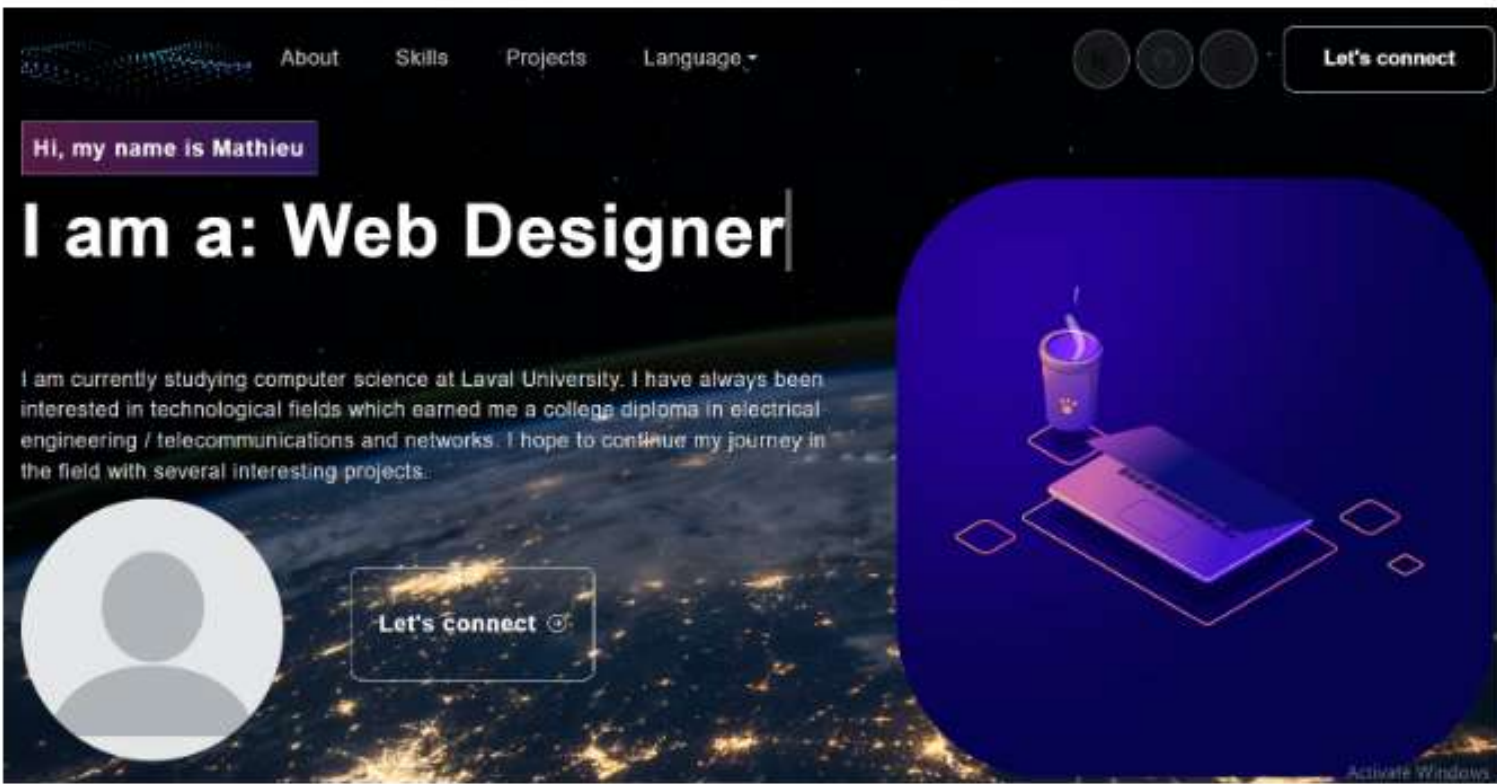


Portefeuille



www.mathieuchretien.com

Ceci est mon portfolio de carrière créé dans React. J'utilise AWS Amplify pour héberger ce site Web.

Dépendance

- L'extrémité avant "@aws-amplify/ui-react": "^5.0.4" (Saisie du formulaire de contact)
"aws-amplify": "^5.3.3" (configuration AWS)
"bootstrap": "^5.3.0" (CSS du cadre de l'interface utilisateur)
"i18next": "^23.2.7" (Gestionnaire de fichiers json du cadre de traduction)
"lottie-react": "^2.4.0" (Cadre d'animation)
"react": "^18.2.0" (cadre React)
"react-bootstrap": "^2.8.0" (cadre d'interface utilisateur)
"react-dom": "^18.2.0" (rendu DOM)
"react-i18next": "^13.0.1" (gestionnaire de fichiers json du cadre de traduction)
"react-lazy-load-image-component": "^1.6.0" (flou d'image à chargement paresseux)
"react-pdf": "^7.1.2" (cadre de lecteur PDF)
- Backend
"aws-sdk": "^2.1413.0" (AWS de service)
"nodemailer": "^6.9.3" (Pièce jointe de support de messagerie simplifiée)

🔗 Front-end

Tous les CSS personnalisés sont dans App.css

- Composant principal
 - Barre de navigation
 - Bannière
 - Compétences
 - Projet
 - Formulaire de contact
 - Bas de page
- Composante modale
 - CV en (barre de navigation, formulaire de contact, pied de page)
 - ESP dans (Projet)
 - Algo dans (Projet)
 - Portefeuille en (Projet)
- Composante Animation
 - AnimationBanner in (Animation de loterie de bannières)
 - AnimationContact in (Animation du formulaire de contact lottie)
 - TexteBanner in (animation de texte de bannière)

Remarque : L'animation dans la barre de navigation et le pied de page utilisent un gif car Lottie créait un DOM trop grand (plus de 12 000 éléments).

Back-end

Le seul backend nécessaire pour mon site Web était le formulaire de contact pour m'envoyer des e-mails. Pour ce faire, j'ai utilisé un lien de mutation graphql vers une fonction lambda dans AWS.

Voici le schéma :

```
type Sender @model @auth(rules:[{allow:public, operations:[create]}]) {  
  id: ID!  
  name: String!  
  email: String!  
  phone: String!  
  message: String!  
}
```

Voici la fonction JavaScript :

```
/* Amplify Params - DO NOT EDIT
  ENV
  REGION
  SES_EMAIL
Amplify Params - DO NOT EDIT */

/**
 * @type {import('@types/aws-lambda').APIGatewayProxyHandler}
 */
const aws = require('aws-sdk');
const nodemailer = require('nodemailer');

const ses = new aws.SES()
let transporter = nodemailer.createTransport({SES: ses})

exports.handler = async(event) => {
  for (const streamedItem of event.Records) {
    if (streamedItem.eventName == 'INSERT') {
      //pull off items from stream
      const userName = streamedItem.dynamodb.NewImage.name.S
      const userEmail = streamedItem.dynamodb.NewImage.email.S
      const userPhone = streamedItem.dynamodb.NewImage.phone.S
      const userMessage = streamedItem.dynamodb.NewImage.message.S

      try{
        const mailFormat = await transporter.sendMail({
          from: process.env.SES_EMAIL,
          to: process.env.SES_EMAIL,
          subject: 'Portfolio message ',
          html:`<p> Name: ${userName}<br /> Email: ${userEmail}<br /> Phone: ${userPhone}<br /> Message: ${us
        })
        return mailFormat;
      } catch(e){
        console.error('ERROR', e)
      }
    }
  }
  return 'done';
}
```

Avec cela, lorsqu'un graphql est créé, la fonction déclenchera automatiquement la fonction node.js. (Déclencheur Lambda)

Remarque : l'e-mail doit être un e-mail vérifié dans le service de messagerie simple d'AWS. (SES_EMAIL = e-mail de choix)