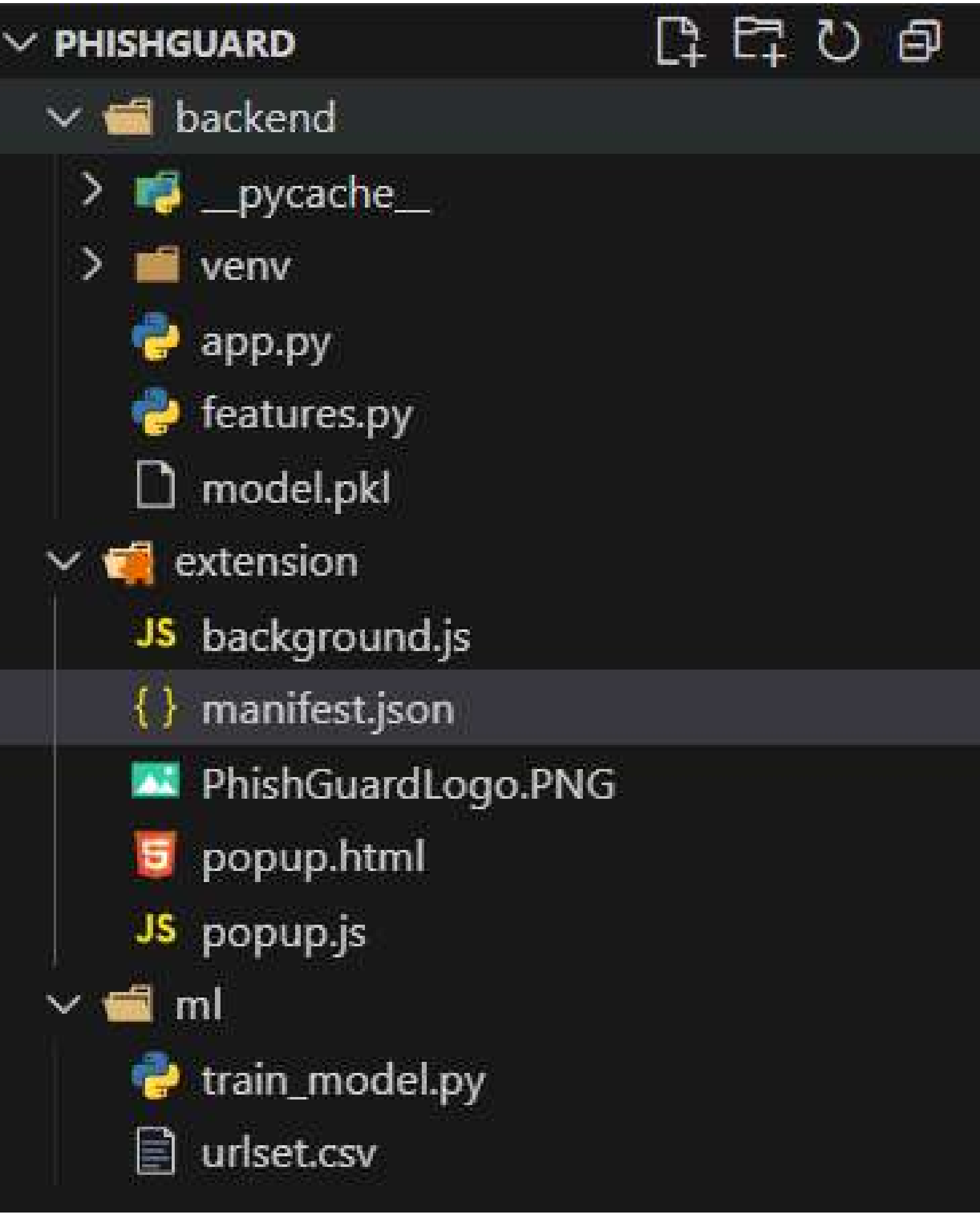

PHISHGUARD
backend
__pycache__
venv
app.py
features.py
model.pkl
extension
background.js
manifest.json
PhishGuardLogo.PNG
popup.html
popup.js
ml
train_model.py
urlset.csv

All Extensions

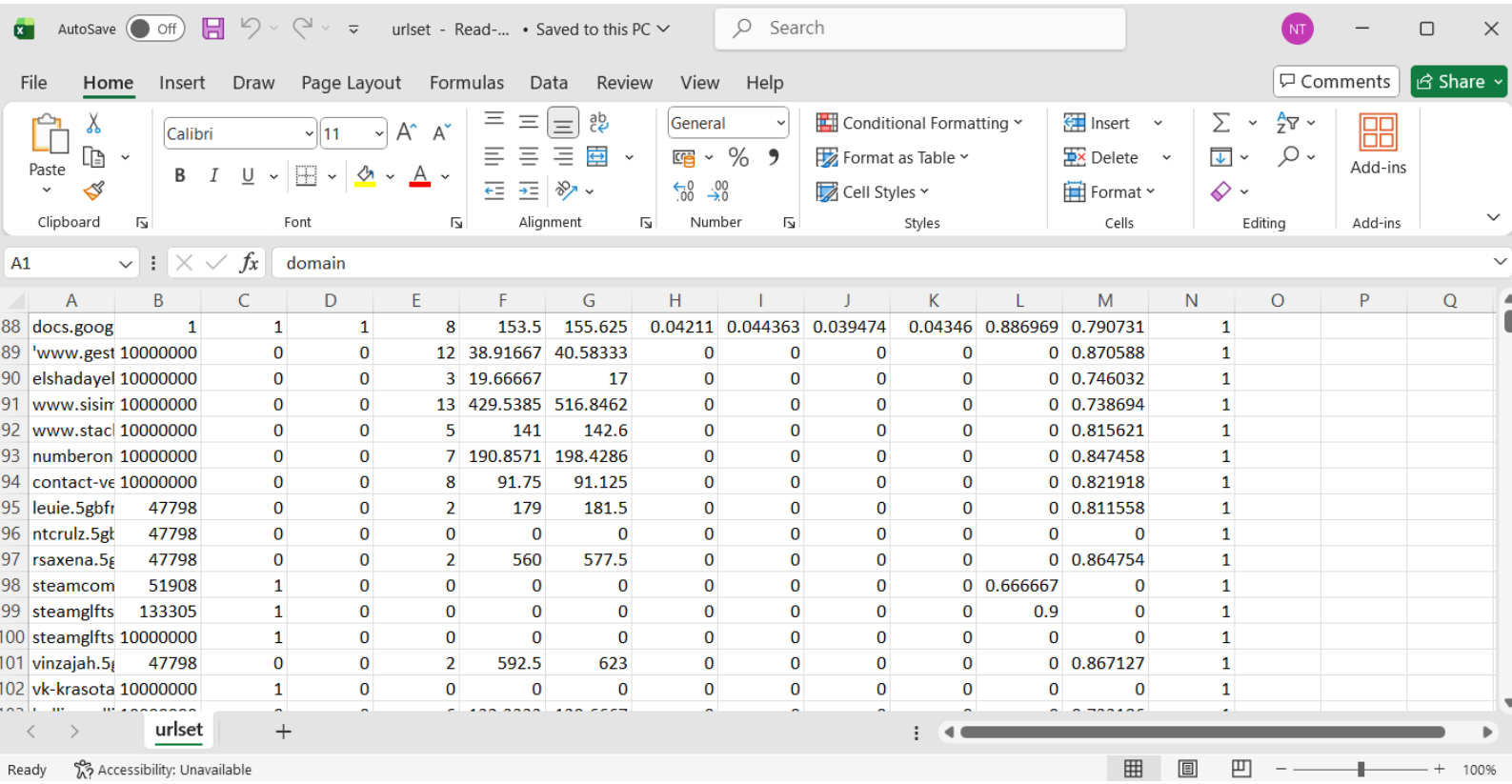
PhishGuard 1.0

AI-powered phishing protection for URLs.

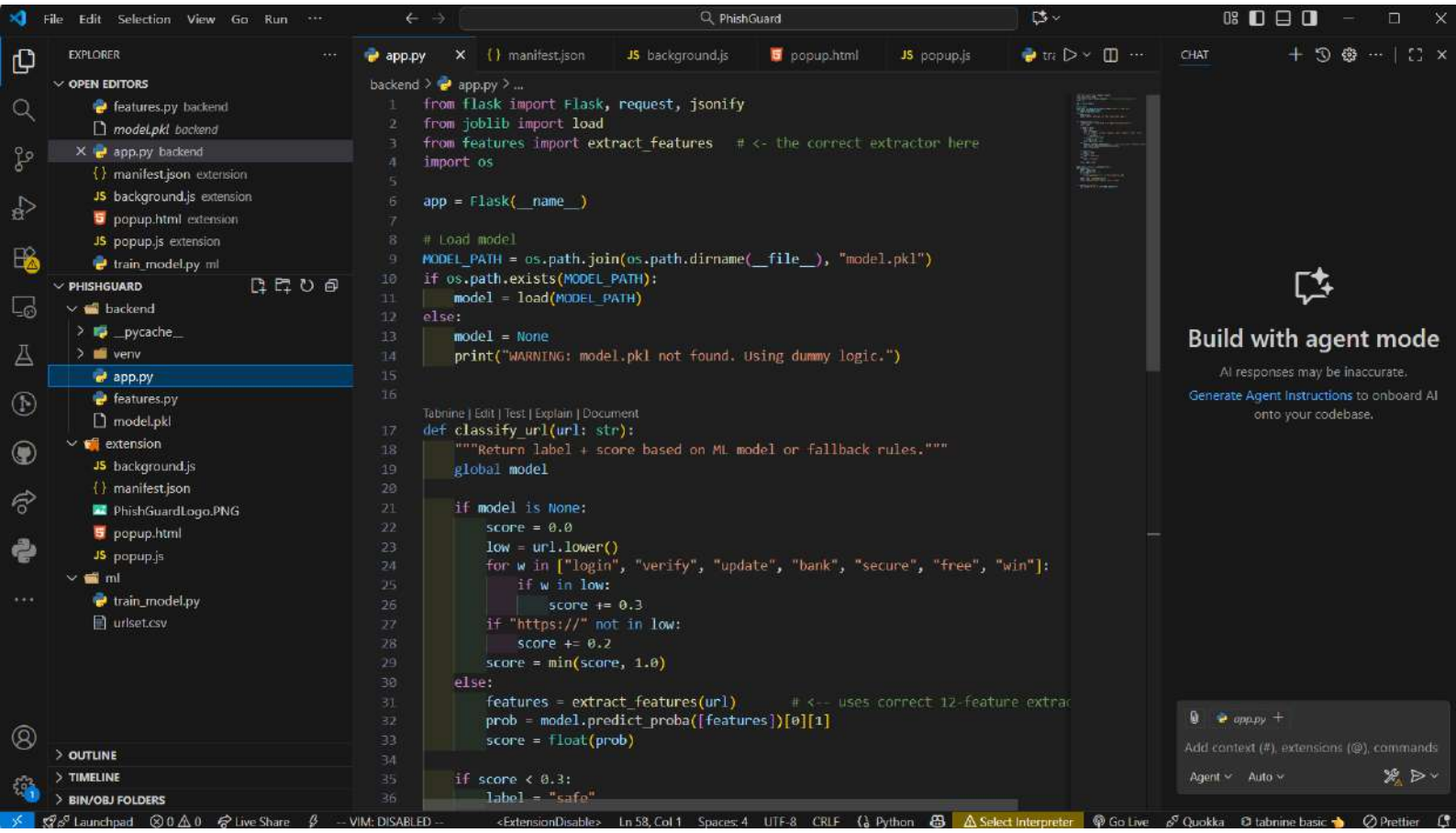
ID: fibkjooljpjcfddlpbmieceipabmeggl

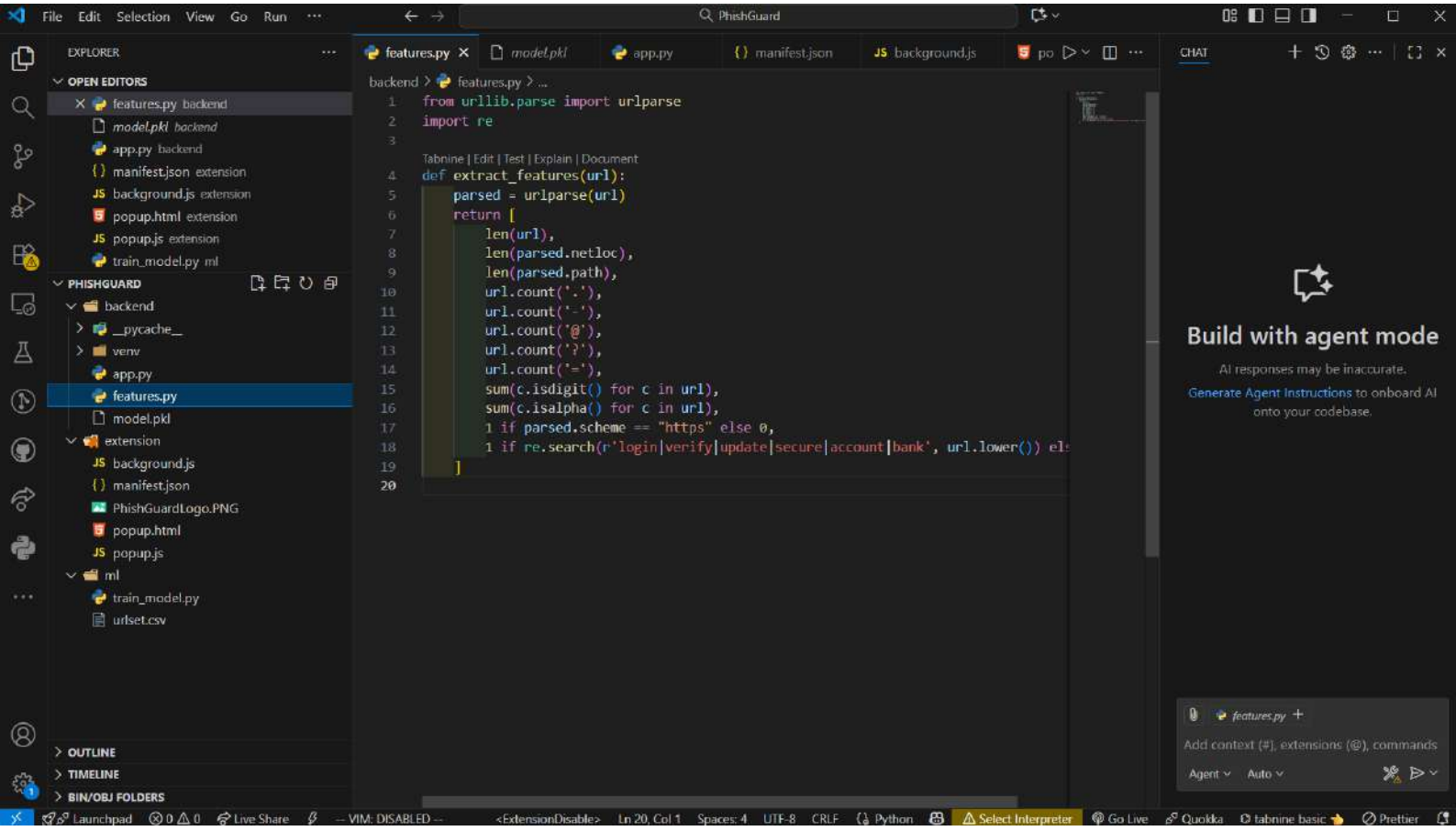
Inspect views service worker

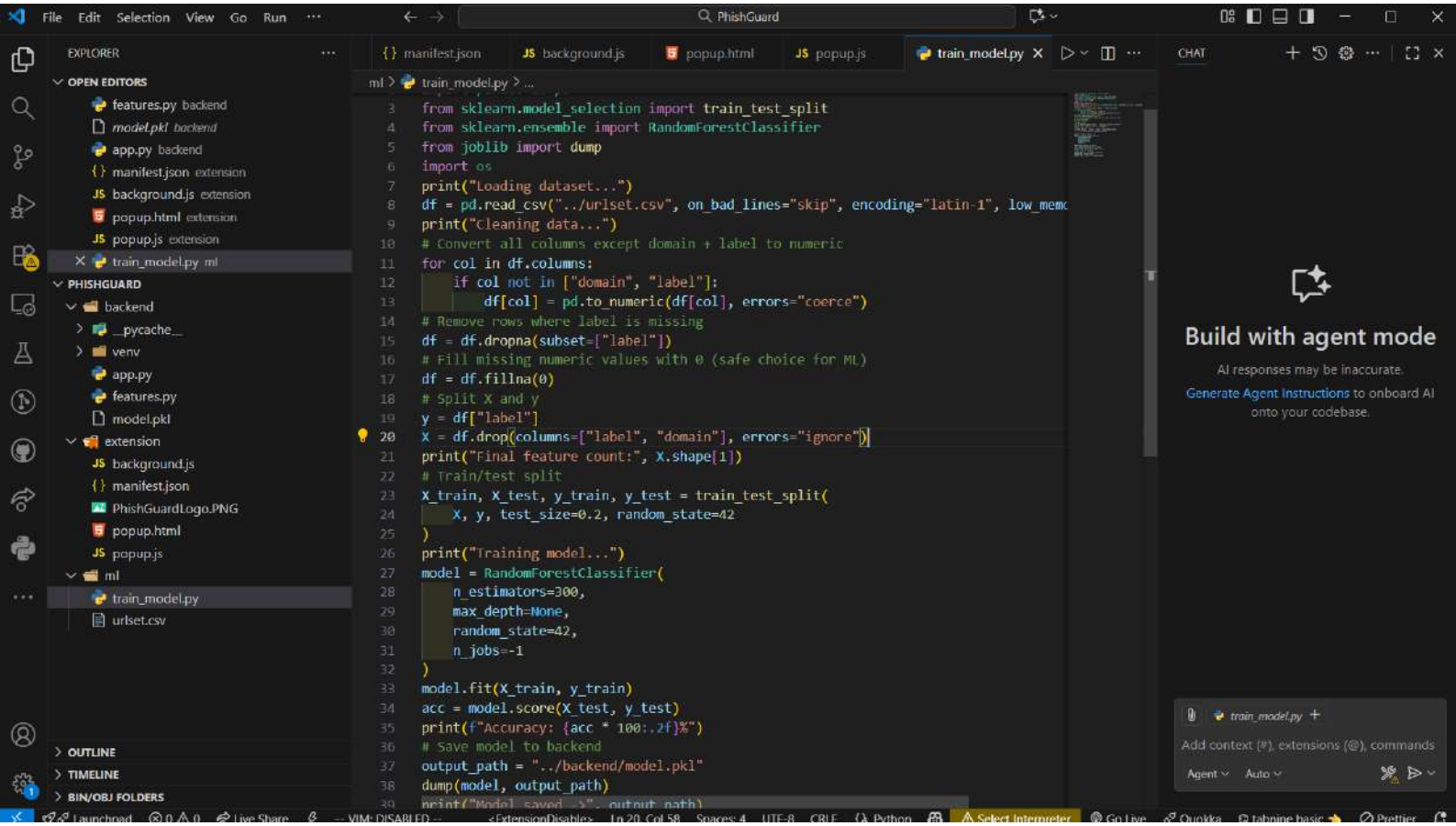
Details Remove



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
88	docs.goog	1	1	1	8	153.5	155.625	0.04211	0.044363	0.039474	0.04346	0.886969	0.790731	1			
89	'www.gest	10000000	0	0	12	38.91667	40.58333	0	0	0	0	0	0.870588	1			
90	elshadayel	10000000	0	0	3	19.66667	17	0	0	0	0	0	0.746032	1			
91	www.sisim	10000000	0	0	13	429.5385	516.8462	0	0	0	0	0	0.738694	1			
92	www.stacl	10000000	0	0	5	141	142.6	0	0	0	0	0	0.815621	1			
93	numberon	10000000	0	0	7	190.8571	198.4286	0	0	0	0	0	0.847458	1			
94	contact-ve	10000000	0	0	8	91.75	91.125	0	0	0	0	0	0.821918	1			
95	leuie.5gbfr	47798	0	0	2	179	181.5	0	0	0	0	0	0.811558	1			
96	ntcrulz.5gl	47798	0	0	0	0	0	0	0	0	0	0	0	1			
97	rsaxena.5g	47798	0	0	2	560	577.5	0	0	0	0	0	0.864754	1			
98	steamcom	51908	1	0	0	0	0	0	0	0	0	0.666667	0	1			
99	steamglfts	133305	1	0	0	0	0	0	0	0	0	0.9	0	1			
100	steamglfts	10000000	1	0	0	0	0	0	0	0	0	0	0	1			
101	vinzajah.5g	47798	0	0	2	592.5	623	0	0	0	0	0	0.867127	1			
102	vk-krasota	10000000	1	0	0	0	0	0	0	0	0	0	0	1			


File Edit Selection View Go Run
PhishGuard
EXPLORER
OPEN EDITORS
features.py backend
model.pkl backend
app.py backend
manifest.json extension
background.js extension
popup.html extension
popup.js extension
train_model.py ml
PHISHGUARD
backend
__pycache__
venv
app.py
features.py
model.pkl
extension
background.js
manifest.json
PhishGuardLogo.PNG
popup.html
popup.js
ml
train_model.py
urlset.csv
OUTLINE
TIMELINE
BIN/OBJ FOLDERS
app.py
manifest.json
background.js
popup.html
popup.js
backend > app.py > ...
from flask import Flask, request, jsonify
from joblib import load
from features import extract_features # <- the correct extractor here
import os
app = Flask(__name__)
# Load model
MODEL_PATH = os.path.join(os.path.dirname(__file__), "model.pkl")
if os.path.exists(MODEL_PATH):
model = load(MODEL_PATH)
else:
model = None
print("WARNING: model.pkl not found. Using dummy logic.")
Tabnine | Edit | Test | Explain | Document
def classify_url(url: str):
"""Return label + score based on ML model or fallback rules."""
global model
if model is None:
score = 0.0
low = url.lower()
for w in ["login", "verify", "update", "bank", "secure", "free", "win"]:
if w in low:
score += 0.3
if "https://" not in low:
score += 0.2
score = min(score, 1.0)
else:
features = extract_features(url) # <-- uses correct 12-feature extrac
prob = model.predict_proba([features])[0][1]
score = float(prob)
if score < 0.3:
label = "safe"
CHAT
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
app.py
Add context (#), extensions (@), commands
Agent
Auto
Launchpad
Live Share
-- VIM: DISABLED --
<ExtensionDisable>
Ln 58, Col 1
Spaces: 4
UTF-8
CRLF
Python
Select Interpreter
Go Live
Quokka
tabnine basic
Prettier


File Edit Selection View Go Run
PhishGuard
EXPLORER
OPEN EDITORS
features.py backend
model.pkl backend
app.py backend
manifest.json extension
background.js extension
popup.html extension
popup.js extension
train_model.py ml
PHISHGUARD
backend
__pycache__
venv
app.py
features.py
model.pkl
extension
background.js
manifest.json
PhishGuardLogo.PNG
popup.html
popup.js
ml
train_model.py
urlset.csv
OUTLINE
TIMELINE
BIN/OBJ FOLDERS
features.py
model.pkl
app.py
manifest.json
background.js
backend > features.py > ...
from urllib.parse import urlparse
import re
Tabnine | Edit | Test | Explain | Document
def extract_features(url):
parsed = urlparse(url)
return [
len(url),
len(parsed.netloc),
len(parsed.path),
url.count('.'),
url.count('-'),
url.count('@'),
url.count('?'),
url.count('='),
sum(c.isdigit() for c in url),
sum(c.isalpha() for c in url),
1 if parsed.scheme == "https" else 0,
1 if re.search(r'login|verify|update|secure|account|bank', url.lower()) el
]
CHAT
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
features.py
Add context (#), extensions (@), commands
Agent
Auto
Launchpad
Live Share
-- VIM: DISABLED --
<ExtensionDisable>
Ln 20, Col 1
Spaces: 4
UTF-8
CRLF
Python
Select Interpreter
Go Live
Quokka
tabnine basic
Prettier


File Edit Selection View Go Run
PhishGuard
EXPLORER
OPEN EDITORS
features.py backend
model.pkl backend
app.py backend
manifest.json extension
background.js extension
popup.html extension
popup.js extension
train_model.py ml
PHISHGUARD
backend
__pycache__
venv
app.py
features.py
model.pkl
extension
background.js
manifest.json
PhishGuardLogo.PNG
popup.html
popup.js
ml
train_model.py
urlset.csv
OUTLINE
TIMELINE
BIN/OBJ FOLDERS
manifest.json
background.js
popup.html
popup.js
train_model.py
ml > train_model.py > ...
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from joblib import dump
import os
print("Loading dataset...")
df = pd.read_csv("../urlset.csv", on_bad_lines="skip", encoding="latin-1", low_memo
print("Cleaning data...")
# Convert all columns except domain + label to numeric
for col in df.columns:
if col not in ["domain", "label"]:
df[col] = pd.to_numeric(df[col], errors="coerce")
# Remove rows where label is missing
df = df.dropna(subset=["label"])
# Fill missing numeric values with 0 (safe choice for ML)
df = df.fillna(0)
# Split X and y
y = df["label"]
X = df.drop(columns=["label", "domain"], errors="ignore")
print("Final feature count:", X.shape[1])
# Train/test split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=42
)
print("Training model...")
model = RandomForestClassifier(
n_estimators=300,
max_depth=None,
random_state=42,
n_jobs=-1
)
model.fit(X_train, y_train)
acc = model.score(X_test, y_test)
print(f"Accuracy: {acc * 100:.2f}%")
# Save model to backend
output_path = "../backend/model.pkl"
dump(model, output_path)
print("Model saved ->", output_path)
CHAT
Build with agent mode
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
train_model.py
Add context (#), extensions (@), commands
Agent
Auto
Launchpad
Live Share
-- VIM: DISABLED --
<ExtensionDisable>
Ln 20, Col 58
Spaces: 4
UTF-8
CRLF
Python
Select Interpreter
Go Live
Quokka
tabnine basic
Prettier

```
(venv) C:\Users\nitan\PhishGuard\backend>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.192.53.110:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 204-567-138
C:\Users\nitan\PhishGuard\backend\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have
valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
127.0.0.1 - - [23/Nov/2025 21:54:52] "POST /check_url HTTP/1.1" 200 -
C:\Users\nitan\PhishGuard\backend\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have
valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
127.0.0.1 - - [23/Nov/2025 21:55:01] "POST /check_url HTTP/1.1" 200 -
C:\Users\nitan\PhishGuard\backend\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have
valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
127.0.0.1 - - [23/Nov/2025 21:55:07] "POST /check_url HTTP/1.1" 200 -
C:\Users\nitan\PhishGuard\backend\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have
valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
127.0.0.1 - - [23/Nov/2025 21:55:11] "POST /check_url HTTP/1.1" 200 -
C:\Users\nitan\PhishGuard\backend\venv\Lib\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have
valid feature names, but RandomForestClassifier was fitted with feature names
```