSS, and JavaScript, while the Flask framework is employed for the backend. The project leverages various Python libraries for web scraping, data analysis, and machine learning.

## 8. Challenges and Future Improvements

- \*\*Challenges Faced:\*\*

- Overcoming the diversity of dark patterns.

- Handling dynamic and complex web interfaces.

- \*\*Future Improvements:\*\*

- Integration of additional machine learning models for improved pattern detection.

- Extension to browser extensions for real-time detection.

## 9. Contributing

We welcome contributions from the community! If you'd like to contribute, please follow our Contribution Guidelines.

* Soumyadip Sarkar
* Adrijo Bhowmik
* Ritam Saha
* Kanhaiya Lal Agarwal
* Kartika Singh

---

Feel free to customize this documentation based on the specifics of your project and its features. Good luck with your Dark Pattern Detector at the hackathon!