# 🛡 CyberFence – Phishing URL Detection Platform
> A \*\*full&stack phishing URL detection platform\*\* with real&time analysis] admin dashboard] and user management- Built with \*\*Node-js] EÛpress] MongoDB] Socket-IO] and ONNX Runtime\*\*- Styled with a professional \*\*cyber&dark theme\*\*-
&&&
## 🌐 Features
& \*\*User authentication\*\* ¼JWT½ and role&based access ¼User/Admin½
& \*\*Real&time phishing URL detection\*\* using \*\*ONNX model\*\*: [pirocheto/phishing&url&detection]¼https://huggingface-co/pirocheto/phishing&url&detection½
& Admin dashboard with:
& KPI stats ¼Total Users] Active/Inactive] Total Uploads½
& Detection ratios ¼Phishing/Suspicious/Safe½
& Upload trends & heatmaps
& \*\*User upload history\*\*
& Real&time notifications via \*\*Socket-IO\*\*
& Secure password hashing ¼\*\*bcryptjs\*\*½
& Clean] professional \*\*cyber&dark UI\*\* for frontend
&&&
## 📁 Folder Structure
cyberfence&backend/│├─ models/│ ├─ User-js # User schema│ ├─ PhishingURL-js # URL analysis results│├─ routes/│ ├─ auth-js # Signup/Login endpoints│ ├─ detect-js # URL detection endpoints│ ├─ history-js # User history endpoints│ ├─ admin-js # Admin dashboard & management│├─ middleware/│ └─ auth-js # JWT auth & role validation│├─ controller/│ └─ adminController-js # Admin&related logic│├─ cyberFence-js # Phishing URL ONNX detection logic├─ server-js # Main entry point├─ package-json├─ -env # Environment variables└─ README-md
&&&
## ⚙ Environment Variables ¼`-env`½
```bash
MONGO\_URI¾<Your MongoDB connection URI>
PORT¾5000
JWT\_SECRET¾<Random secure string for JWT>
Make sure MongoDB is accessible via your URI-
🧩 Models
User Model ¼models/User-js½
Field
Type
Description
username
String] required] unique
Username of the user
email
String] required] unique
User email
password
String] required
Hashed password
role
String ¼user/admin½
Role&based access
isSuspended
Boolean] default false
Admin can suspend users
uploadsCount
Number] default 0
Number of URLs submitted
lastLogin
Date
Last login timestamp
PhishingURL Model ¼models/PhishingURL-js½
Field
Type
Description
url
String] required
The URL that was checked
results
MiÛed
Raw model output ¼probabilities½
verdict
String
"Phishing" / "Suspicious" / "Safe"
confidence
Number
Confidence score ¼0&100½
checkedAt
Date] default now
When the URL was analyzed
user
ObjectId ¼User½
Reference to the user who checked the URL
🔌 Routes
1- Authentication ¼/api/auth½
Endpoint
Method
Body
Description
/register
POST
¿ username] email] password À
Register new user
/login
POST
¿ email] password À
Login & receive JWT token
/me
GET
Header: Authorization: Bearer <JWT>
Get current user info
2- Phishing Detection ¼/api/detect/url½
Endpoint
Method
Body
Description
/url
POST
¿ urls: ["https://eÛample-com"] À
Detect phishing URLs
Sample Response:
¿
"success": true]
"results": [
¿
"id": "632f0b23---"]
"url": "http://phishing-com"]
"results": ¿ "probability": 0-98 À]
"verdict": "Phishing"]
"confidence": 98-0]
"user": "631f1a23---"]
"checkedAt": "2025&09&20T08:30:00-000Z"
À
]
À
3- User History ¼/api/history½
Endpoint
Method
Query Params
Description
/
GET
\page¾1&limit¾5&prediction¾Safe
Fetch user upload history
/reanalyze/:id
POST
N/A
Re&analyze a URL ¼future implementation½
4- Admin Routes ¼/api/admin½
Endpoint
Method
Description
/users
GET
List all users with pagination
/user/:id
PATCH
Update user role / suspend / unsuspend
/user/:id
DELETE
Delete user
/dashboard
GET
Admin stats & analytics ¼KPI + trends½
⚡ Phishing Detection Logic ¼cyberFence-js½
Uses ONNX Runtime Node-js
Input: array of URLs
Output: array of ¿ url] probability À
Verdict assignment:
if ¼probability >¾ 0-9½ verdict ¾ "Phishing";
else if ¼probability >¾ 0-5½ verdict ¾ "Suspicious";
else verdict ¾ "Safe";
Saves results in MongoDB
Emits real&time notifications via Socket-IO
Model Reference: pirocheto/phishing&url&detection
🖥 Frontend Integration ¼Info½
Admin & User dashboards consume /api/admin/dashboard and /api/history
Socket-IO updates for completed phishing checks:
socket-on¼"phishing\_check\_complete"] ¼data½ ¾> ¿
console-log¼data-newResults½;
console-log¼data-totalUploads½;
À½;
Frontend classes:
-dashboard&container] -kpi&card] -chart&card] -result&card
Cyber&dark theme: #0d1117 background] neon accent #58a6ff
📝 Setup Instructions
Clone the repository:
git clone <repo\_url>
cd cyberfence&backend
Install dependencies:
npm install
Setup -env file:
MONGO\_URI¾<your\_mongo\_uri>
JWT\_SECRET¾<your\_jwt\_secret>
PORT¾5000
Download ONNX model:
Place model file at -/models/model-onnÛ
Recommended model: pirocheto/phishing&url&detection
Start the server:
npm start
Access:
Backend API: http://localhost:5000/api/---
Frontend SPA: http://localhost:5000/
🛠 Technologies
Node-js & EÛpress
MongoDB & Mongoose
ONNX Runtime Node
Socket-IO
bcryptjs] jsonwebtoken
React ¼Frontend½
Chart-js / Recharts ¼Admin KPIs½
Cyber&dark UI theme
🔒 Security Notes
JWT authentication for all protected routes
Role&based access: user vs admin
Passwords hashed with bcrypt
ONNX model used for production&safe phishing detection
🧠 Model Details
Model Name: pirocheto/phishing&url&detectionType: ONNX model for phishing URL classificationPurpose: Classifies URLs as Safe] Suspicious] or Phishing
Input:
const urls ¾ [
"https://eÛample-com"]
"http://phishing&site-com"
];
ONNX Node-js Runtime:
const ort ¾ require¼"onnÛruntime&node"½;
const session ¾ await ort-InferenceSession-create¼"-/models/model-onnÛ"] ¿
eÛecutionProviders: ["cpu"]
À½;
const inputTensor ¾ new ort-Tensor¼"string"] urls] [urls-length]½;
const results ¾ await session-run¼¿ inputs: inputTensor À½;
Verdict Assignment:
let verdict;
if ¼probability >¾ 0-9½ verdict ¾ "Phishing";
else if ¼probability >¾ 0-5½ verdict ¾ "Suspicious";
else verdict ¾ "Safe";
Output: Array of probabilities per URL] saved in MongoDB for user history-