

4IIR

Projet Fin D'année PFA 2025

Application web de détection des emails frauduleux et tentatives de phishing



Réalisé par :

AL ARHWANY Fatima Ezzahra
ENNADIRI Ichrak

Tuteur:

Encadrant pédagogique: Mr A.Bakhouyi

Année scolaire : 2024/2025

DÉDICACE

Nous dédions ce travail, avec une profonde gratitude, à nos chers parents, pour leur amour inconditionnel, leur patience, leur soutien indéfectible et la confiance qu'ils nous ont toujours témoignée tout au long de notre parcours.

À nos familles et à nos amis, dont les encouragements sincères, la bienveillance et la présence à nos côtés ont été une source précieuse de motivation.

À toutes les personnes, de près ou de loin, qui ont contribué, par un geste, un mot ou un regard, à l'accomplissement de ce projet : recevez ici l'expression de notre reconnaissance la plus sincère.

REMERCIEMENTS

Nous aimerions en premier lieu remercier notre dieu Allah qui nous a donné le courage pour la réalisation de ce travail.

Nous tenons tout d'abord à adresser nos plus vifs remerciements à notre encadrent Monsieur BAKHOUYI pour son encadrement, ses conseils ses corrections et sa disponibilité tout au long de ce projet..

Nous tenons également à remercier et à exprimer notre profond respect aux membres du jury d'avoir accepté de juger ce travail.

RÉSUMÉ

Ce projet de fin d'année présente le développement d'une application web complète de détection et d'analyse des emails frauduleux et tentatives de phishing. Dans un contexte où les cyberattaques deviennent de plus en plus sophistiquées, notre solution propose une approche innovante combinant des techniques d'analyse avancées avec une interface utilisateur moderne.

L'application a été développée en utilisant une architecture moderne basée sur Angular 17 pour le frontend et Spring Boot pour le backend, permettant une analyse en temps réel des emails suspects. La méthodologie de développement a suivi une approche agile, avec des phases d'analyse, de conception, de développement et de tests.

Les résultats obtenus démontrent une efficacité significative dans la détection des tentatives de phishing, avec un taux de détection élevé et un faible taux de faux positifs. L'application offre également des fonctionnalités avancées de génération de rapports et d'analyse statistique.

Mots-clés : Phishing, Cybersécurité, Angular, Spring Boot, Détection de fraudes, Analyse d'emails, Application web

ABSTRACT

This end-of-year project presents the development of a complete web application for the detection and analysis of fraudulent emails and phishing attempts. In a context where cyberattacks are becoming increasingly sophisticated, our solution offers an innovative approach that combines advanced analysis techniques with a modern user interface.

The application was developed using a modern architecture based on Angular 17 for the frontend and Spring Boot for the backend, enabling real-time analysis of suspicious emails. The development methodology followed an agile approach, with phases of analysis, design, development, and testing.

The results obtained demonstrate significant effectiveness in detecting phishing attempts, with a high detection rate and a low false positive rate. The application also offers advanced features for report generation and statistical analysis.

Keywords: Phishing, Cybersecurity, Angular, Spring Boot, Fraud Detection, Email Analysis, Web Application

Table des matières

Introduction générale	5
Chapitre1:.....	5
1. Contexte général	8
2. Problématique	8
3. Motivation.....	8
4. Étude de l'existant.....	9
4.1. Solutions commerciales	9
4.2. Solutions open-source	10
5. Cahier des charges	10
5.1. Analyse d'emails	10
5.2. Interface utilisateur	11
5.3. Authentification et sécurité	11
5.4. Conformité et traçabilité	11
Chapitre 2 :.....	10
1. Diagrammes UML.....	11
1.2. Diagramme de cas d'utilisation.....	11
1.3. Diagramme de classe.....	12
1.4. Diagramme de séquence.....	13
Chapitre 3:.....	14
1. Vue d'ensemble technique	17
2. Technologies et outils utilisés	17
3. Fonctionnement global de l'application.....	20
4. Structure du projet.....	20
5. Architecture Fonctionnelle de l'Application	22
5.1. Implementation	23
6. Valorisation de l'IA et du Machine Learning	28
7. Résultat attendu.....	28
Chapitre 4 :.....	27
1. Planning et organisation.....	30
1.2. Découpage en phases	30
1.3. Gestion des risques	30
2. Suivi et contrôle	31
2.1. Métriques de suivi.....	31
2.2. Tests et validation	31
3. Bilan de la réalisation	32
3.1. Objectifs atteints	32

3.2. Points d'amélioration	32
Conclusion générale	33

Liste des figures

Figure 1: diagramme de cas d'utilisation	13
Figure 2:Diagramme de séquence.....	14
Figure 3:Diagramme de classes	15
Figure 4:Structure Frontend	21
Figure 5:Structure Backend	21
Figure 6: Schéma de la Base de Données Anti-Phishing.....	22
Figure 7: Architecture fonctionnelle.....	22
Figure 8:Page d'Accueil	23
Figure 9: Page d'inscription	23
Figure 10: Page de connexion.....	24
Figure 11:Page d'analyse d'email.....	24
Figure 12: Résultat d'analyse	25
Figure 13:Rapport généré	26
Figure 14: Historique des analyses	26
Figure 15:Détails de l' analyse	27
Figure 16:Questionnaire Anti-Phishing	27

Introduction générale

Dans un monde numérique en constante évolution, la sécurité des communications par email est devenue une préoccupation majeure. Les attaques par phishing représentent une menace croissante, avec des techniques de plus en plus sophistiquées visant à tromper les utilisateurs et à compromettre la sécurité des données.

Cette étude présente un intérêt majeur à plusieurs niveaux :

- Sécurité : Réduction des risques de compromission des données
- Innovation : Développement de nouvelles techniques de détection
- Formation : Application pratique des connaissances en cybersécurité
- Utilité : Solution concrète

La détection efficace des emails frauduleux nécessite une approche sophistiquée combinant analyse technique et compréhension des techniques de manipulation psychologique utilisées par les attaquants.

Comment développer une application web efficace pour la détection et l'analyse des emails frauduleux tout en offrant une expérience utilisateur optimale ?

Quelles sont les techniques les plus efficaces pour détecter les tentatives de phishing ?

Comment concevoir une interface utilisateur intuitive pour l'analyse des emails ?

Quelles mesures de sécurité mettre en place pour protéger les données des utilisateurs ?

Comment assurer la scalabilité et la performance de l'application ?

L'objectif principal est de développer une application web complète capable de détecter en temps réel les emails frauduleux, d'analyser en profondeur le contenu et les liens suspects, de générer des rapports détaillés, tout en assurant la protection des données des utilisateurs.

La méthodologie adoptée pour ce projet s'est articulée autour de quatre phases principales:

La première, la phase d'analyse, a consisté en l'étude des besoins, l'analyse des solutions existantes, ainsi que la définition des spécifications fonctionnelles et techniques.

La deuxième phase, celle de la conception, a permis de définir l'architecture du système, de concevoir l'interface utilisateur et de modéliser les données.

La phase de développement s'est ensuite déroulée en deux volets : le développement du frontend avec Angular et du backend avec Spring Boot, suivi de l'intégration des différents composants.

Enfin, la phase de tests a inclus des tests unitaires, des tests d'intégration et des tests de performance afin de garantir la fiabilité et l'efficacité de l'application.

Ce rapport est structuré en dix chapitres principaux. Après cette introduction, nous présenterons l'architecture du système, suivie des fonctionnalités principales et des aspects techniques. Nous détaillerons ensuite l'interface utilisateur, les mesures de sécurité, les tests et la validation, ainsi que le déploiement et la maintenance. Le rapport se conclura par une conclusion générale et des perspectives d'évolution, suivies des annexes techniques.

CHAPITRE 1 :

CADRE GÉNÉRAL DU PROJET

1. Contexte général

Avec l'explosion des usages numériques et la démocratisation des services en ligne, les communications électroniques – en particulier les emails – sont devenues un vecteur central d'interactions personnelles et professionnelles. Ce canal, aussi indispensable qu'efficace, est toutefois exploité par des acteurs malveillants qui y voient une opportunité facile et rentable pour mener des attaques de phishing. Ces dernières, souvent difficiles à repérer pour l'utilisateur lambda, visent à dérober des informations sensibles à des fins frauduleuses.

Dans ce contexte de cybersécurité critique, il devient impératif de mettre en place des solutions automatisées, intelligentes et accessibles pour renforcer la détection de ces menaces. Ce projet s'inscrit précisément dans cette dynamique, en combinant plusieurs approches pour analyser et détecter les courriels frauduleux à l'aide des technologies modernes.

2. Problématique

L'augmentation du volume d'emails échangés quotidiennement, combinée à la sophistication croissante des attaques de phishing, complexifie considérablement leur détection. Les méthodes classiques, reposant sur des règles statiques ou des listes noires, s'avèrent insuffisantes face aux techniques évolutives d'usurpation et de falsification.

Ainsi, la problématique principale à laquelle ce projet tente de répondre est la suivante :

Comment développer une application intelligente capable d'analyser les courriels afin de détecter efficacement les tentatives de phishing, tout en assurant une expérience utilisateur simple et intuitive ?

3. Motivation

La réalisation de ce projet s'appuie sur plusieurs éléments clés qui soulignent l'urgence et l'intérêt d'une telle démarche :

Une croissance alarmante du phishing

Les rapports en cybersécurité indiquent une hausse constante des campagnes de phishing, qui ciblent aussi bien les entreprises que les particuliers. Ces attaques entraînent des pertes économiques considérables, des violations de données sensibles et un impact négatif sur la réputation des victimes.

Des solutions existantes parfois limitées

Les outils de sécurité actuels ne parviennent pas toujours à identifier les menaces les plus récentes. L'évolution rapide des techniques employées nécessite des mécanismes de détection plus intelligents, adaptatifs et performants.

La protection des données, une exigence réglementaire

La législation, notamment à travers le Règlement Général sur la Protection des Données (RGPD), impose des standards élevés en matière de sécurité de l'information. Toute organisation se doit de garantir la confidentialité et l'intégrité des données traitées.

L'importance d'une interface accessible

Pour qu'une solution soit véritablement adoptée par les utilisateurs, elle doit être simple à utiliser. Une interface claire, intuitive et bien pensée favorise l'appropriation de l'outil, même par des personnes sans expertise technique.

Ces constats motivent la conception d'une application web combinant performance, intelligence et convivialité. L'objectif est d'offrir un outil capable de renforcer la sécurité des communications électroniques, tout en restant accessible à un large public.

4. Étude de l'existant

L'analyse des solutions actuelles de détection de phishing permet de situer notre projet dans son contexte technologique et d'identifier les axes d'innovation possibles. On distingue deux grandes catégories : les solutions commerciales et les solutions open-source.

4.1. Solutions commerciales

Microsoft Defender for Office 365

Offre une protection avancée contre les menaces (phishing, malwares, attaques ciblées), avec des technologies telles que l'IA, le sandboxing et la vérification des liens en temps réel. Cependant, sa configuration complexe et son coût peuvent être des freins pour les petites structures.

Google Workspace Security

Utilise le machine learning pour détecter les spams et tentatives de phishing. Bien que puissant grâce à sa base de données constamment mise à jour, ce service reste limité en personnalisation pour les cas spécifiques.

Proofpoint

L'une des solutions les plus complètes, notamment pour la détection de BEC (Business Email Compromise). Elle repose sur une analyse comportementale et des filtres dynamiques. Toutefois, elle est coûteuse et s'adresse principalement aux grandes entreprises.

4.2. Solutions open-source

SpamAssassin

Basé sur des règles heuristiques et des filtres bayésiens. Bien qu'il soit personnalisable, il nécessite des ajustements pour être efficace contre les attaques de phishing récentes.

Rspamd

Plus moderne et performant, avec une architecture modulaire et un support de l'apprentissage automatique. Il s'intègre bien aux systèmes existants mais exige une certaine expertise technique.

MailScanner

Agit comme un framework combinant divers outils antivirus et antispam. Il est flexible mais présente une courbe d'apprentissage importante, ce qui peut limiter son adoption.

5. Cahier des charges

Ce projet vise à développer une application web intuitive et sécurisée, capable de détecter les emails frauduleux. Les fonctionnalités sont regroupées en quatre grandes catégories : analyse d'emails, interface utilisateur, gestion de la sécurité, et formation.

5.1. Analyse d'emails

Soumission et traitement

Interface dédiée à la soumission (copier-coller, validation des formats)

Analyse automatique du contenu textuel, des en-têtes, et détection des patterns suspects

Résultats d'analyse

Affichage structuré des menaces détectées avec explication des risques
Suggestions de bonnes pratiques et score de risque global

5.2. Interface utilisateur

Page d'accueil

Présentation de l'outil, accès rapide aux fonctions clés, statistiques d'utilisation

Page d'analyse

Zone de saisie, boutons d'action, progression en temps réel, historique des analyses

Page de rapports

Accès aux analyses précédentes avec filtres, tri, export et archivage

Page de formation

Contenus pédagogiques : exemples de phishing, bonnes pratiques, quiz, cas pratiques

5.3. Authentification et sécurité

Connexion et inscription

Interfaces sécurisées avec validation, gestion des erreurs, confirmation par email

Gestion des sessions

Déconnexion automatique, mémorisation sécurisée, journalisation des connexions

Protection des données

Chiffrement des informations sensibles, protection contre les injections, conformité RGPD

5.4. Conformité et traçabilité

Application des normes de sécurité

Historique des actions utilisateur

Respect de la confidentialité et de la politique de données

CHAPITRE 2 :

ANALYSE ET CONCEPTION

1. Diagrammes UML

Dans le cadre du projet d'analyse et de sécurisation d'emails, nous avons conçu trois types de diagrammes UML pour décrire le fonctionnement global du système :

1.2. Diagramme de cas d'utilisation :

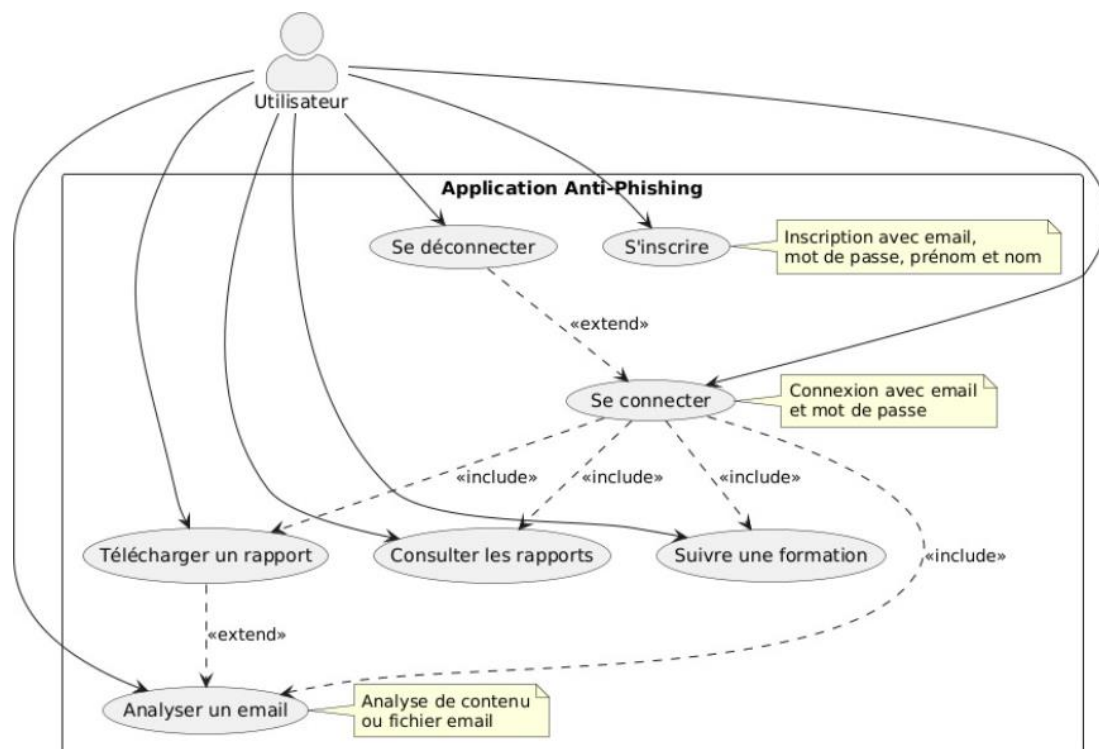


Figure 1: diagramme de cas d'utilisation

Ce diagramme illustre l'ensemble des fonctionnalités disponibles pour les utilisateurs de l'application anti-phishing. Il met en évidence les principales interactions telles que l'authentification, l'analyse des e-mails, la gestion des quiz, la consultation de l'historique, la génération de rapports au format PDF. Ce diagramme offre une vue d'ensemble claire des services proposés par l'application.

1.3. Diagramme de séquence : Processus d'analyse d'email:

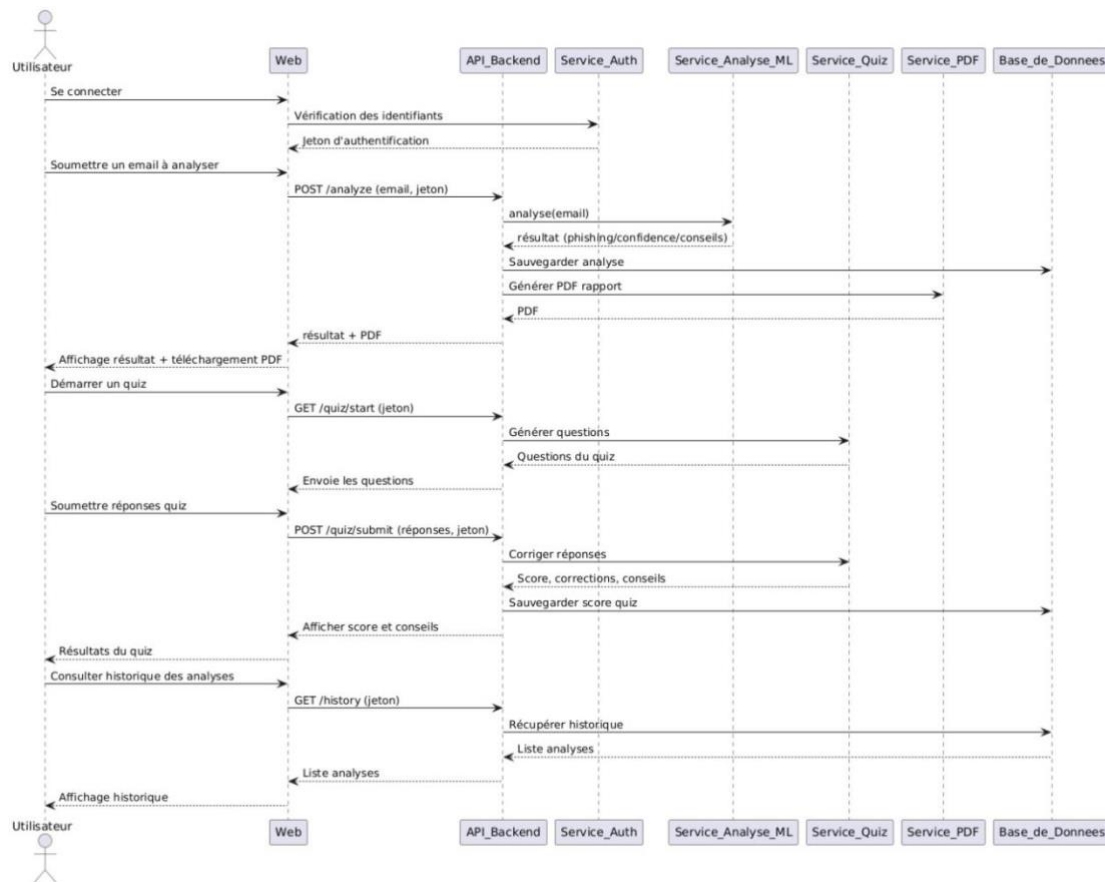


Figure 2:Diagramme de séquence

Ce diagramme décrit le déroulement des échanges entre les différents composants de l'application lors de l'utilisation de ses principales fonctionnalités. Il illustre les étapes clés, notamment l'authentification de l'utilisateur, l'analyse d'un e-mail (incluant la génération d'un rapport PDF et la sauvegarde de l'historique), la participation à un quiz et la consultation de l'historique. Le diagramme met en évidence la communication entre l'interface web, l'API backend, les services spécialisés (authentification, machine learning, quiz, génération de PDF) et la base de données. Il permet ainsi de visualiser le flux d'informations et la coordination entre les modules techniques de l'application.

1.4. Diagramme de classes :

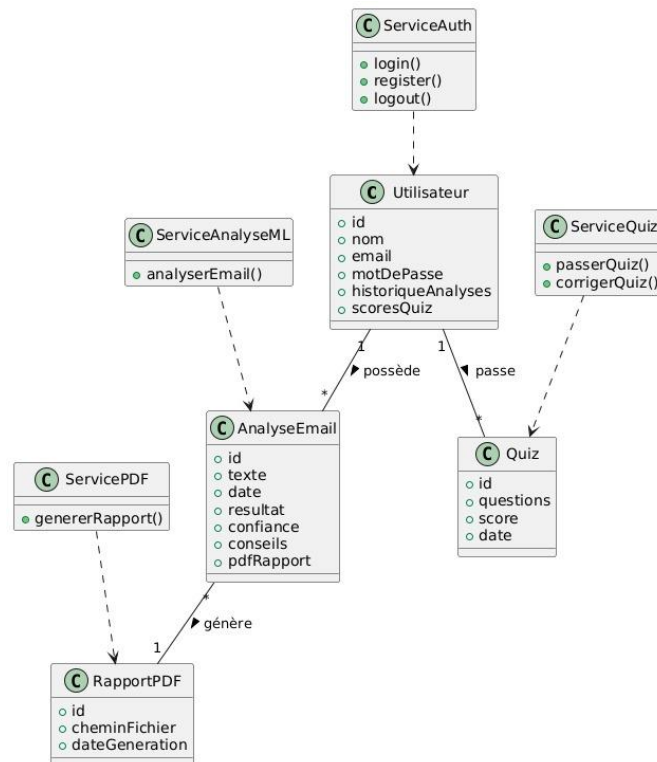


Figure 3:Diagramme de classes

Ce diagramme illustre l'architecture interne de l'application. Il présente les principales entités (Utilisateur, AnalyseEmail, Quiz, RapportPDF) ainsi que les services associés (ServiceAuth, ServiceAnalyseML, ServiceQuiz, ServicePDF). Les relations entre ces classes soulignent la modularité du système et l'organisation du code, ce qui facilite la maintenance, les tests et les évolutions futures de l'application. Ce diagramme est un support essentiel pour comprendre la structure logicielle et les responsabilités attribuées à chaque composant.

CHAPITRE 3 :

RÉALISATION

1. Vue d'ensemble technique

Le projet repose sur une architecture modulaire combinant des technologies modernes issues du développement web, de l'intelligence artificielle et du Machine Learning, afin de détecter automatiquement les tentatives de phishing dans les emails soumis par les utilisateurs. Chaque composant de l'application a été choisi pour répondre à des objectifs précis : efficacité, robustesse, maintenabilité, et valorisation de l'intelligence artificielle.

2. Technologies et outils utilisés



Python

Rôle : Développement du backend IA/ML.

Fonction : Traitement des requêtes d'analyse et exécution du modèle de Machine Learning pour identifier les emails frauduleux.

Justification : Python est la référence en IA/ML grâce à ses bibliothèques puissantes, sa simplicité d'écriture et sa large communauté.



Flask

Rôle : Micro-framework Python pour le web.

Fonction : Fournit une API REST permettant la communication entre le backend principal (Spring Boot) et le modèle ML.

Justification : Léger, rapide à déployer, idéal pour servir des modèles IA sous forme de microservices.



✓ Scikit-learn

Rôle : Bibliothèque de Machine Learning.

Fonction : Prétraitement des textes (via TfidfVectorizer), apprentissage automatique (modèle Naive Bayes), et inférence.

Justification : Référence dans le domaine, elle facilite le développement rapide de modèles fiables.



✓ Joblib

Rôle : Outil de sérialisation.

Fonction : Sauvegarde et rechargement du modèle IA entraîné, pour une intégration fluide en production.

Justification : Optimisée pour les objets lourds comme les modèles de ML.



✓ Java & Spring Boot

Rôle : Backend principal de l'application.

Fonction : Gestion des utilisateurs, sécurité, orchestration des analyses, gestion des rapports, services REST.

Justification : Plateforme robuste et adaptée aux systèmes d'entreprise, elle assure une base solide et évolutive.

APACHE PDFBox



✓ Apache PDFBox

Rôle : Génération de rapports PDF.

Fonction : Création de rapports d'analyse personnalisés, clairs et imprimables pour les utilisateurs.

Justification : Outil professionnel open source dédié à la manipulation de fichiers PDF.



✓ Angular

Rôle : Développement du frontend.

Fonction : Interface utilisateur, soumission des emails, visualisation des résultats, export des rapports.

Justification : Framework moderne, performant et structuré pour les applications web interactives.



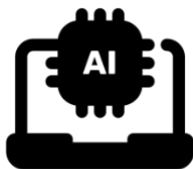
Bootstrap

✓ Bootstrap

Rôle : Framework CSS.

Fonction : Mise en forme responsive et esthétique de l'interface utilisateur.

Justification : Permet de créer des interfaces harmonieuses et compatibles avec tous les appareils.



✓ Machine Learning & IA

Rôle : Cœur de l'innovation du projet.

Fonction : Analyse automatique du contenu textuel, détection intelligente de phishing, extraction de caractéristiques pertinentes.

Justification : Apporte une valeur ajoutée forte en automatisant un processus critique de cybersécurité.

3. Fonctionnement global de l'application

L'application s'organise selon le flux suivant :

Soumission de l'email :

L'utilisateur colle ou téléverse un email via l'interface Angular.

Traitement côté backend (Spring Boot) :

Appel à des règles heuristiques internes.

Requête HTTP vers l'API Python/Flask pour l'analyse IA.

Analyse IA (Python/Flask) :

Prétraitement du texte.

Prédiction avec le modèle Naive Bayes.

Renvoi du résultat (phishing ou non) avec un score de confiance.

Restitution des résultats :

Affichage immédiat dans l'interface utilisateur Angular.

Option d'**export au format PDF** (via PDFBox) ou **CSV** pour l'archivage ou l'analyse manuelle.

Interface utilisateur :

Conçue avec Angular et Bootstrap, elle offre une expérience fluide et intuitive.

Accès à l'historique des analyses, aux rapports, et à une navigation ergonomique.

4. Structure du projet

L'application suit une architecture en trois couches principales :

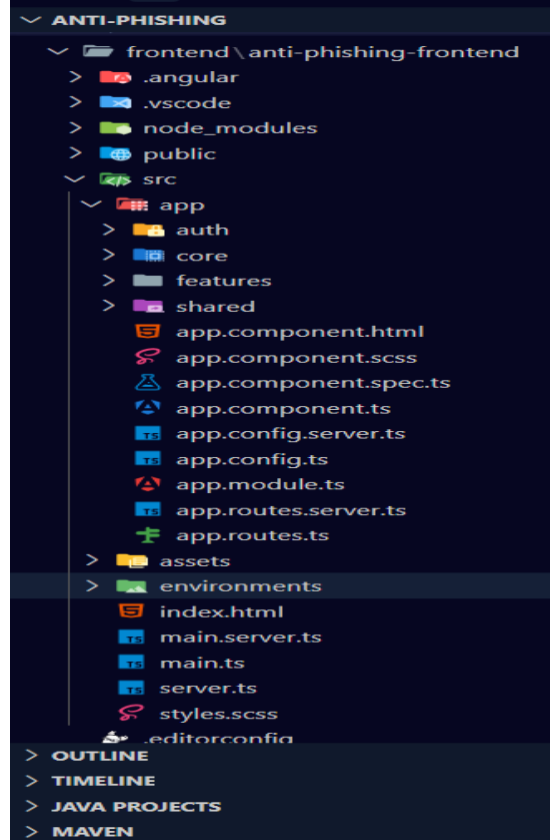


Figure 4: Structure Frontend

Frontend (Angular) : interface utilisateur, formulaires de soumission, affichage des résultats, visualisation des rapports.

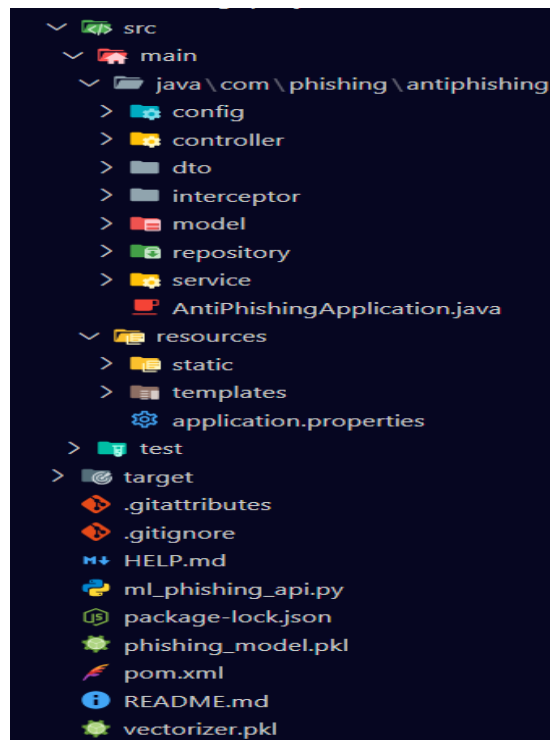


Figure 5: Structure Backend

Frontend (Angular) : interface utilisateur, formulaires de soumission, affichage des résultats, visualisation des rapports.

Base de données (MySQL) :

The screenshot shows the HeidiSQL interface for the 'anti-phishing' database. The left sidebar displays the database structure with tables: email_analyses, email_attachments, email_links, users, user_roles, information_schema, mysql, performance_schema, phpmadmin, and test. The main window shows the 'email_analyses' table with columns: id, analysis_details, content, and created_at. The table contains 11 rows of data, all with 'Email Analysis Report' as the analysis_details and various timestamps as created_at. The bottom status bar indicates the connection to MariaDB 10.4.32.

#	id	analysis_details	content	created_at
1	1	Email Analysis Report=====	De: "DRH Entreprise" <rh@entreprise-legit.com>	2025-05-11 22:48:23.000000
2	2	Email Analysis Report=====	noreply@globalbank-phishing.com Urgent: AL...	2025-05-11 23:07:04.000000
3	3	Email Analysis Report=====	Cher Client, Nous avons détecté une activité sus...	2025-05-11 23:12:00.000000
4	4	Email Analysis Report=====	Bonjour, Suite à une activité suspecte détectée ...	2025-05-11 23:19:01.000000
5	5	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-11 23:26:17.000000
6	6	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-11 23:32:42.000000
7	7	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-11 23:36:37.000000
8	8	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-11 23:38:31.000000
9	9	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-12 00:32:29.000000
10	10	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-12 00:38:31.000000
11	11	Email Analysis Report=====	Bonjour, Votre compte a été temporairement su...	2025-05-12 21:36:01.000000

Figure 6: Schéma de la Base de Données Anti-Phishing

Base principale : antiphishing

Tables clés :

- email_analyses (résultats)
- email_attachments & email_links (éléments analysés)
- users & user_roles (gestion des comptes)

5. Architecture Fonctionnelle de l'Application

Le fonctionnement global repose sur une collaboration fluide entre les différents composants du système :

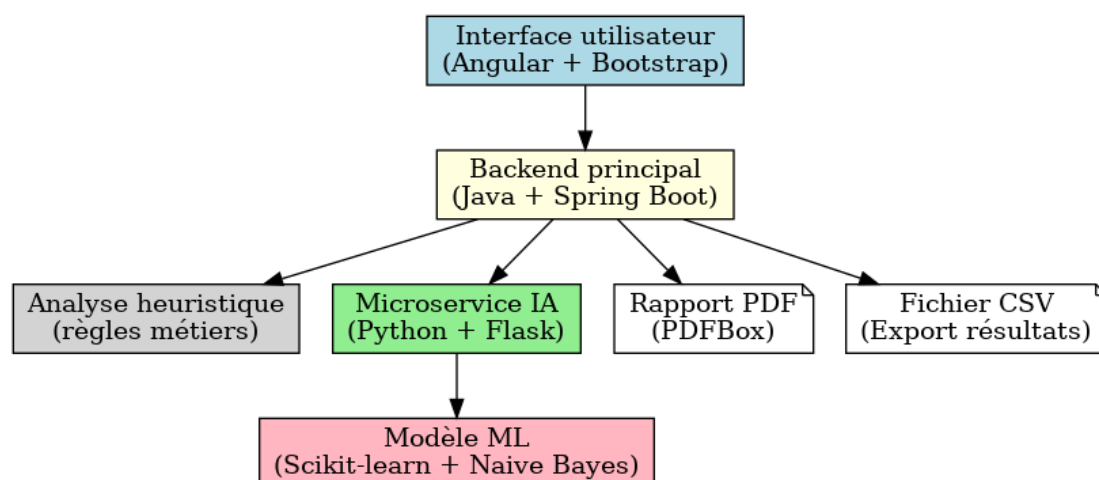


Figure 7: Architecture fonctionnelle

5.1. Implementation

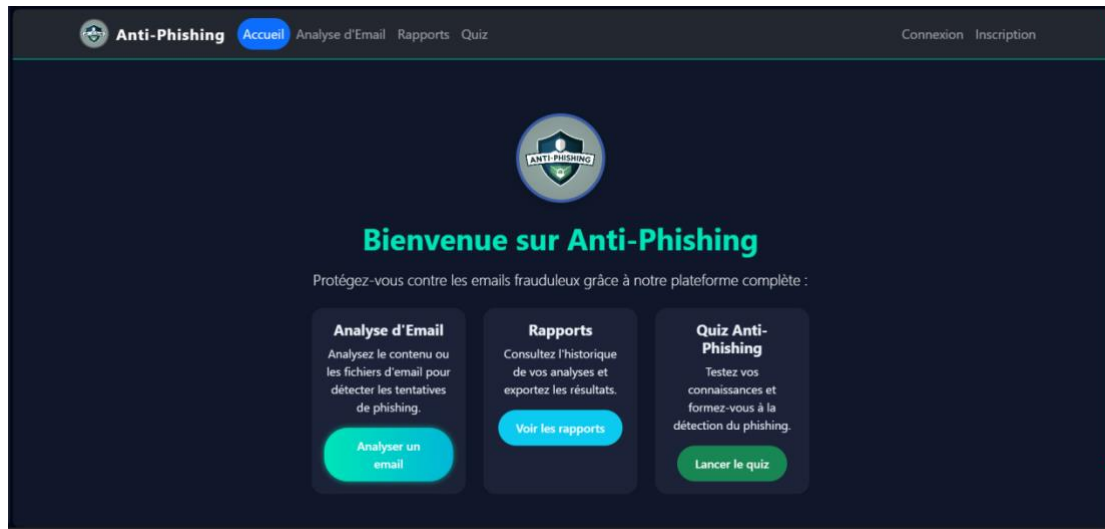


Figure 8: Page d'Accueil

Cette interface est la page d'accueil d'une plateforme anti-phishing **réservée aux utilisateurs connectés**. Elle offre trois fonctionnalités principales :

Analyse d'emails pour détecter les tentatives de phishing

Consultation des rapports d'analyses précédentes

Quiz éducatif pour se former à la détection des fraudes

Toutes ces fonctionnalités nécessitent une connexion préalable pour y accéder.

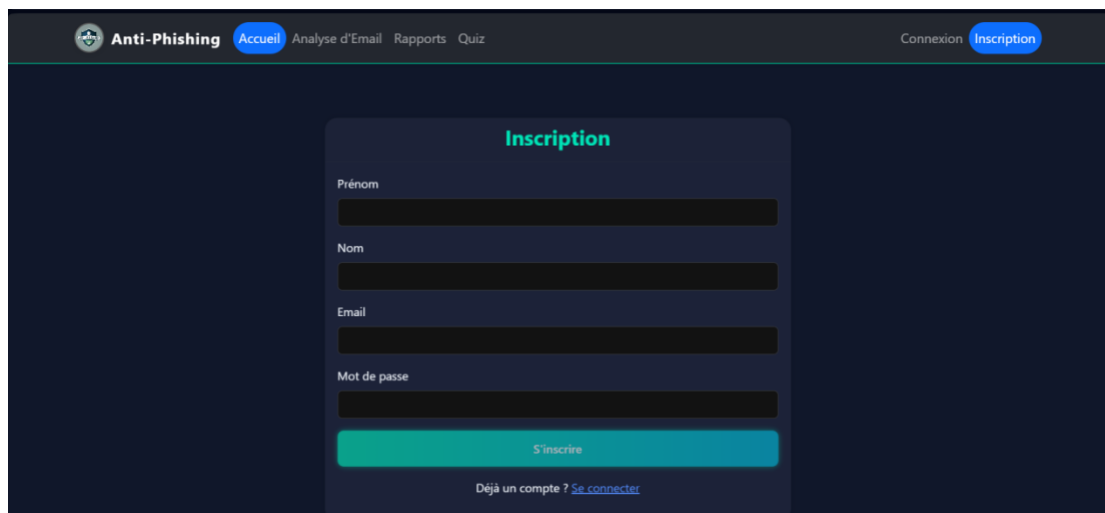


Figure 9: Page d'inscription

Cette interface présente un formulaire d'inscription pour accéder aux fonctionnalités anti-phishing (analyse d'emails, rapports et quiz), avec un menu de navigation en haut. L'accès à ces outils nécessite obligatoirement une connexion ou une inscription préalable.

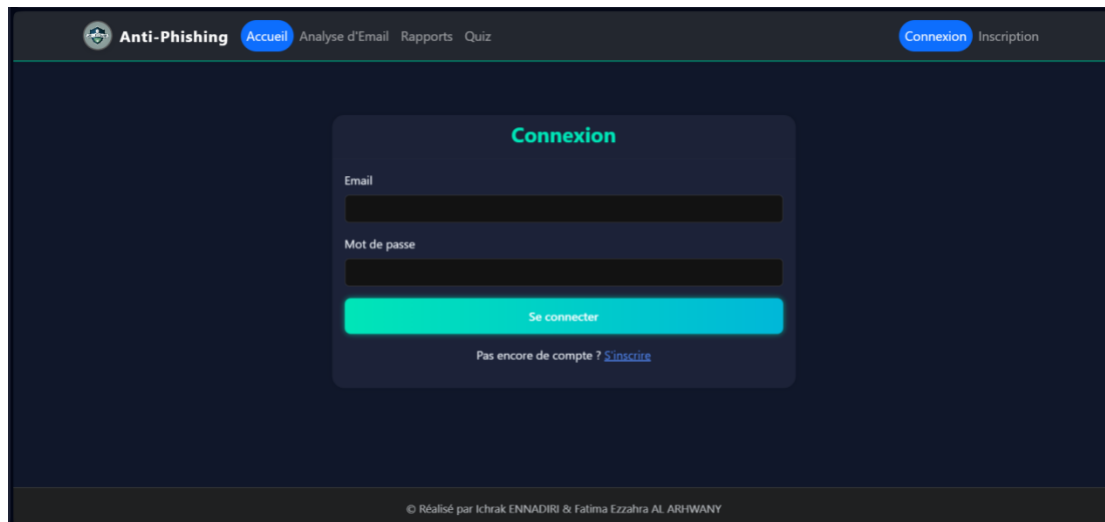


Figure 10: Page de connexion

Cette interface est une **page de connexion** à la plateforme Anti-Phishing, permettant aux utilisateurs de s'authentifier pour accéder aux fonctionnalités (analyse d'emails, rapports et quiz). Elle propose également un lien vers l'inscription pour les nouveaux utilisateurs, avec une mention des créateurs en bas de page.

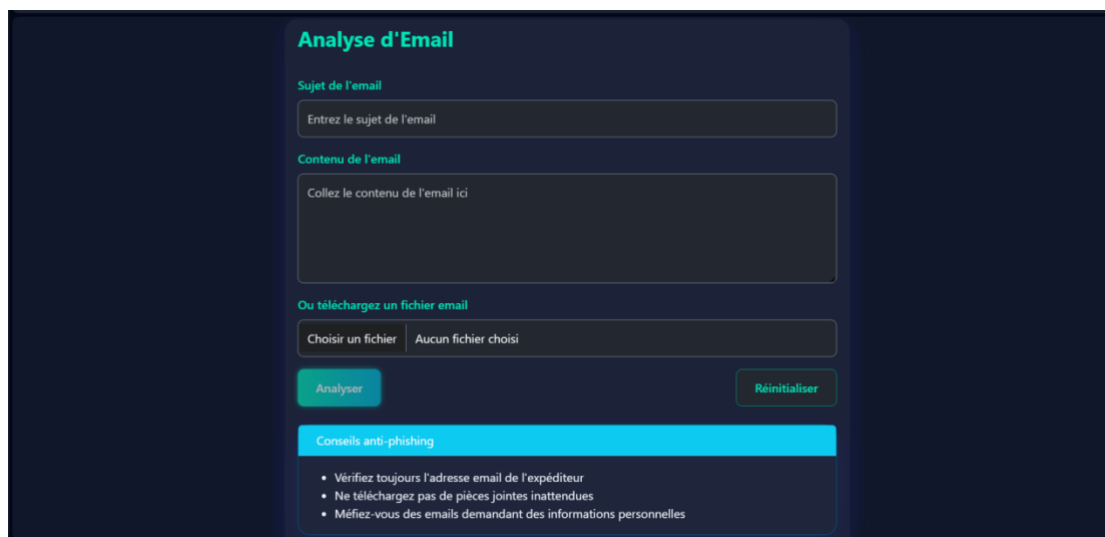


Figure 11: Page d'analyse d'email

Page d'analyse d'email permettant de vérifier les emails suspects par saisie manuelle ou import de fichier, avec options d'analyse et conseils de sécurité. Accès réservé aux utilisateurs connectés.

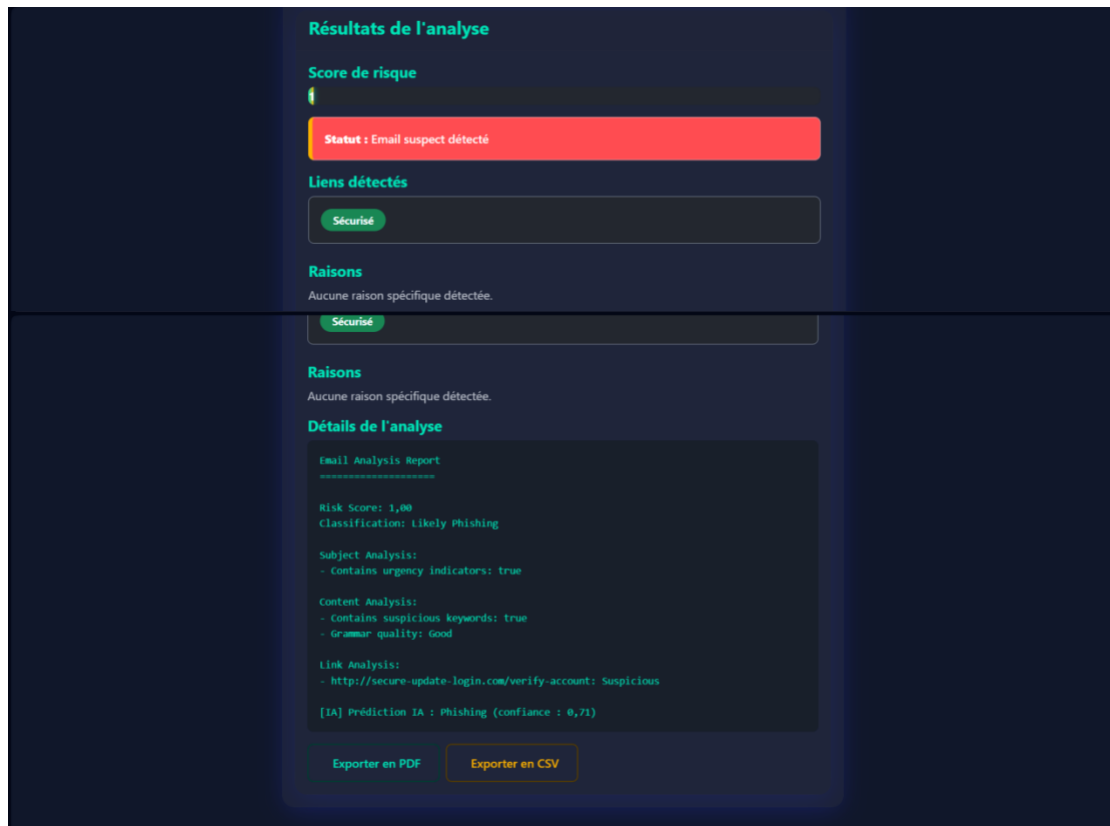


Figure 12: Résultat d'analyse

Synthèse et retour à l'utilisateur : Les résultats sont affichés dans l'interface avec options d'export PDF/CSV.

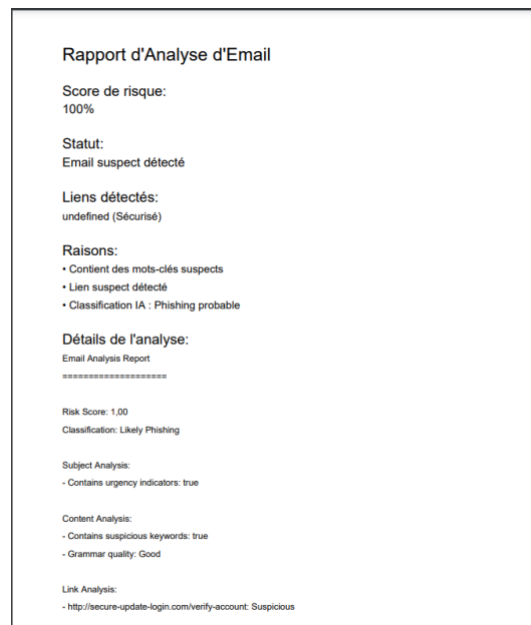



Figure 13: Rapport généré

Rapport generer suite à la demande de l'utilisateur.  génère une **analyse détaillée des risques** (score, liens suspects, mots-clés) et des **recommandations d'action** (ne pas cliquer, supprimer) pour se protéger face à un email identifié comme phishing.

Date	Sujet	Risque	Score	Actions
20/05/2025 23:19	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
19/05/2025 18:32	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 22:42	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 22:37	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 22:34	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 22:00	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 21:58	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details
18/05/2025 21:45	▲ URGENT : Votre compte sera supprimé dans 24h ! ▲	Phishing	1%	Details

Figure 14: Historique des analyses

Archivage & Historique : L'utilisateur peut consulter les analyses précédentes des emails analysés.

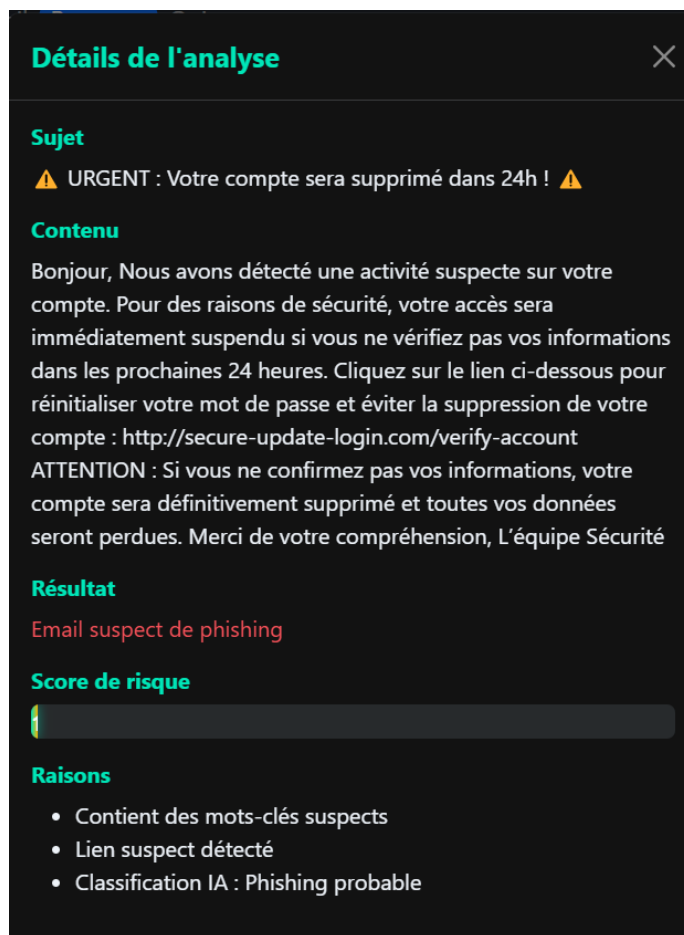


Figure 15: Détails de l'analyse

Cette page sert à **identifier et expliquer** les emails de phishing en analysant leur contenu (texte, liens, ton), puis fournit des **recommandations concrètes** pour se protéger.



Figure 16: Questionnaire Anti-Phishing

Quiz et formation : L'utilisateur peut passer les quiz et s'informer sur le monde du phishing

6. Valorisation de l'IA et du Machine Learning

L'originalité du projet repose principalement sur l'intégration intelligente de l'IA :

Automatisation : L'analyse se fait en temps réel sans intervention humaine.

Explicabilité : Un score de confiance et des indices d'analyse sont restitués pour aider l'utilisateur à comprendre la menace.

Interopérabilité : Le modèle ML, développé indépendamment en Python, s'intègre parfaitement avec le backend Java, démontrant la capacité du projet à assembler des technologies hétérogènes.

Renforcement par heuristiques : L'IA est augmentée de règles métiers, assurant une détection plus fine et réaliste.

7. Résultat attendu

Grâce à cette architecture, l'application permet une détection fiable, rapide et explicable des emails frauduleux, tout en offrant à l'utilisateur des outils d'exploitation des résultats (PDF, CSV), le tout dans une interface moderne, claire et sécurisée.

CHAPITRE 4 :

SUIVI DE LA RÉALISATION

1. Planning et organisation

1.2. Découpage en phases

Le projet a été structuré en plusieurs phases successives afin de garantir une progression méthodique et contrôlée :

Phase d'analyse et de conception

Cette phase a permis de poser les fondations du projet :

Étude des besoins : Recueil et analyse des attentes des utilisateurs et des contraintes techniques.

Conception de l'architecture : Définition des composants du système, choix des technologies et structuration générale de l'application.

Spécifications fonctionnelles et techniques : Rédaction d'un document détaillant les fonctionnalités, les interactions entre modules, et les exigences de performance et de sécurité.

Phase de développement

Cette étape a consisté à implémenter les fonctionnalités prévues :

Développement frontend : Réalisation de l'interface utilisateur en veillant à l'ergonomie et à l'accessibilité.

Développement backend : Mise en place des traitements serveurs, de la base de données et des API nécessaires.

Intégration : Assemblage des différentes couches de l'application pour obtenir une solution cohérente et opérationnelle.

Phase de tests

Des tests ont été effectués à chaque étape pour garantir la qualité du produit final :

Tests unitaires : Vérification du bon fonctionnement de chaque module indépendamment.

1.3. Gestion des risques

L'identification et la gestion proactive des risques ont constitué un axe essentiel de la réussite du projet.

Principaux risques identifiés

Risques techniques : Liés à la complexité du développement ou à des choix technologiques inadéquats.

Complexité de l'analyse des emails : Difficulté à identifier avec précision les courriels frauduleux sans générer de faux positifs.

Performance du système : Risque que l'application ne réponde pas aux exigences en termes de rapidité et de fluidité.

Sécurité des données : Risques d'exploitation de failles de sécurité ou de perte d'informations sensibles.

Mesures préventives mises en œuvre

Réalisation de **tests réguliers** tout au long du développement.

Rédaction d'une **documentation continue** pour assurer la traçabilité et la compréhension du code.

Mise en place de **revues de code** entre développeurs pour détecter les anomalies précocement.

Suivi régulier des performances via des outils de **monitoring**.

2. Suivi et contrôle

2.1. Métriques de suivi

Pour garantir le bon déroulement du projet et prendre des décisions éclairées, plusieurs indicateurs clés ont été utilisés :

Indicateurs de développement

Taux d'avancement global : Pourcentage de tâches réalisées par rapport au total.

Taux de complétion des fonctionnalités : Fonctionnalités développées vs fonctionnalités prévues.

Nombre de bugs identifiés et résolus : Indicateur de la stabilité du système.

Qualité du code : Évaluée à l'aide de métriques comme la complexité cyclomatique ou le respect des bonnes pratiques.

Indicateurs de performance

Temps de réponse moyen du système.

Utilisation des ressources (CPU, mémoire, etc.).

Taux de détection des emails frauduleux.

Taux de faux positifs/négatifs.

2.2. Tests et validation

Des campagnes de tests ont été menées selon une stratégie progressive :

Tests fonctionnels

Tests unitaires : Vérification de chaque composant isolé.

Couverture de code : Taux de lignes de code testées.

Tests des composants et services : Vérification du bon fonctionnement des interactions internes.

Tests d'intégration et de performance

Tests end-to-end (E2E) : Simulation des parcours utilisateur complets.

Tests de performance : Évaluation de la rapidité et de la fluidité de l'application sous différentes charges.

Tests de sécurité : Analyse des vulnérabilités (ex. : injection, XSS, accès non autorisé).

3. Bilan de la realization

3.1. Objectifs atteints

Le projet a permis de réaliser une application complète et fiable. Les objectifs initiaux ont été atteints :

- Mise en place d'une **application fonctionnelle**, stable et sécurisée.
- Création d'une **interface intuitive** facilitant la navigation des utilisateurs.
- Développement d'un **système performant d'analyse des emails** frauduleux.
- Renforcement de la **sécurité** au niveau de l'authentification et du traitement des données.

3.2. Points d' amélioration

Des pistes d'amélioration ont été identifiées pour les évolutions futures :

Optimisation des performances pour un traitement encore plus rapide des courriels.

Ajout de fonctionnalités avancées, telles que l'apprentissage automatique ou un tableau de bord statistique.

Amélioration de l'ergonomie et du design de l'interface.

Extension des capacités d'analyse, notamment pour d'autres types de fraudes ou de menaces.

Conclusion générale

Dans un contexte numérique marqué par la recrudescence des menaces de cybersécurité, et plus particulièrement des attaques de type phishing, ce projet s'est inscrit dans une démarche proactive visant à développer une solution intelligente, accessible et efficace pour la détection automatique des courriels frauduleux.

L'application développée s'appuie sur des technologies web modernes – Angular 17 pour le frontend et Spring Boot pour le backend – tout en intégrant la puissance de l'intelligence artificielle via un modèle d'apprentissage automatique conçu en Python. Cette combinaison a permis de concevoir un outil complet, capable de traiter et d'analyser automatiquement les emails suspects.

L'étude de l'existant a révélé les limites des solutions traditionnelles et mis en évidence les besoins réels des utilisateurs en matière de sécurité. Sur cette base, un cahier des charges fonctionnel et technique a été établi pour guider les différentes étapes de conception et de développement. L'intégration du modèle d'IA, notamment pour l'entraînement, l'optimisation et le déploiement, a constitué un véritable défi, tant sur le plan technique que méthodologique.

Les tests menés sur des jeux de données réels ont permis d'évaluer et de valider les performances du système. Les résultats se sont révélés prometteurs, ouvrant la voie à une utilisation en conditions réelles. Bien que des axes d'amélioration subsistent – en particulier concernant l'optimisation du modèle et l'enrichissement de l'interface utilisateur – la solution développée constitue une base robuste et évolutive pour renforcer la lutte contre le phishing.

En définitive, ce projet nous a permis de mobiliser des compétences pluridisciplinaires, allant du développement full-stack au machine learning, en passant par les problématiques de sécurité informatique. Il illustre concrètement l'apport de l'intelligence artificielle dans des applications de cybersécurité, tout en consolidant notre capacité à conduire un projet technologique de bout en bout, dans un contexte exigeant et à fort impact.

Bibliographie et Webographie

- AlEroud, A., & Karabatis, G. (2017). *Using Machine Learning Techniques for Phishing Detection*. Journal of Computer and Communications.
- Jain, A. K., & Gupta, B. B. (2018). *Phishing Detection: Analysis of Visual Similarity Based Approaches*. Security and Communication Networks.
- Scikit-learn documentation. <https://scikit-learn.org/stable/>
- Angular Official Documentation. <https://angular.io/docs>
- Spring Boot Documentation. <https://spring.io/projects/spring-boot>
- OWASP Foundation – Phishing. <https://owasp.org/www-community/Phishing>
- Google Cloud. (n.d.). *Best practices for deploying ML models*. <https://cloud.google.com>
- MySQL Documentation. <https://dev.mysql.com/doc/>
- Tsai, C.-F., Hsu, Y.-F., & Lin, C.-Y. (2020). *A Hybrid Approach for Phishing Detection Using Deep Learning and Natural Language Processing*. IEEE Access.

