Présenté par :

**BIAGNE LIGUE ELIE**

**KPASSOKRO MOISE**

**FOFANA YAYA**

**LAGO SIBA CHANTALE**

**EZEJIDEAKU EMMANUEL**

INDUSTRIE 4.0 AU SERVICE DE LA PÊCHE EN CÔTE D`IVOIRE

**RAPPORT DE PROJET D’APP3**

Ministère de la transition numérique et de la digitalisation

- - - - - - - - - - - - - - - - - - - - - - - - -

Ecole Supérieure Africaine des Technologies de l’information et de la communication

**ANNÉE UNIVERSITAIRE: 2024-2025**

**LISTE DES SIGLES ET ACRONYMES**

**ESATIC :** École Supérieure Africaine des Technologies de l’Information et de la Communication

**HTTP :** HyperText Transfert Protocole

**API:** Application Programming Interface

**JSON:** Javascript Object Notation

**UML:** Unified Modeling Language

**HTML:** HyperText Markup Language

**CSS:** Cascading Style Sheet

**SOMMAIRE**

**INTRODUCTION**

1. **BESOINS ET OBJECTIFS DU PROJET**

**1.1. Contexte**

**1.2. Motivation**

**1.3. Les Enjeux**

**1.4. Objectifs et Contraintes**

* + - * **Les objectifs techniques**
      * **Les contraintes**

1. **GESTION DU PROJET AVEC LA METHODE AGILE**
   1. **Présentation de la méthode Agile**
   2. **Organisation des sprints**
   3. **Utilisation de Pivotal Tracker**
2. **ANALYSE DES BESOINS**
   1. **Description des besoins fonctionnels et non fonctionnels**
   2. **Analyse des utilisateurs**
3. **CONCEPTION UML**
   1. **Présentation de UML**
   2. **Descriptions textuelles de cas d’utilisation**
   3. **Diagrammes de classe**
4. **ARCHITECTURE ET CONCEPTION TECHNIQUE**
   1. **Architecture technique**
   2. **Gestion des bases de données**
   3. **Technologies Utilisées**
5. **RESULTATS ET TRAVAIL REALISE**

**CONCLUSION**

**INTRODUCTION**

Dans un contexte mondial marqué par la montée en puissance des technologies numériques, la pêche illégale représente un problème majeur qui affecte l’économie et l’environnement, en particulier en Côte d’Ivoire. Ce phénomène menace non seulement les ressources halieutiques, mais également la subsistance de nombreuses communautés locales.

Pour relever ces défis, le recours à l’Industrie 4.0, intégrant des technologies telles que l’Internet des objets (IoT), les systèmes d’identification automatique (AIS), et les plateformes numériques, s’avère indispensable.

Ce projet propose le développement d’une plateforme innovante dédiée à l’identification des navires légaux et pirates afin de lutter efficacement contre la pêche illégale. Grâce à des bouées intelligentes équipées de capteurs et à une architecture numérique avancée, cette solution vise à améliorer la surveillance des zones maritimes et à renforcer la durabilité des pratiques de pêche. Ce rapport explore les besoins, les enjeux, et les objectifs associés à la mise en œuvre de ce système, tout en soulignant son impact potentiel sur la gestion des ressources marines et le développement économique de la Côte d’Ivoire.

1. **BESOINS ET OBJECTIFS DU PROJET**
   1. **Contexte**

En Côte d’Ivoire, la pêche illégale représente une menace croissante pour les écosystèmes marins et les communautés dépendantes de cette ressource. Les navires pirates, souvent non identifiables, exploitent abusivement les ressources halieutiques, causant une surpêche et une diminution drastique des stocks de poissons. L’introduction d’une plateforme basée sur les technologies de l’Industrie 4.0 permettrait de surveiller efficacement les zones maritimes et d’identifier en temps réel les activités suspectes.

* 1. **Motivation**

Le besoin de protéger les ressources marines et de garantir une pêche durable constitue la principale motivation de ce projet. En combinant des capteurs ultrason, des récepteurs AIS, et des systèmes de communication avancés, cette plateforme permettra de détecter et de différencier les navires légaux des navires illégaux. Une telle solution favorisera non seulement la protection de l’environnement, mais aussi l’amélioration des conditions de vie des pêcheurs locaux.

* 1. **Les Enjeux**

Les enjeux majeurs du projet sont :

- Surveillance accrue : Mettre en place un réseau de bouées intelligentes pour surveiller les zones maritimes sensibles.

- Lutte contre la pêche illégale : Réduire les activités des navires pirates grâce à une identification précise et rapide.

- Traçabilité : Assurer une meilleure traçabilité des produits de la mer pour renforcer la transparence et la conformité aux normes internationales.

- Soutien aux communautés locales : Offrir des outils numériques pour améliorer la gestion et la productivité des pêcheurs.

* 1. **Objectifs et Contraintes**
* **Les objectifs techniques**

- Concevoir des bouées intelligentes intégrant des capteurs pour détecter les navires et collecter des données en temps réel.

- Développer une plateforme centralisée qui analyse les données transmises par les bouées et génère des alertes sur les activités suspectes.

- Intégrer un système d’interface web interactif pour permettre aux autorités de surveiller les zones maritimes et d’intervenir rapidement.

- Assurer la connectivité et la sécurité des données transmises entre les bouées, le serveur, et l’interface utilisateur.

* **Les contraintes**

- Coût initial élevé : Le déploiement des bouées intelligentes et des infrastructures associées nécessitera un investissement financier important.

- Technologie et infrastructure : Certaines zones côtières pourraient manquer de la connectivité nécessaire pour la transmission des données.

- Acceptation par les utilisateurs : Les communautés locales devront être sensibilisées et formées à l’utilisation de ces nouvelles technologies.

- Conditions environnementales : Les intempéries et les conditions maritimes pourraient affecter le fonctionnement des équipements.

1. **GESTION DU PROJET AVEC LA METHODE AGILE**
   1. **Présentation de la méthode Agile**

La méthode Agile est une approche de gestion de projet qui favorise l'adaptabilité et la collaboration. Contrairement aux méthodes traditionnelles, telles que la méthode en cascade, où toutes les étapes sont planifiées à l'avance, Agile se base sur des itérations courtes appelées "sprints", permettant une livraison progressive des fonctionnalités. Cette approche met l'accent sur la collaboration entre les équipes, une communication constante et une réévaluation continue des besoins du client.

Dans le cadre de notre projet, qui consiste à créer une plateforme d'identification des navires légaux et pirates pour lutter contre la pêche illégale, la méthode Agile nous a permis de développer le système de manière progressive. Les retours des parties prenantes (autorités, communautés locales et experts marins) ont été intégrés après chaque itération. Cela garantit que la plateforme répond aux besoins réels des utilisateurs, tout en permettant d'ajuster les fonctionnalités au fur et à mesure du développement.

* 1. **Organisation des sprints**

L'organisation des sprints pour ce projet s'est étalée sur 5 semaines. Chaque sprint était dédié à une phase spécifique du développement de la plateforme. Voici un résumé des différentes étapes et tâches associées :

* **Sprint 1 : Analyse des besoins et conception de la base de données**  
  Durant cette phase, nous avons recueilli les besoins des utilisateurs, notamment les autorités de surveillance maritime et les pêcheurs locaux. Nous avons également défini les principales fonctionnalités de la plateforme, telles que la détection des navires à l'aide des bouées intelligentes, et conçu la base de données pour le stockage des informations recueillies.
* **Sprint 2 : Développement de l'interface utilisateur**  
  Cette phase a consisté à concevoir et à développer une interface web interactive. Cette interface permettra aux utilisateurs de surveiller les zones maritimes, de visualiser les données en temps réel et de recevoir des alertes sur les activités suspectes.
* **Sprint 3 : Développement du backend et mise en place des fonctionnalités de base**  
  Nous avons développé le backend pour gérer les fonctionnalités essentielles, comme la collecte des données des bouées intelligentes, leur stockage dans la base de données et la génération d'alertes en cas d 'activité suspecte.
* **Sprint 4 : Intégration des fonctionnalités avancées**  
  Cette phase a inclus l'intégration de fonctionnalités avancées, telles que la différenciation automatique entre navires légaux et illégaux grâce à des algorithmes, la visualisation des trajets des navires et des outils de recherche et de filtrage pour analyser les données divers.
* **Sprint 5 : Tests et déploiement**  
  La dernière phase a consisté à tester l'ensemble du système, à corriger les bugséventuels et à déployer une version opérationnelle de la plateforme. Les tests ont été effectués dans un environnement simulé pour garantir la fiabilité de la solution.
  1. **Utilisation de Git**

Git a été utilisé comme principal outil de gestion du code source tout au long du projet. Chaque fonctionnalité de la plateforme a été développée dans des branches distinctes pour garantir un travail organisé et éviter les conflits. Les engagements ont été réalisés régulièrement et accompagnés de messages explicites pour suivre les changements effectués. L'utilisation de Git a également permis une collaboration fluide entre les membres de l'équipe en facilitant l'intégration des différentes parties

1. **ANALYSE DES BESOINS**
   1. **Description des besoins fonctionnels et non fonctionnels**

1. **Besoins fonctionnels**

* **Surveillance maritime :**
* Détection en temps réel des navires présents dans les zones surveillées.
* Identification des navires légaux via les données AIS (Automatic Identification System).
* Signalement des navires suspects ou pirates en l'absence de signal AIS valide.
* **Gestion des bouées intelligentes :**
* Enregistrement des bouées déployées, incluant leur position GPS, leur état de fonctionnement, et leurs capteurs embarqués.
* Collecte et transmission des données captées (présence de navires, données environnementales).

• **Gestion des alertes :**

* Génération automatique d'alertes en cas de détection d'activités suspectes.
* Notifications envoyées aux autorités compétentes via des interfaces web ou mobiles.

• **Consultation des données :**

* Visualisation des données en temps réel sur une carte interactive.
* Accès aux historiques des activités maritimes pour une analyse approfondie.

• **Gestion des utilisateurs :**

* Gestion des rôles et des permissions pour les différents types d'utilisateurs (administrateurs, opérateurs, responsables maritimes).
* Enregistrement sécurisé des utilisateurs avec authentification.

#### **b) Besoins non fonctionnels**

• **Performance :**

* La plateforme doit pouvoir traiter un grand volume de données provenant des bouées intelligentes en temps réel.

• **Sécurité :**

* Les données (positions des bouées, informations AIS des navires) doivent être protégées grâce à des protocoles de chiffrement sensibles et une authentification renforcée.

• **Utilisabilité :**

* L'interface doit être intuitive et permettre une navigation rapide, même pour des utilisateurs peu expérimentés.

• **Accessibilité :**

* La plateforme doit être accessible depuis divers appareils (ordinateurs, tablettes, smartphones) et compatible avec les principaux navigateurs web.

• **Évolutivité :**

* L'architecture doit permettre l'ajout de nouvelles bouées et le traitement de volumes de données s'accumule sans altérer les performances.

### **1.2. Analyse des utilisateurs**

L'analyse des utilisateurs est cruciale pour identifier les besoins spécifiques des parties impliquées par la plateforme de lutte contre la pêche illégale.

Les principaux utilisateurs identifiés sont :

* **Les autorités maritimes** : elles supervisent les activités en mer, analysent les alertes générées par la plateforme et interviennent en cas de pêche illégale.
* **Les opérateurs de surveillance** : responsables de la gestion des bouées intelligentes et du suivi des données transmises.
* **Les développeurs de la plateforme** : en charge des mises à jour et du maintien du système pour garantir sa fiabilité.
* **Les décideurs politiques** : utiliser les données consolidées pour définir des stratégies et des réglementations visant à protéger les ressources halieutiques.

Chaque utilisateur a des attentes spécifiques :

* Les **autorités maritimes** souhaitent une interface simplifiée pour une prise de décision rapide.
* Les **opérateurs** ont besoin d'outils de gestion détaillés pour configurer les bouées et surveiller leur fonctionnement.
* Les **décideurs** recherchent des rapports et des analyses consolidées sur les activités maritimes.

Cette analyse garantit que la plateforme réponde efficacement aux besoins variés des parties prenantes et oriente les priorités de développement.

1. **CONCEPTION UML**
   1. **Presentation UML**

**UML (Unified Modeling Language)** est un langage visuel standardisé utilisé pour modéliser et concevoir des systèmes logiciels. Il permet de représenter des aspects structurels (comme les classes et les objets) et comportementaux (comme les interactions et les séquences d'actions) d'un système. UML est composé de plusieurs types de diagrammes, comme les diagrammes de classes, de cas d’utilisation et de séquence, pour faciliter la compréhension, la communication et la documentation du projet tout au long de son développement.

* 1. **Description textuelle des cas d’utilisation**

**Tableau de description du cas d’utilisation « S’authentifier  »**

**Tableau de description du cas d’utilisation « Consulter l’historique »**

**Tableau de description du cas d’utilisation « Gestion des utilisateurs »**

**Tableau de description du cas d’utilisation « Gestion des utilisateurs »**

**Tableau de description du cas d’utilisation « Rechercher un bateau»**

**Tableau de description du cas d’utilisation « Se deconnecter »**

**Figure 5: Use case Global**

**Source: StarUML**

**Tableau des regles de gestion**

* 1. **Diagramme de classe**

**Diagramme de classe**

**Source : StarUML**

1. **ARCHITECTURE ET CONCEPTION TECHNIQUE**
   1. **Architecture technique**

Pour le développement de notre application web de gestion des films documentaires, nous avons opté pour une combinaison de technologies modernes afin de garantir une interface utilisateur réactive, une gestion efficace des données et une architecture serveur robuste. Voici un aperçu des principaux composants technologiques que nous avons utilisés :

**Frontend :**

Nous avons utilisé **HTML, CSS et JavaScript**, les technologies de base du développement web, pour développer l'interface utilisateur.

* **HTML** (HyperText Markup Language) est utilisé pour structurer le contenu de la page, en définissant les éléments tels que les titres, les paragraphes, les formulaires et les boutons.
* **CSS** (Cascading Style Sheets) permet de styliser les pages web, en ajoutant des couleurs, des polices, des animations et en définissant la mise en page pour garantir une interface attrayante et conviviale.
* **JavaScript** est utilisé pour ajouter de l'interactivité et des fonctionnalités dynamiques, telles que la validation des formulaires, la gestion des événements (clics, survols, etc.), et les mises à jour de contenu en temps réel.

L'utilisation combinée de ces technologies garantit une interface utilisateur fluide et efficace, adaptée aux besoins des utilisateurs. Elle offre également une compatibilité étendue avec les navigateurs web modernes et un contrôle précis sur l'apparence et le comportement de l'application.

**Backend :**

Le serveur a été construit avec **Django**, un framework Python robuste et polyvalent qui facilite le développement rapide et sécurisé d'applications web.

* **Django REST Framework (DRF)** a été utilisé pour exposer des API RESTful, permettant la gestion des requêtes HTTP (ajouts, suppressions, modifications et consultations de données) et facilitant l'interaction avec les clients frontend.
* Django offre une structure modulaire, intégrant des fonctionnalités comme l'authentification, la gestion des sessions et la protection contre les attaques CSRF, ce qui améliore la sécurité globale de l'application.

**Architecture MVC (Modèle-Vue-Contrôleur) :**

Notre application suit une architecture basée sur le paradigme **MTV** (Modèle-Template-Vue) de Django, qui est une variante de l'architecture MVC. Cela garantit une séparation claire des responsabilités, facilitant la maintenance et l'évolutivité.

* **Modèle (Model)** :  
  Les modèles définissent la structure des données et interagissent avec la base de données MySQL. Ils représentent des entités essentielles comme les utilisateurs, les films et les administrateurs.
* **Template (équivalent de View dans MVC)** :  
  Les templates Django permettent de créer des interfaces utilisateur dynamiques en affichant les données reçues du contrôleur.
* **Vue (View)** :  
  Les vues Django gèrent la logique métier. Elles reçoivent les requêtes des utilisateurs, interagissent avec les modèles pour récupérer ou modifier des données, et renvoient une réponse (via des pages HTML ou des API REST).

L'architecture MTV/MVC permet de développer une application bien structurée, où chaque composant est indépendant, facilitant ainsi l'évolution et la collaboration entre équipes.

**Base de données :**

Nous avons choisi **MySQL**, une base de données relationnelle robuste, pour stocker et gérer les données de manière structurée et efficace. MySQL est particulièrement adapté aux applications nécessitant des relations complexes entre les données et garantit des performances élevées même avec des volumes importants.

Les principales tables créées dans la base de données sont :

* **Utilisateurs** :  
  Cette table contient les informations relatives aux utilisateurs, telles que leur nom, email, mot de passe (sécurisé via un hachage avec la bibliothèque **bcrypt**), et leur rôle dans l’application (administrateur ou utilisateur standard).
* **Films** :  
  Les données concernant les films sont stockées dans cette table, avec des champs pour le titre, la description, la durée, le réalisateur, et d’autres métadonnées pertinentes.
* **Administrateurs** :  
  Cette table contient les informations spécifiques aux administrateurs.

Le choix de **MySQL** repose sur sa capacité à gérer des relations complexes entre les données grâce à des jointures et des contraintes d'intégrité, tout en offrant une performance fiable et une large adoption dans l'industrie.

**Gestion des bases de données :**

Django simplifie la gestion des bases de données grâce à son **ORM (Object-Relational Mapping)**, permettant de manipuler les données directement via des modèles Python, sans écrire de requêtes SQL complexes.

Les principales opérations incluent :

1. **Migration des schémas** : Les modèles définis dans Django sont automatiquement traduits en tables MySQL via le système de migrations.
2. **Requêtes dynamiques** : Django ORM permet d'effectuer des requêtes complexes de manière intuitive, comme les filtrages, les jointures, et les agrégations.
3. **Gestion sécurisée des données** : Les mots de passe des utilisateurs sont hachés avant d’être stockés, et des mécanismes comme la gestion des transactions garantissent l'intégrité des données.
   1. **Technologies Utilisées**

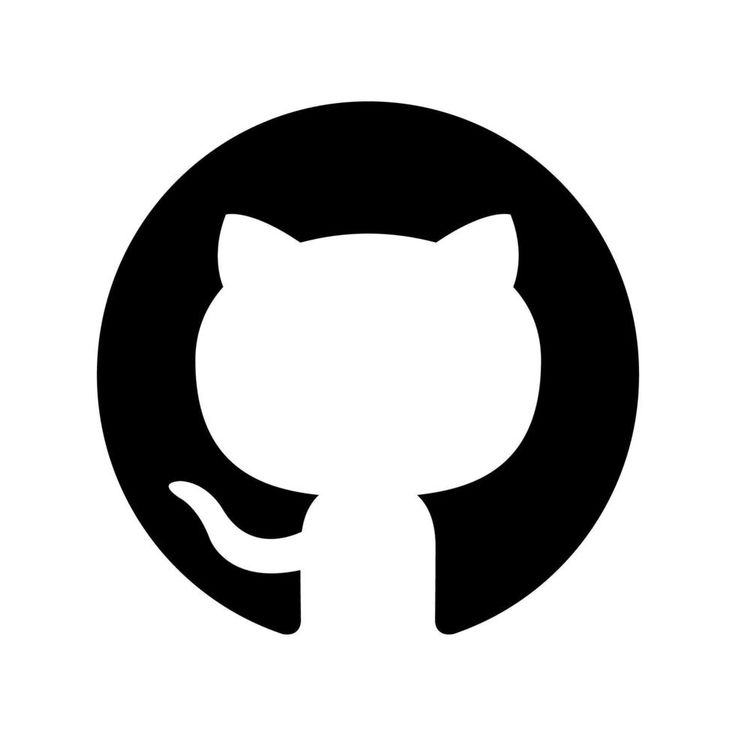
* **Gestion de version : Git**

Pour encadrer le développement de notre projet et afin de pouvoir gérer nos codes nous avons choisi d’utiliser Git et Github.

Git est un logiciel de gestion de version (Versioning) créé par Linus Torvalds. Il permet à plusieurs développeurs de travailler de façon collaborative sur les mêmes sources.

Github est un site web fournissant des repositories (dépôts de projets) hébergés. Ce site permet de voir facilement les dernières tâches effectuées. Voici celui qui a hébergé le projet :

<https://github.com/habibrolandt123/Projet-movies-management-Version1>





* **Couche base de données** : Mysql

MongoDB est une base de données NoSQL, et qui n’est donc pas relationnelle. Elle est composée de collections et non pas de tables. Chaque entrée est appelée document et a une syntaxe JSON et un identifiant.



**Couche serveur : Django / Django REST Framework / Django ORM**

La couche serveur est composée de trois éléments principaux :

1. **Django** :  
   Django est un framework web Python qui offre une structure robuste et sécurisée pour développer des applications web. Il gère l'ensemble des fonctionnalités côté serveur, comme le routage des requêtes, l'authentification, la gestion des sessions, et bien plus encore.
2. **Django REST Framework (DRF)** :  
   DRF est une extension puissante de Django qui permet de créer des API RESTful de manière simple et efficace. Il facilite la gestion des requêtes HTTP (GET, POST, PUT, DELETE) pour exposer des données et interagir avec les clients. DRF prend également en charge l'authentification, la pagination, et la validation des données, ce qui améliore la fiabilité des API.
3. **Django ORM (Object-Relational Mapping)** :  
   Django ORM simplifie l'interaction avec la base de données en permettant de manipuler les données via des modèles Python. Il remplace l'utilisation de requêtes SQL brutes, rendant le code plus lisible et maintenable

* **Couche Client/Présentation : HTML5, CSS3**

Cette couche est celle que verra l’utilisateur sur son navigateur web ou mobile. L’affichage se fait donc classiquement en HTML 5/CSS 3. Afin de simplifier le développement CSS, on utilise SASS qui permet d’utiliser entres autres des variables dans un fichier que l’on compilera en CSS.



**Mais qu’est-ce que le responsive ?**

Le responsive web design est le fait que le contenu d’une page web s’adapte à la taille d’écran utilisée. Cela permet de créer un unique site web qui sera adapté aussi bien pour un écran de PC fixe que pour un smartphone ou une tablette.

* **Conception :**

**Qu'est-ce que StarUML ?**

**StarUML** est un logiciel de modélisation UML (Unified Modeling Language) utilisé pour concevoir et documenter l'architecture de systèmes logiciels. Il est très prisé pour son support d'un large éventail de diagrammes UML, ce qui en fait un outil idéal pour les développeurs et architectes logiciels.



**Principales caractéristiques:**

* **Support des diagrammes UML :** StarUML permet de créer tous les principaux types de diagrammes UML, tels que les diagrammes de classes, de séquence, de cas d'utilisation, et d’activité.
* **Interface intuitive :** L'interface de StarUML est conçue pour être facile à utiliser, avec des outils de glisser-déposer et une navigation simplifiée.
* **Extensions :** StarUML propose des extensions pour étendre ses fonctionnalités et prendre en charge des langages spécifiques (comme Java ou Python) ou d’autres types de diagrammes (comme les diagrammes ER pour les bases de données).
* **Génération de code :** Il permet de générer du code à partir de modèles et même de créer des modèles à partir de code existant, facilitant la transition entre la modélisation et le développement.

**Avantages:**

* **Gain de temps :** StarUML automatise la génération de code et de documentation, réduisant les efforts manuels.
* **Compatible multi-plateformes :** Fonctionne sur Windows, Mac, et Linux.
* **Modularité et personnalisation :** Grâce à des extensions et des plugins, les utilisateurs peuvent personnaliser et étendre les fonctionnalités selon leurs besoins.

**Qu'est-ce que Figma ?**

**Figma** est un outil de conception d'interfaces utilisateur (UI) et de prototypage collaboratif basé sur le cloud. Il est largement utilisé pour créer des maquettes interactives et des designs d'application, permettant aux équipes de travailler ensemble en temps réel sur des interfaces de haute qualité.



**Principales caractéristiques:**

* **Design collaboratif en temps réel :** Grâce à sa fonction cloud, plusieurs personnes peuvent travailler simultanément sur un même projet.
* **Prototypage interactif :** Figma permet de créer des prototypes fonctionnels qui simulent la navigation et l'interactivité de l'application.
* **Organisation des éléments et composants :** Les éléments et composants réutilisables facilitent la cohérence des designs et la gestion des versions.

**Avantages :**

* **Accessible sur plusieurs plateformes :** Utilisable directement depuis un navigateur web, compatible avec Mac, Windows, et même mobile.
* **Facilité de partage et de feedback :** Figma permet de partager des liens de prototypes et de recueillir des retours en direct, ce qui simplifie la révision.
* **Outil tout-en-un :** Combine le design, le prototypage, et le feedback, ce qui le rend pratique pour les projets complets.

**Notre utilisation de Figma :**

Pour mon projet, Figma m'a aidé à concevoir l'interface utilisateur en facilitant le prototypage et la présentation des différentes interactions de l'application. Grâce à ses fonctionnalités, j'ai pu créer des designs cohérents et intuitifs pour les utilisateurs.

1. **RESULTATS ET TRAVAIL REALISE**

Dans cette partie, nous allons vous montrer les résultats que nous avons pu obtenir au cours de nos travaux sur notre projet a travers différentes images :

* 1. **Dashboard Administrateur**

* 1. **Historique**
  2. **Notification de détection de bateau**
  3. **Maquette du scanner**

**CONCLUSION**

Ce projet de développement d'une application web dédiée à la surveillance maritime et à l'identification des bateaux légaux et illégaux représente une avancée significative dans la lutte contre la pêche illégale. En intégrant des technologies modernes telles que Django, Django REST Framework, et MySQL, nous avons conçu une plateforme performante, sécurisée et évolutive, répondant aux besoins des administrateurs et des utilisateurs dans la gestion et l'analyse des données maritimes.

L'application offre une interface utilisateur intuitive et ergonomique, facilitant la consultation de l’historique des bateaux identifiés, la gestion des utilisateurs, et l’analyse des statistiques liées aux activités maritimes. Les administrateurs peuvent surveiller les zones maritimes grâce à des bouées équipées de capteurs, garantissant une détection précise et en temps réel des activités suspectes.

Ce projet met en lumière l'impact des nouvelles technologies dans la protection des ressources marines et la prévention des activités illégales. Grâce à son architecture bien structurée et à ses fonctionnalités avancées, cette solution permet d'améliorer l'efficacité des opérations de surveillance maritime et de gestion des données.

Les perspectives d'évolution de cette plateforme sont nombreuses. Elles incluent l’intégration de systèmes d’intelligence artificielle pour prédire les zones à risque, des alertes en temps réel basées sur des modèles comportementaux, ou encore une interopérabilité avec d'autres systèmes de surveillance maritime. Ce projet constitue une étape essentielle dans la modernisation des outils de gestion maritime, ouvrant la voie à une surveillance plus performante et à une meilleure préservation des ressources marines pour les générations futures.

**Webographie :**

 <https://www.djangoproject.com/> : Site et documentation officielles pour Django.

 <https://www.django-rest-framework.org/> : Site et documentation officielles pour Django REST Framework.

 <https://www.mysql.com/> : Site et documentation officielles pour MySQL.

 <https://getbootstrap.com/> : Site et documentation officielle pour Bootstrap.

 <https://www.canva.com/> : Outil de création graphique et banque d’images.

 <https://docs.python.org/> : Documentation officielle pour Python.

 <https://tailwindcss.com/> : Site et documentation officielle pour Tailwind CSS.