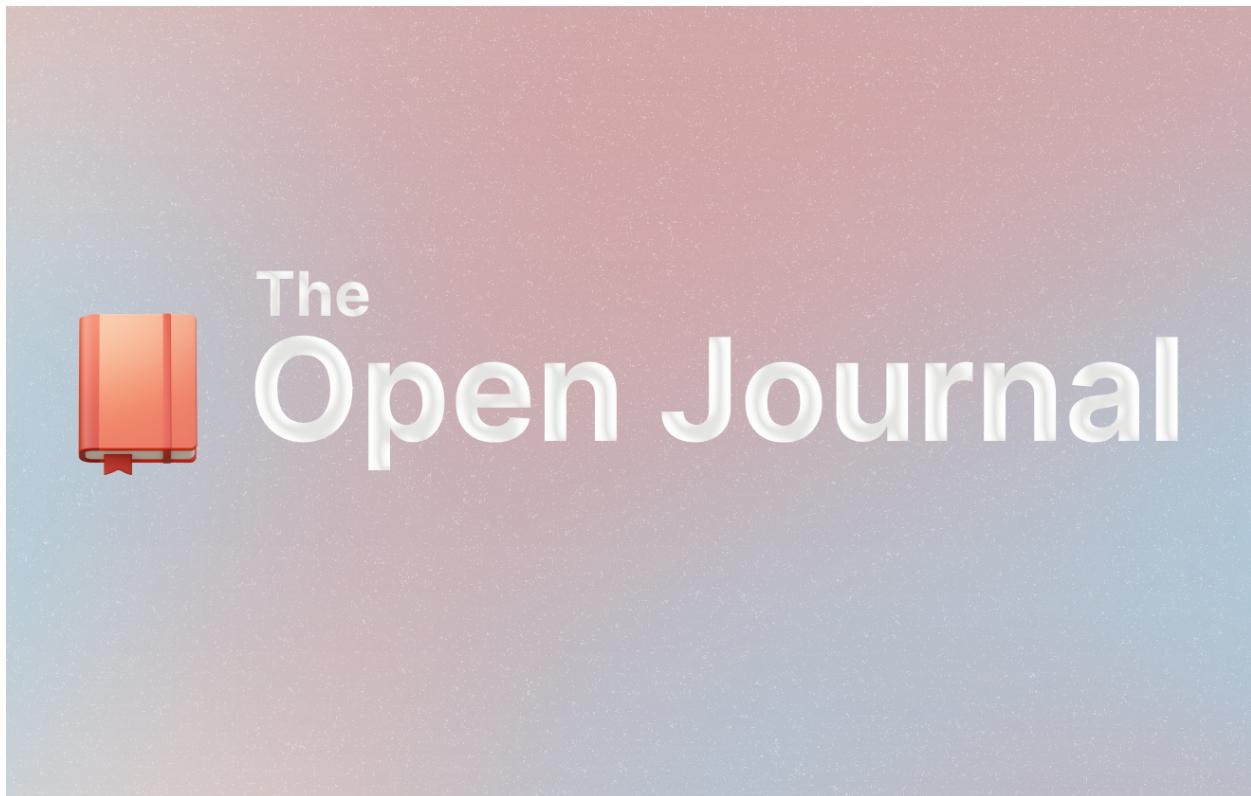


Rapport de Projet: The Open Journal

Par: BOUBCHER ISSAM | GLSID-1



I. Introduction

Dans notre ère numérique, les plateformes en ligne jouent un rôle central dans la diffusion de l'information et les interactions communautaires. Ainsi, la création et la gestion de sites web sont devenues essentielles. Ce rapport présente une vue d'ensemble d'un projet web réalisé dans le cadre d'un projet de fin d'études : un blog révolutionnaire qui permet à tous les utilisateurs de publier et de partager du contenu facilement.

Ce projet consiste à créer un blog avec contenu généré par les utilisateurs, afin de proposer un espace dynamique et inclusif où chacun peut s'exprimer, partager des expériences et interagir

avec un large public. L'objectif principal est de développer une plateforme conviviale et riche en fonctionnalités, favorisant une participation active, des points de vue variés et des interactions significatives entre les utilisateurs.

Ce rapport examine les différents aspects du projet, notamment sa conceptualisation, son processus de développement, ses fonctionnalités clés et ses perspectives d'avenir. Nous analyserons les principes de conception, la mise en œuvre technique et les considérations d'expérience utilisateur, afin de déterminer le succès global du projet et d'identifier les axes d'amélioration potentiels.

Visiter l'application sur le web: <https://open-journal.issaminu.com>

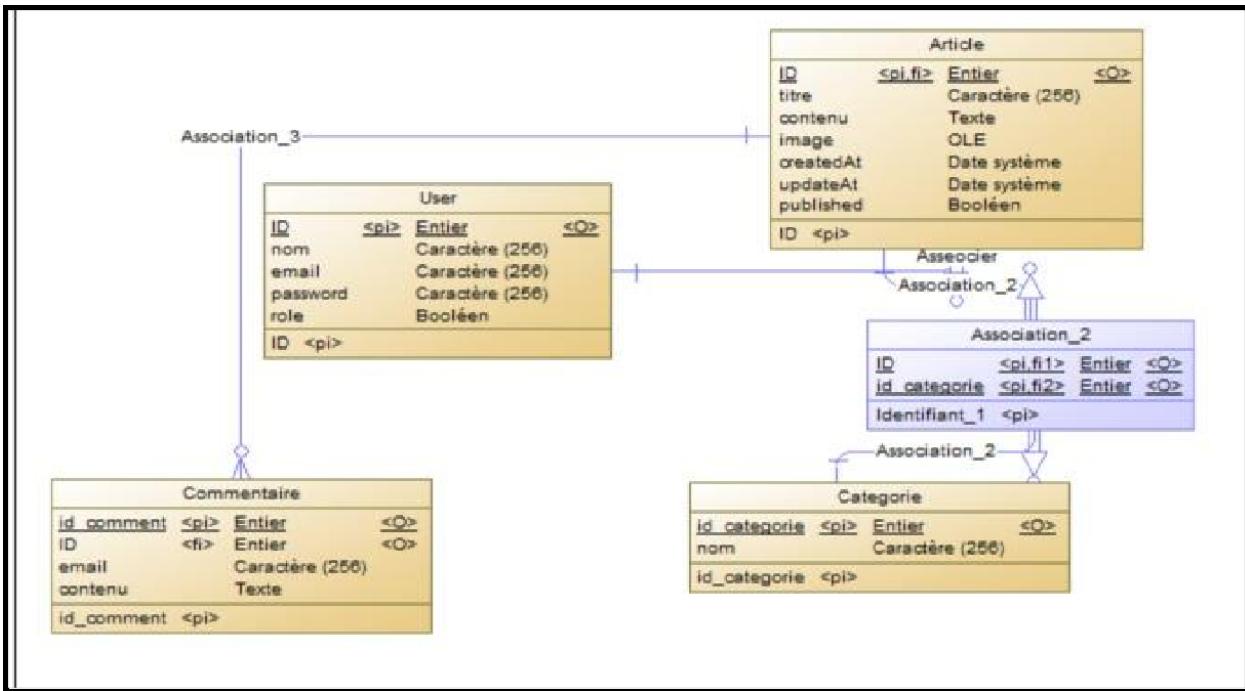
II. Conception

1. Introduction

Dans cette partie nous traitons l'aspect conceptuel de notre application. Pour la conception et la réalisation de cette dernière, nous utilisons le formalisme MERISE basé sur les diagrammes et offrant une flexibilité marquante.

2. Modèle Logique de données

Le modèle logique de données est une représentation abstraite et structurée des informations contenues dans une base de données. Il permet de définir les entités, les attributs et les relations entre les données, sans se préoccuper des détails de leur implémentation physique. Le modèle logique de données offre une vue conceptuelle claire et cohérente de l'organisation des données, facilitant ainsi la compréhension et la communication entre les différents acteurs impliqués dans la conception et la gestion d'une base de données.



III. Implementation

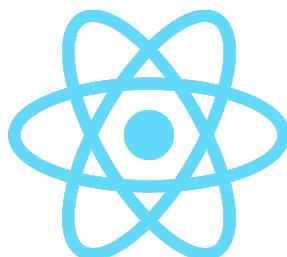
1. Outils:

1. Node.js



Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur JavaScript V8 de Google. Il permet aux développeurs de créer des applications web rapides et évolutives en utilisant JavaScript à la fois côté client et côté serveur. Node.js utilise un modèle asynchrone, non bloquant, ce qui le rend particulièrement adapté pour gérer un grand nombre de connexions simultanées et des opérations d'entrée/sortie intensives. Il est largement utilisé pour le développement d'API RESTful, de serveurs web, d'applications en temps réel, ainsi que pour la création d'outils en ligne de commande.

1. React:



ReactJS est une bibliothèque JavaScript populaire utilisée pour le développement d'interfaces utilisateur interactives et réactives. Créée par Facebook, ReactJS propose une approche déclarative pour la création de composants réutilisables et modulaires. Grâce à sa structure basée sur les composants, ReactJS permet de diviser l'interface utilisateur en morceaux indépendants, simplifiant ainsi le processus de développement et de maintenance.

2. Next.js



Next.js est un framework JavaScript basé sur React qui permet de créer des applications web côté serveur (Server-side Rendering) et côté client (Client-side Rendering). Il offre une approche

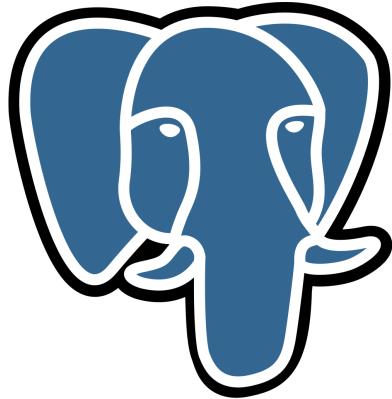
simplifiée du développement d'applications web modernes en combinant le rendu côté serveur avec les fonctionnalités avancées du rendu côté client. Next.js facilite la création de pages dynamiques et réactives grâce à sa prise en charge native du routage, de la gestion des états, de la génération de pages statiques et de la préparation des données. Il offre également des fonctionnalités de performance optimisées.

3. Prisma



Prisma est un ORM (Object-Relational Mapping) moderne et puissant qui facilite l'interaction entre une application et une base de données. En utilisant Prisma, les développeurs peuvent modéliser leur schéma de base de données à l'aide d'une syntaxe intuitive, déclarative et basée sur TypeScript. Prisma prend ensuite en charge la génération automatique de requêtes SQL sécurisées et performantes, ainsi que la manipulation des données de manière efficace.

5. Postgres



PostgreSQL, également connu sous le nom de Postgres, est un système de gestion de base de données relationnelle open source réputé pour sa robustesse, sa fiabilité et sa conformité aux normes SQL. Il offre une large gamme de fonctionnalités avancées, telles que la prise en charge des transactions ACID (Atomicité, Cohérence, Isolation, Durabilité), les vues matérialisées, les déclencheurs, les fonctions stockées et la réPLICATION. PostgreSQL est apprécié pour sa grande extensibilité, permettant aux développeurs de créer des types de données et des extensions personnalisées pour répondre aux besoins spécifiques de leurs projets.

4. Vercel



Vercel est une plateforme de déploiement web qui simplifie le processus de mise en ligne d'applications et de sites web. Elle offre une infrastructure cloud évolutive et sécurisée pour

héberger des projets basés sur des technologies telles que React, Vue.js, Next.js et bien d'autres. Vercel se distingue par sa facilité d'utilisation, sa configuration zéro et sa prise en charge intégrée du déploiement continu. Grâce à ses fonctionnalités de prérendu, de mise en cache et de distribution mondiale de contenu, Vercel permet d'obtenir des performances élevées et une expérience utilisateur optimale.

5. Tailwind



Tailwind CSS est une bibliothèque CSS utilitaire qui permet de développer rapidement des interfaces web élégantes et réactives. Contrairement aux frameworks CSS traditionnels, qui fournissent des composants prédéfinis, Tailwind CSS se concentre sur la création d'une collection de classes CSS réutilisables.

5. Next-Auth



NextAuth.js est une bibliothèque open source qui facilite l'authentification et l'autorisation dans les applications web développées avec Next.js. Elle offre une solution complète pour gérer les flux d'authentification tels que l'inscription, la connexion et la déconnexion des utilisateurs, en prenant en charge une variété de fournisseurs d'identité tels que Google, Facebook, GitHub et bien d'autres. NextAuth.js simplifie également la mise en œuvre des stratégies d'authentification basées sur les tokens tels que JWT (JSON Web Tokens) et les cookies sécurisés. Grâce à sa configuration simple et flexible, NextAuth.js permet de mettre en place rapidement et facilement un système d'authentification robuste, offrant une expérience utilisateur fluide et sécurisée.

6. TypeScript:



TypeScript est un langage de programmation open source qui étend JavaScript en y ajoutant des fonctionnalités de typage statique. Il permet aux développeurs de déclarer les types de variables, de paramètres de fonction et de retour de fonction, offrant ainsi une vérification statique des erreurs potentielles dès la phase de développement. Grâce à cette fonctionnalité, TypeScript permet de détecter et de résoudre rapidement les erreurs de type, ce qui améliore la robustesse et la maintenabilité des applications.

7. Zod



Zod est une bibliothèque JavaScript open source qui facilite la validation des données en fournissant un moyen simple et déclaratif de définir des schémas de validation. Cette bibliothèque est conçue pour garantir l'intégrité et la cohérence des données manipulées dans une application. En utilisant Zod, les développeurs peuvent définir des schémas de validation pour des objets, des tableaux, des chaînes de caractères et bien d'autres types de données. Zod offre des fonctionnalités avancées telles que la validation conditionnelle, les validations imbriquées et les transformations de données. Avec sa syntaxe claire et concise, Zod simplifie le processus de validation des données, aidant ainsi les développeurs à détecter et à gérer les erreurs plus facilement.

8. Framer Motion



Framer Motion est une bibliothèque JavaScript open source qui permet d'ajouter des animations fluides et interactives à des interfaces utilisateur. Spécialement conçue pour React, Framer Motion offre une syntaxe intuitive et déclarative pour créer des transitions, des mouvements et des effets visuels attrayants. Cette bibliothèque permet de manipuler facilement les propriétés CSS, les transformations et les interpolations, offrant ainsi un contrôle précis sur le comportement des animations.

2. Back-end

1. Utilisation des React Server Components

Dans le backend, j'ai opté pour l'utilisation des dernières fonctionnalités de Nextjs telles que le répertoire App (qui remplace le répertoire Pages), qui permet des mises en page imbriquées, il a également un support pour les React Server Components, permettant d'exécuter le code React dans le serveur. Cela simplifie considérablement le processus de récupération des données.

Par exemple, au lieu d'utiliser un

`useEffect` côté client pour récupérer des données, nous récupérons les données avant même d'envoyer au client le bundle JS, nous effectuons nos appels à la base de données, puis nous générerons le HTML sur le serveur, et enfin nous le récupérons pour le client. Voici un exemple d'utilisation des React Server Components dans Next.js :

```
import Home from "@/app/home/Home";
import prisma from "prisma/prisma";
import { Article } from "@/lib/zod";

const getArticles = async (numberOfArticles: number): Promise<Article[]> =>
  return (await prisma.article.findMany({
    where: {
      isPublished: true,
    },
    select: {
      id: true,
      title: true,
      content: true,
      createdAt: true,
      image: true,
      author: {
        select: {
          id: true,
          name: true,
        },
      },
      category: {
        select: {
          id: true,
          name: true,
        },
      },
    },
    orderBy: { id: "desc" },
    take: numberOfArticles,
  })
  .map((article) => {
    const author = article.author;
    const category = article.category;

    return {
      ...article,
      author: {
        id: author.id,
        name: author.name,
      },
      category: {
        id: category.id,
        name: category.name,
      },
    };
  });

```

```

        })) as Article[];
    };

const getMetaData = async () => {
    const articleCount = await prisma.article.count({}); 
    const userCount = await prisma.user.count({}); 
    let latestAddition = await prisma.article.findFirst({
        orderBy: { createdAt: "desc" },
        select: { createdAt: true },
    });
    return { articleCount, userCount, latestAddition };
};

export type getMetaDataType = Awaited<ReturnType<typeof getMeta

const HomePage = async () => {
    const articles = await getArticles(50);
    const metaData: getMetaDataType = await getMetaData();
    return <Home articles={articles} metaData={metaData} />;
};

export default HomePage;

```

2. Utilisation des Route Handlers

Dans Next.js 13.2, les gestionnaires de routes (Route Handlers) ont été annoncés, ils sont l'équivalent `/api` pour le `App Directory`. Les gestionnaires de routes sont cachés dans un `Route Group` dont le path est `/app/(Route Handlers)`, ils acceptent un objet de requête (`Request` / `NextRequest`) et retournent un objet de réponse (`Response` / `NextResponse`).

En outre, j'ai utilisé la bibliothèque `Zod`, pour valider toutes les entrées fournies par l'utilisateur et les traiter de manière sûre.

Voici un exemple:

```

import { CustomError, isCustomError } from "@/lib/utils";
import { categorySchemaCreate, categorySchemaUpdate } from "@/l

```

```

import { NextRequest, NextResponse } from "next/server";
import prisma from "@/prisma/prisma";
import { ZodError } from "zod";

export async function PATCH(req: NextRequest) {
  try {
    const validatedBody = categorySchemaUpdate.parse(req.body);
    const { id, name } = validatedBody;
    const categories = await prisma.category.findMany({
      where: {
        authorId: 0,
      },
    });
    if (categories.some((category) => category.name === name)) {
      throw CustomError(`The "${name}" category already exists`);
    }
    const category = await prisma.category.update({
      where: { id: id },
      data: { name },
    });
    if (!category) {
      throw CustomError(`The category with id "${id}" does not exist`);
    }
    return NextResponse.json({ category }, { status: 201 });
  } catch (error) {
    if (error instanceof ZodError) {
      return NextResponse.json({ message: error.message }, { status: 400 });
    }
    if (isCustomError(error)) {
      return NextResponse.json(
        { message: error.message },
        { status: error.status }
      );
    }
    return NextResponse.json({ message: error }, { status: 500 });
  }
}

```

```
    }  
}
```

3. Utilisation des Server Actions

Les Server Actions (fonctions de serveur) sont une toute nouvelle fonctionnalité de Next.js (sortie le 4 mai 2023, il y a à peine un mois!), il s'agit d'une fonctionnalité visant à simplifier drastiquement le processus de mutation des données pour les applications React. Une action serveur est un RPC (Remote Procedure Call) qui évoque une fonction sur le serveur depuis le client. Les Server Actions remplacent le besoin de POST de données en utilisant une API, puisque nous pouvons maintenant simplement provoquer des fonctions serveur et leur passer des données client.

Voici un exemple :

```
// Code côté client:  
// RPC dans le client  
  
import { createComment } from "@/app/actions"; // Importer un "S  
...  
.  
. .  
  
<form  
action={(formData) => {  
    createComment(formData); // ===== Appel d'un Server Action  
    setIsAddComment(false);  
}}  
className="w-full flex flex-col justify-start items-start"  
>  
<Label  
    className="text-lg font-semibold text-[#f1e5e5]"  
    htmlFor="send-comment"  
>  
    Write a comment  
</Label>  
<div className="w-full flex flex-col justify-center items-end">
```

```

<Textarea
  name="content"
  id="send-comment"
  autoFocus
  required
  placeholder="Type your message here..."
  className=" rounded-lg min-h-[10rem] outline-none bg-[#3c3a4d] border border-[#3c3a4d] p-4 text-white"
/>
<input
  required
  type="hidden"
  name="articleId"
  value={article.id}
/>
<input
  required
  type="hidden"
  name="userId"
  value={session.user.id}
/>
<div className="flex flex-row mt-4 gap-4">
  <Button
    variant={"secondary"}
    type="submit"
    className="bg-cyan-600 w-44 font-semibold hover:bg-cyan-700 py-2 px-4 rounded-md text-white"
  >
    Send
  </Button>
  <Button
    className="bg-gray-300 w-24 text-black hover:bg-gray-400 py-2 px-4 rounded-md"
    onClick={() => setIsAddComment(false)}
  >
    Cancel
  </Button>
</div>

```

```
</div>
</form>
```

```
// Code côté serveur
// Server Action ici (fonctions provoqué dans le serveur):

export const createComment = async (formData: FormData) => {
  try {
    const comment = formData.get("content") as unknown as string
    const articleId = formData.get("articleId") as unknown as string
    const userId = formData.get("userId") as unknown as string
    if (!comment || !articleId || !userId) return;
    await prisma.comment.create({
      data: {
        articleId: parseInt(articleId),
        content: comment,
        authorId: parseInt(userId),
      },
    });
    revalidatePath(`/read/${articleId}`);
    revalidatePath("/home");
  } catch (e) {
    console.error(e);
  }
};
```

3. Front-end

Pour le front-end, on a utilisé React, Tailwind et une nouveau bibliothèque des composants Tailwind nommé [Shadcn/ui](#).

Homepage

The ENSET Journal

306 articles · 21 students registered · Latest publication: Jun 3rd

Write an article

The online blog of ENSET Mohammedia. The open platform where students can freely express themselves, offering a captivating virtual hub for intellectual exploration and thought-provoking articles, personal narratives, cutting-edge research, and engaging discussions.

Back To School
By Issam Boubcher | 16 min read | ekeke

...
Continue reading →

Highest Mountains
By Issam Boubcher | 6 min read | ACTIVE GROUNDS

...
Continue reading →

Culpa corporis labore
By Mack Stiedemann | 9 min read | Pentagon

...
Continue reading →

Magni exercitationem accusamus eius enim suscipit natus eos incident
By Ms. Jeanette Christiansen | 8 min read | Giraffe

...
Amet porro porro dignissimos doloremque ratione voluptatum a facere. Optio beatoe inventore distinctio. Quae dolor dolorum. Ut blanditis iuu... quibusdam maiores rerum at soepe quee. Culpa iu...

Ajout d'un article

Write an article

Title

Category

Publish Cancel

Voir un article



ARTIFICIAL INTELLIGENCE

Edit Article

How Artificial Intelligence Is Transforming The World

Issam Boubcher · Jun 6th 2023 · 18 min read



Although there is no uniformly agreed upon definition, AI generally is thought to refer to "machines that respond to stimulation consistent with traditional responses from humans, given the human capacity for contemplation, judgment and intention."^[3] According to researchers Shubhendu and Vijay, these software systems "make decisions which normally require [a] human level of expertise" and help people anticipate problems or deal with issues as they come up.^[4] As such, they operate in an intentional, intelligent, and adaptive manner.

Intentionality

Artificial intelligence algorithms are designed to make decisions, often using real-time data. They are unlike passive machines that are capable only of mechanical or predetermined responses. Using sensors, digital data, or remote inputs, they combine information from a variety of different sources, analyze the material instantly, and act on the insights derived from those data. With massive improvements in storage systems, processing speeds, and analytic techniques, they are capable of

Modifier un article

Save Edit

ARTIFICIAL INTELLIGENCE

Cancel

How Artificial Intelligence Is Transforming The World

Issam Boubcher · Jun 6th 2023 · 18 min read



Although there is no uniformly agreed upon definition, AI generally is thought to refer to "machines that respond to stimulation consistent with traditional responses from humans, given the human capacity for contemplation, judgment and intention."^[3] According to researchers Shubhendu and Vijay, these software systems "make decisions which normally require [a] human level of expertise" and help people anticipate problems or deal with issues as they come up.^[4] As such, they operate in an intentional, intelligent, and adaptive manner.

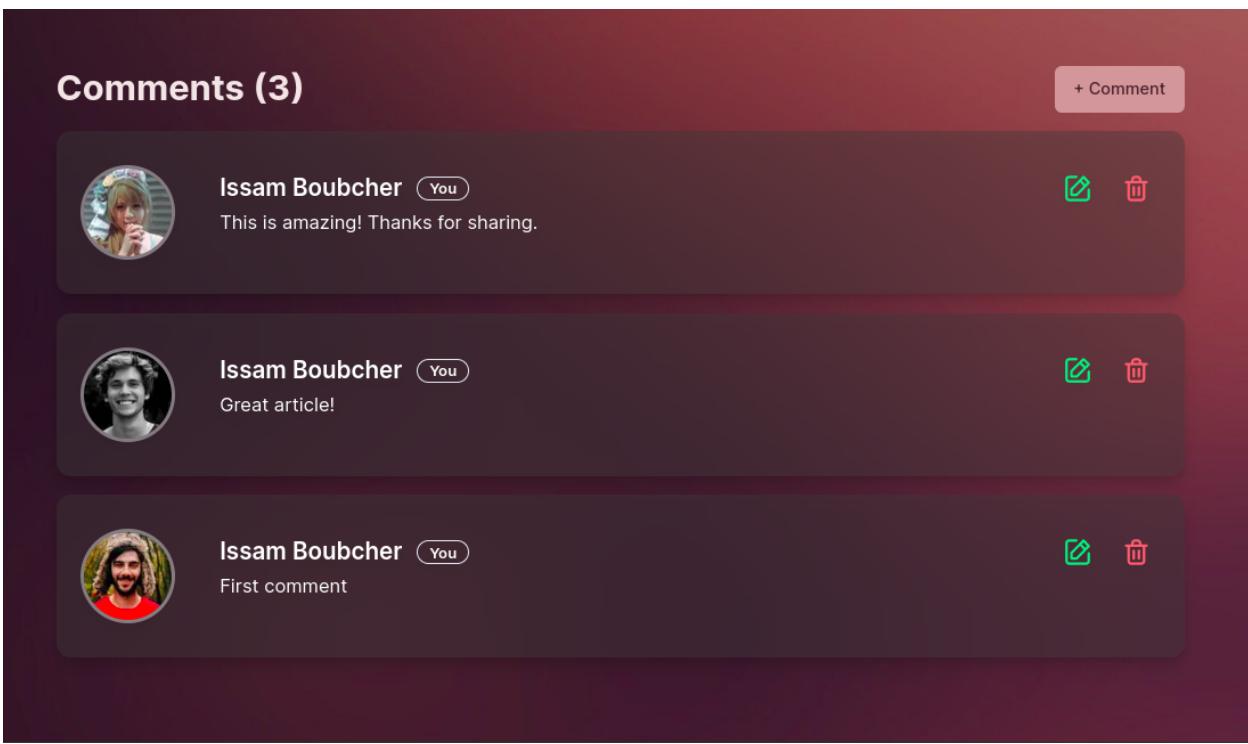
Intentionality

Artificial intelligence algorithms are designed to make decisions, often using real-time data. They are unlike passive machines that are capable only of mechanical or predetermined responses. Using sensors, digital data, or remote inputs, they combine information from a variety of different sources, analyze the material instantly, and act on the insights derived from those data. With massive improvements in storage systems, processing speeds, and analytic techniques, they are capable of tremendous sophistication in analysis and decisionmaking.

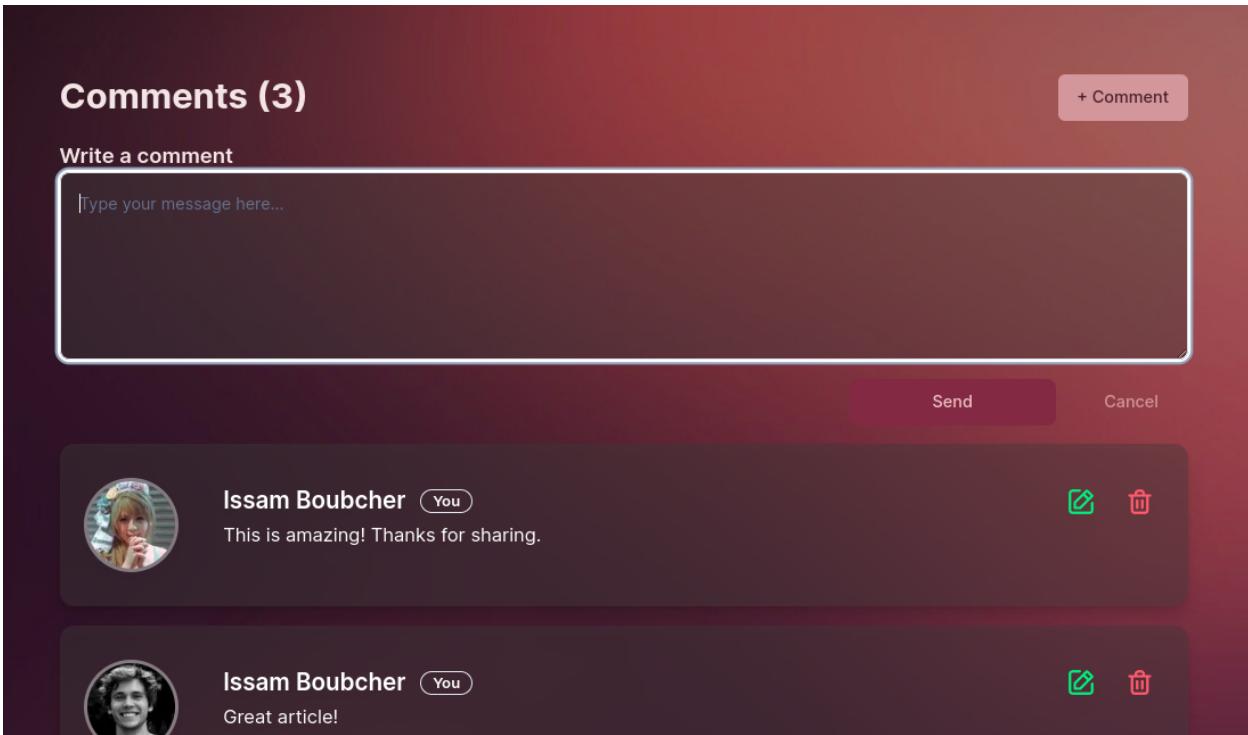
Intelligence

AI generally is undertaken in conjunction with machine learning and data analytics.^[5] Machine learning takes data and looks for underlying trends. If it spots something that is relevant for a practical problem, software designers can take that knowledge and use it to analyze specific issues. All that is required are data that are sufficiently robust that algorithms can discern useful patterns. Data can come in the form of digital information, satellite imagery, visual information, text, or unstructured data.

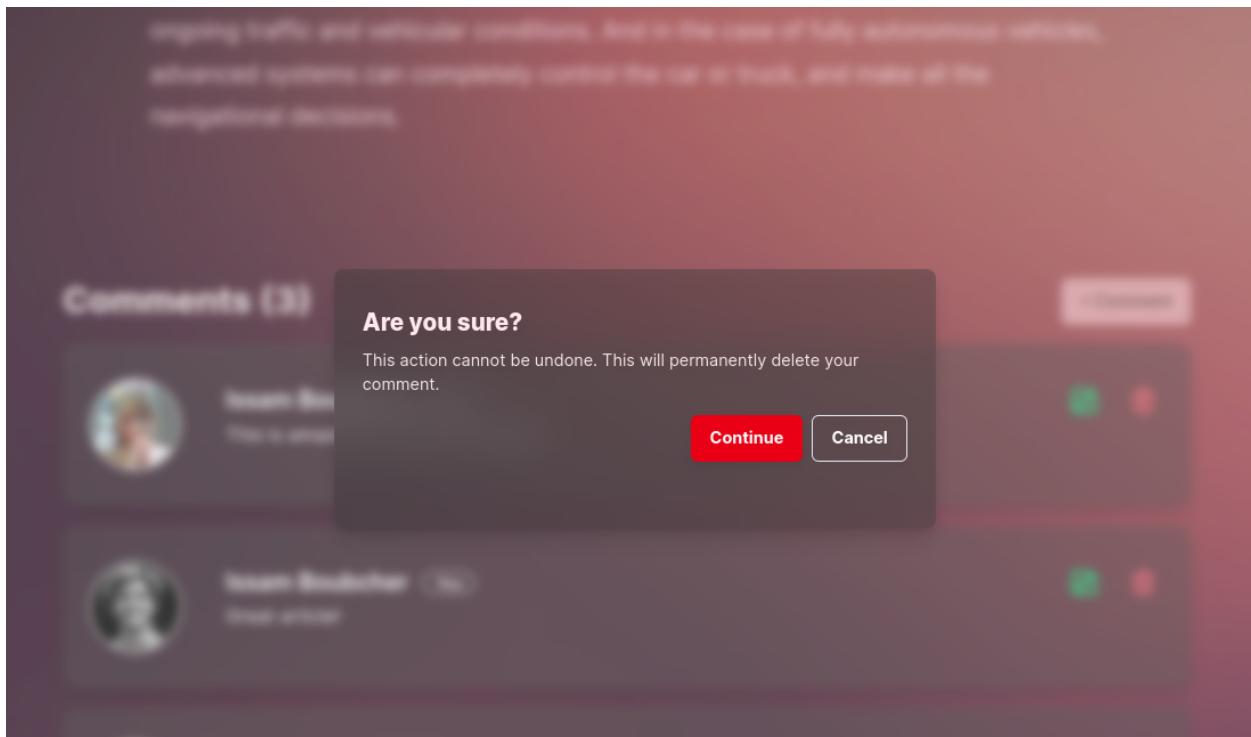
Commentaires



Ajouter un commentaire



Supprimer un commentaire



Conclusion

Ce rapport nous a permis d'examiner en détail un projet web novateur de création d'un blog avec contenu généré par les utilisateurs. Tout au long de cette étude, nous avons exploré les différentes phases du projet, de sa conceptualisation à son développement, en passant par la mise en œuvre des fonctionnalités clés et la prise en compte de l'expérience utilisateur.

L'objectif principal de ce projet était de fournir aux utilisateurs une plateforme conviviale et inclusive où ils peuvent partager leurs idées, leurs expériences et interagir avec d'autres membres de la communauté. Grâce à des fonctionnalités innovantes et une conception centrée sur

l'utilisateur, le blog a réussi à offrir une expérience engageante et à favoriser des interactions significatives.

Cependant, il convient de noter que ce projet n'est qu'une étape initiale, et qu'il existe encore des opportunités d'amélioration et de croissance. Par exemple, des efforts supplémentaires pourraient être déployés pour promouvoir l'engagement de la communauté, mettre en place des mesures de modération de contenu plus robustes et envisager des stratégies d'expansion pour atteindre un public plus large.

En conclusion, ce rapport a mis en évidence l'ampleur et le potentiel du projet de blog avec contenu généré par les utilisateurs. Il a également souligné les défis, les réalisations et les perspectives d'avenir de cette initiative. En capitalisant sur les forces et en abordant les domaines d'amélioration identifiés, ce projet peut continuer à évoluer et à offrir une expérience en ligne enrichissante pour ses utilisateurs.