DeepCytes Capstone Project

Problem Statement	Cyber Resource Intelligence Platform (CyberGuard)
Mentor	Manav Rupani(DeepCytes)
Group Members	1) Aditya Sinha 2) Paresh Sonnekar 3) Shamita Jagarlamudi
Region Coverage	1) Italy 2) Latvia
Website URL	https://cyberguardbot.netlify.app/
Video Link	DeepCytes Capstone Project

Table of Content

1. Project Overview and Problem Statement

Cybercrime victims frequently face difficulty navigating fragmented reporting systems, legal authorities, and technical advisory bodies. This leads to delays in incident reporting, confusion regarding jurisdiction, and increased psychological stress.

CyberGuard addresses this challenge by providing a unified cyber resource intelligence platform that aggregates official cybersecurity resources, reporting portals, emergency contacts, and advisories into a centralized interface.

The primary objective of this platform is to reduce reporting delays and simplify access to reliable cyber intelligence resources through a region aware dashboard and AI assisted guidance.

2. User View Documentation

2.1 Overall User Experience

The platform provides a centralized cybersecurity assistance interface designed for:

1. Cybercrime victims seeking reporting resources
2. Users wanting threat awareness
3. Individuals needing emergency cybersecurity contacts

2.2 Main Pages and Features

Module	Features	Primary Tech
National Dashboard	Real-time incident counts, active threat levels, and quick-access widgets.	React + Lucide Icons
Emergency Directory	Regionalized cards for helplines with "Direct Connect" (tel:) capabilities.	JSON + Tailwind

Portal Mapping	Use-case specific links for reporting financial fraud, identity theft, or harassment.	Conditional Rendering
Advisory Feed	Chronological list of threats with type-based "High-Alert" tagging.	JavaScript Sort API

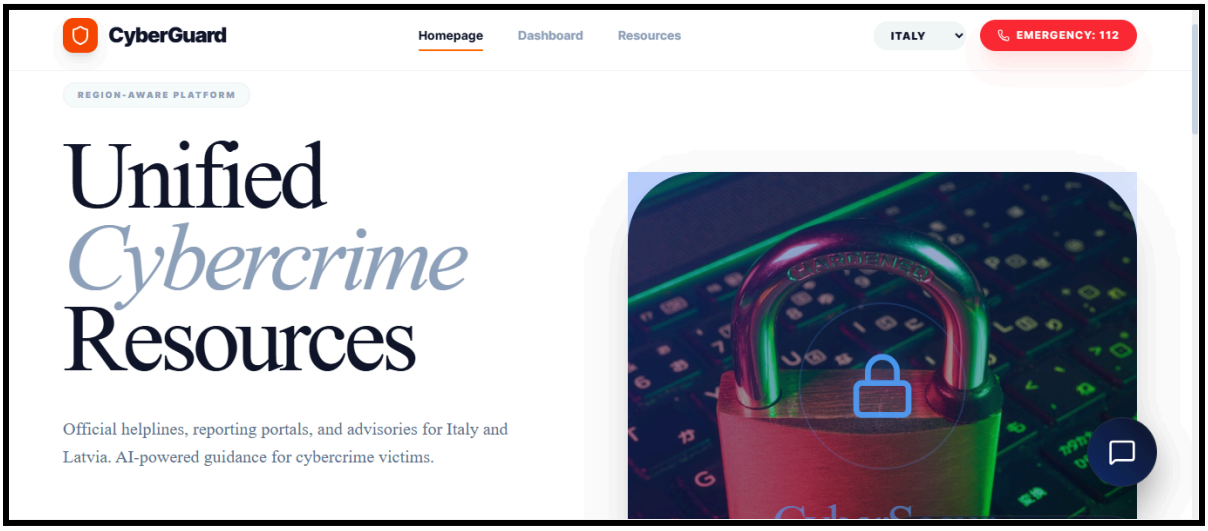

Fig 1: CyberGuard Platform Homepage

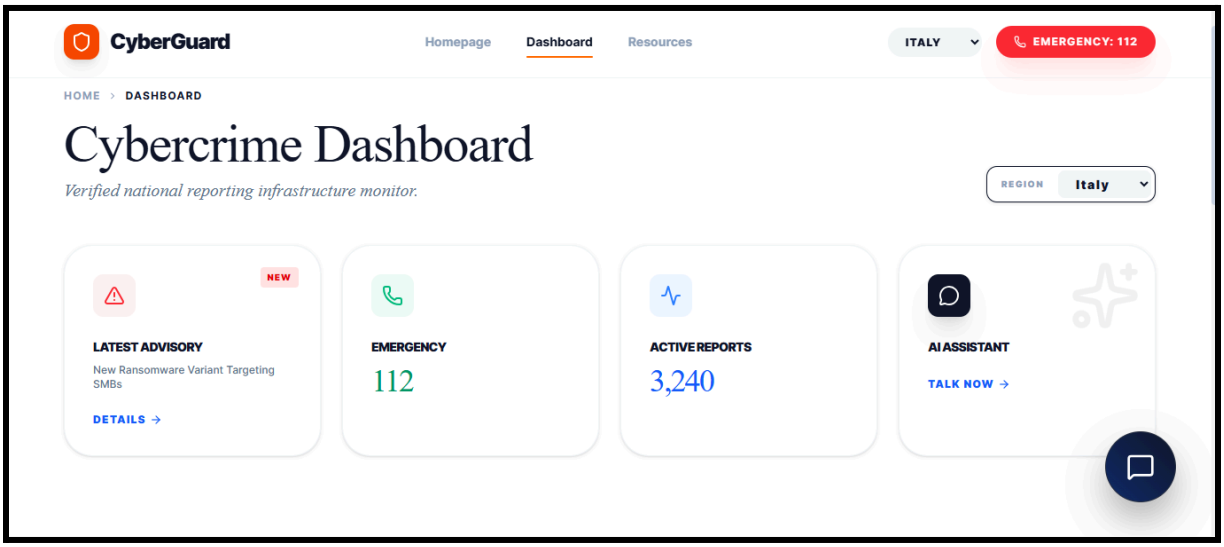

Fig 2: CyberGuard Platform Dashboard

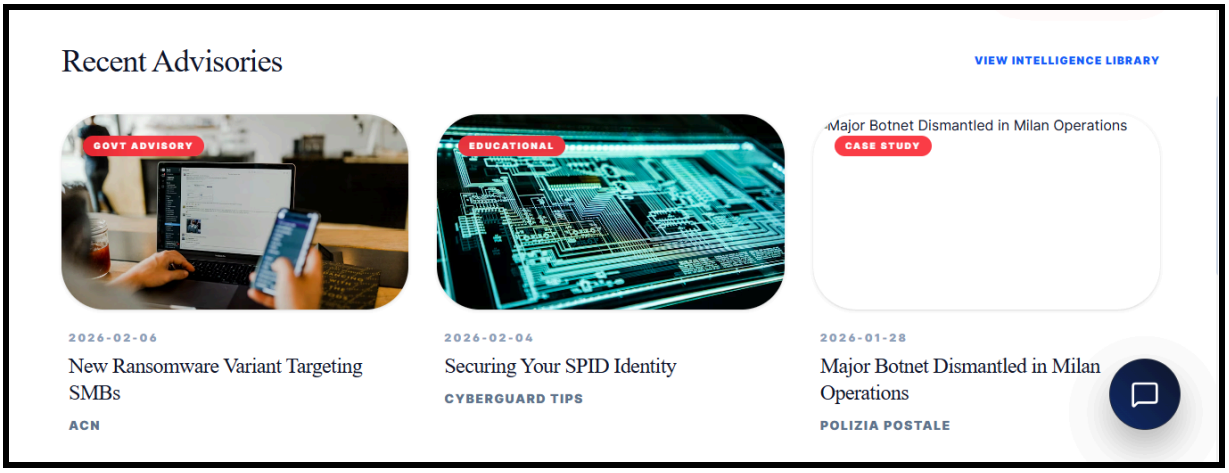

Fig 3: CyberGuard Platform Intelligence Library

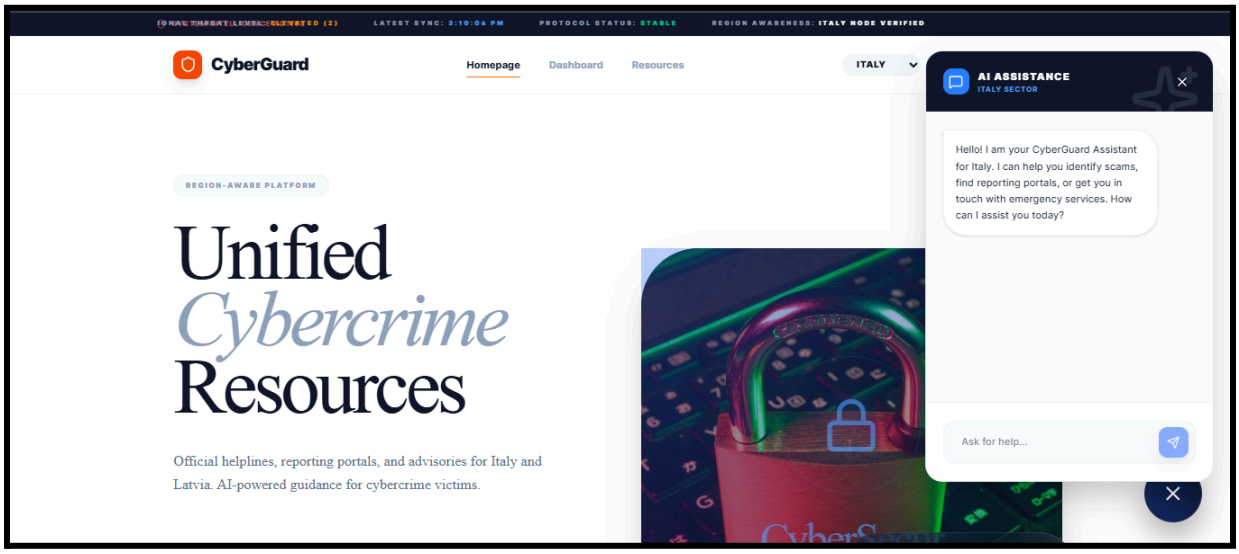

Fig 4: CyberGuard Platform Chatbot

2.3 Website Structure Overview

Main components include:

1. Landing interface
2. Regional dashboard
3. Resource directory
4. Advisory intelligence feed
5. AI chatbot module

3. Developer View Documentation

3.1 Technology Stack

Frontend:

1. React.js SPA architecture
2. Vite build tool
3. Tailwind CSS styling

Deployment and Version Control:

1. Netlify hosting platform
2. GitHub collaborative repository

3.2 System Architecture

The platform follows a modular component based architecture:

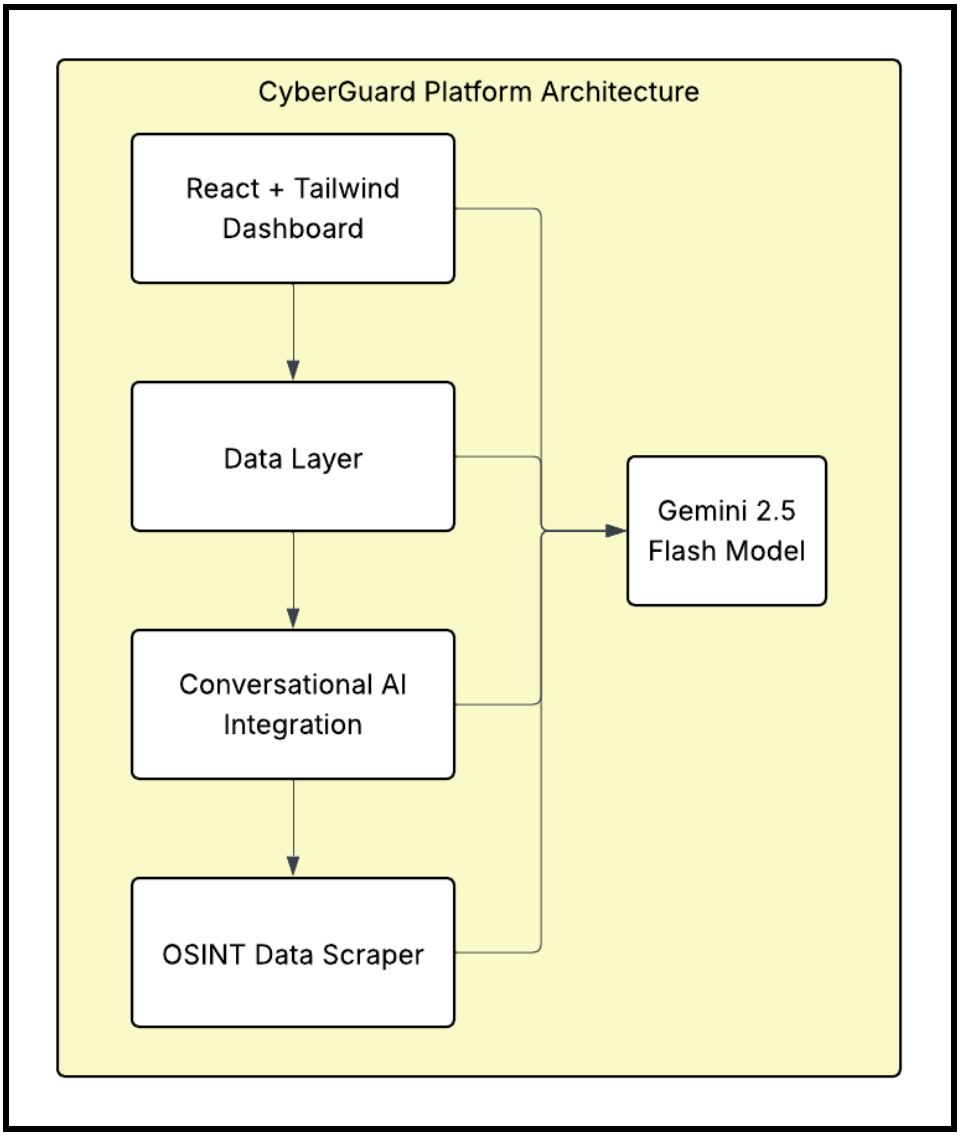

Fig 5: CyberGuard Platform Architecture (Made on lucidchart.io)

3.3 Data Layer Architecture

Three primary data categories:

1. **Static Emergency Data:** Hard coded emergency services for reliability.
2. **Portal Mapping Data:** Structured JSON linking cybercrime types to official reporting portals.
3. **Advisory Intelligence Feed:**Time sorted and categorized cybersecurity advisories.

3.4 Conversational AI Integration

The chatbot uses a context aware prompt system:

1. Region metadata injected dynamically
2. Response logic prioritizes emergency escalation
3. Stateless session handling for privacy.

3.5 Automation and OSINT Strategy

The platform conceptualizes an automated intelligence update system:

1. RSS advisory scraping
2. Government cybersecurity feeds integration
3. Periodic data synchronization via backend scheduler.

3.6 Project File Structure

Typical structure:

```
CyberGuard-AI/
|
├── src/
|   ├── main.jsx
|   ├── App.jsx
|   ├── components/
|   ├── styles/
|
├── index.html
├── package.json
├── vite.config.js
└── README.md
```

3.7 Local Setup Instructions

1. Clone repository:
   ```
   git clone https://github.com/shamitajan/CyberGuard-AI.git
   ```

2. Install dependencies:
   ```
   npm install
   ```
3. Run development server:
   ```
   npm run dev
   ```
4. Build production version:
   ```
   npm run build
   ```
5. Preview build:
   ```
   npm run preview
   ```

4. Ethical Considerations and Compliance

Privacy by Design

1. No storage of user personal data
2. Stateless chatbot sessions
3. Minimal telemetry collection.

Ethical Scraping Principles

If automation is implemented:

1. Request throttling
2. Source attribution
3. Compliance with official data policies.

Emergency Disclaimers

The platform clearly states:

1. It is an information aggregator.
2. It does not replace law enforcement.
3. Emergency contacts must be used directly for urgent cases.

5. Conclusion and Future Scope

CyberGuard demonstrates the feasibility of centralized cyber intelligence aggregation for improved victim response time and awareness.

Future enhancements include:

1. Direct reporting API integration
2. Multilingual advisory translation
3. Real time threat visualization maps
4. Automated intelligence ingestion pipelines.