|  |
| --- |
| Université du Québec à Rimouski |
| Travail Pratique I |
| INF37407 – Technologies de l’inforoute  Professeur – Yacine Yaddaden, Ph. D. |

|  |
| --- |
| ⭢ Chadha Hammadi  ⭢ Guillaume Terriere |

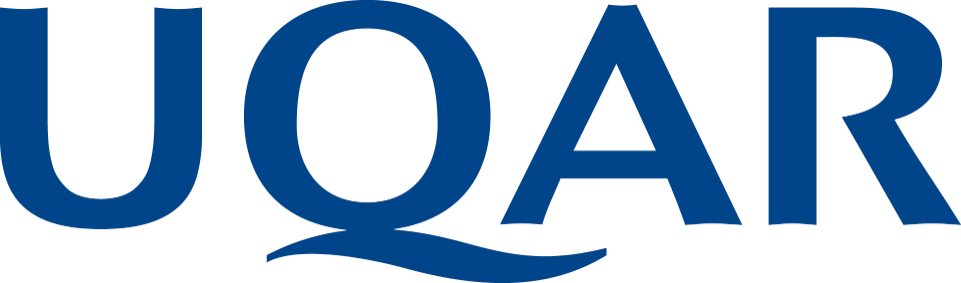


Table des matières :

Introduction & Contexte

Les technologies utilisées

Architecture et composition du site créé

La gestion des états avec Redux

API externe

Ce que vous avez aimé et moins aimé dans le projet

Problèmes & Difficultés rencontrées

Conclusion

Références

Introduction & Contexte

# Une plateforme de réservation de vols ou d'hôtels simplifie le processus de recherche, de comparaison et de réservation, offrant ainsi aux utilisateurs une expérience fluide et pratique. Ces applications exploitent généralement des technologies Web telles que HTML, CSS et JavaScript pour fournir des fonctionnalités interactives et dynamiques. Notre objectif initial est de créer une application Web utilisant la bibliothèque JavaScript React.js, axée sur la réservation de billets d'avion et de chambres d'hôtel.

Les technologies utilisées

# Dans le cadre de ce Travail pratique, on a utilisé le Microsoft Visual Studio Code (version 2022) dans un environnement de développement avec Node.js .La programmation Web développé et maintenu par Javascript (ES6) et HTML5 et elle se base sur les bibliothèques JavaScripts React.js, Redux, React\_Router, Axios et Redux\_Thunk.Pour le coté apparence du notre site on a utilisé le framework React- Bootstrap et React-FontAwsome.

Architecture et composition du site créé

Dans notre application Web de réservation de billets d'avion et d’hôtels, vous trouverez généralement les éléments suivants :

1. ***La structure globale et les fonctionnalités mises en place*** : Pour simplifier la gestion des données de notre application, nous avons opté pour l'utilisation de Redux. Ainsi, notre structure de code repose sur la mise en place d'un store dans index.js, servant à stocker l'ensemble des données, et à injecter par la suite les mises à jour correspondantes. Par la suite, les reducers sont créés pour effectuer des opérations sur le store existant et générer un nouveau, comme illustré par des fichiers tels que ConnectUserReducer, FactureReducer, HotelReducer, etc., en tant qu'exemples de fichiers .jsx.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Les actions, quant à elles, sont chargées de contenir toutes les instructions nécessaires pour apporter les modifications souhaitées dans le store. Parmi les exemples de fichiers .jsx figurent FactureActions, HotelAction, UtilisateursAction, etc.…

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementLe répertoire de routage permet d'afficher la composante appropriée en fonction de l'URL saisie. Dans notre projet, nous avons les itinéraires suivants : Connexion, CreerCompte, Facture, Hotel, Vol, etc.….

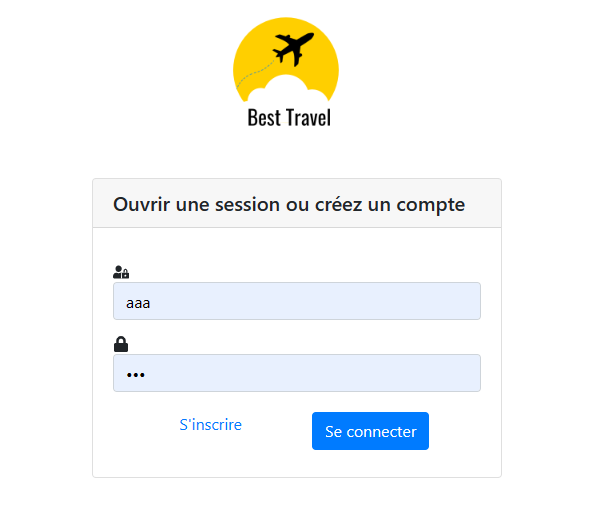
Dans cette structure de code, nous accordons également une attention particulière au répertoire des composants, qui comprend plusieurs fichiers. jsx définissant les composants de notre application tels que le menu, ResultatRechercheVol, ResultatRechercheHotel, LigneFactureHotel, LigneFactureVol, Facture, et ListeReservation.

Une image contenant texte, capture d’écran, Police

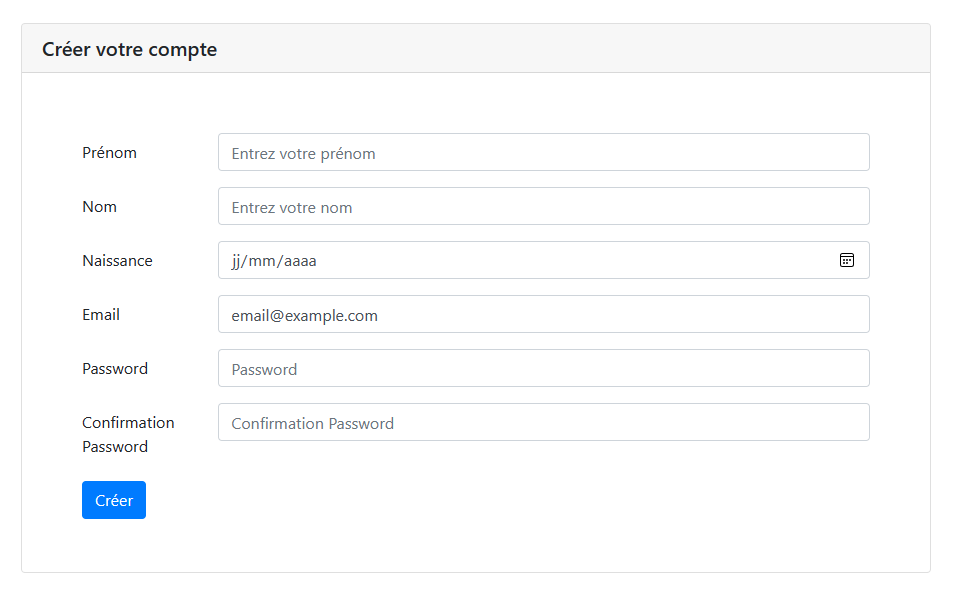
Description générée automatiquement

1. ***Interfaces utilisateur conviviales*** :

il s'agit d'interfaces faciles à utiliser qui permettent aux utilisateurs de se connecter (ou de s'inscrire), de rechercher, de sélectionner, de réserver, de payer, de consulter leurs factures et même de modifier leurs profils.

***2.1 Interface connexion :***

C'est la page principale de notre application où chaque utilisateur doit se connecter pour rechercher un vol ou un hôtel. Si un utilisateur accède à notre plateforme pour la première fois, il doit créer un compte en remplissant le formulaire suivant :



***2.2 Interface vol :***

Il s'agit d'une interface dédiée à la recherche et à la réservation de vols. Il est crucial de spécifier l'origine, la destination, la date de départ, la date de retour et le nombre de passagers pour effectuer une recherche précise. En cliquant sur le bouton de recherche, un appel est envoyé à l'API Travelpayouts/travel, qui est utilisée dans le contexte de notre projet.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

***2.3 Interface Hôtel :***

Il s'agit d'une interface dédiée à la recherche et à la réservation d'hôtels. Il est crucial de spécifier la destination, la date d'arrivée, la date de départ et le nombre de personnes pour effectuer une recherche précise. En cliquant sur le bouton de recherche, un appel est envoyé à l'APITravelpayouts/hotel utilisée dans le cadre de notre projet.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

***2.4 Interface Facturation et réservation :***

Il s'agit d'une interface qui présente les réservations effectuées pour les vols et les hôtels, offrant la possibilité de supprimer la réservation ou de la confirmer en appuyant sur "Payer". Cela déclenche la génération de la facture correspondante.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

***2.5 Interface modifier profil :***

Une image contenant texte, capture d’écran, Police, Rectangle

Description générée automatiquement

Il s'agit d'une interface qui permet la modification des données de l'utilisateur, ainsi que la possibilité de se déconnecter et de quitter l'application.

# La gestion des états avec Redux

Notre modèle de données traverse divers états gérés par Redux, permettant la gestion locale des réservations de vols, des réservations d'hôtels, des factures, ainsi que la gestion des utilisateurs.Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

# API externe

Afin de consommer les services et d'utiliser l'API de TravelPayouts, nous avons employé Axios pour la gestion des requêtes HTTP. L'API externe de TravelPayouts se divise en deux parties distinctes, l'une pour les vols et l'autre pour l'hébergement. En nous appuyant sur la documentation, nous avons utilisé les méthodes suivantes pour :

-Hotel :API : GET <http://engine.hotellook.com/api/v2/cache.json?location=Saint-Petersburg&hotelId=277083&checkIn=2019-09-13&checkOut=2019-09-18&currency=rub&limit=1&token=PasteYourTokenHere> ,

qui prend comme paramètre : location ,CheckIn,CheckOut,locationId… et retourne comme réponse :stars,locationId,PriceFrom,PriceAVG…

-Vol :API : GET <https://api.travelpayouts.com/v1/prices/calendar?depart_date=2016-11&origin=MOW&destination=BCN&calendar_type=departure_date&token=PutHereYourToken>

qui prend comme paramètre : origin,destination,departure\_date,return\_date… et retourne comme réponse : origin,destination,price,transfers,airline,flight\_number…

# Ce que vous avez aimé et moins aimé dans le projet

Le projet satisfait pleinement les objectifs énoncés dans le travail pratique. Il a grandement contribué à notre familiarisation avec JavaScript React.js, ainsi qu'à l'organisation efficace de nos états de réservation en exploitant les fonctionnalités offertes par Redux.

Par ailleurs, il a constitué une excellente occasion de mettre en pratique l'utilisation d'une API externe et de maîtriser l'outil Axios pour simplifier la gestion des requêtes HTTP.

En résumé, ce projet a été une expérience constructive permettant de mettre en pratique l'ensemble des concepts étudiés au cours de la formation.

Néanmoins, la charge de travail dans les interfaces a été parfois intense, surtout compte tenu du calendrier serré et de la présence d'autres travaux pratiques simultanés.

# Problèmes & Difficultés rencontrées

Voici les principaux défis rencontrés durant le projet :

Le premier défi réside dans la gestion de l'API externe, notamment en ce qui concerne la manipulation des jetons d'authentification pour accéder aux données nécessaires.

Le deuxième défi concerne la persistance des données, en l'absence d'une base de données adéquate. Il a été nécessaire d'initialiser manuellement les profils des utilisateurs ainsi que leurs informations.

La troisième difficulté majeure a été liée à la gestion des réservations de vols et d'hôtels, et à l'acheminement des confirmations vers le processus de facturation en vue du règlement des paiements.

# Conclusion

Le projet d'application web de réservation de vols et d'hôtels illustre efficacement l'application pratique des notions abordées en front-end dans le cadre du cours. Il met en lumière des problèmes techniques spécifiques à JavaScript et à React.js, offrant ainsi une opportunité d'apprentissage significative.

Dans la session précédente, nous avons réussi à créer une application web similaire en utilisant HTML, ASP.Net Core et PHP, démontrant ainsi la polyvalence des différentes technologies. En fin de compte, le choix de la technologie à utiliser repose sur les compétences et la maîtrise de l'outil par le développeur.

# Références

1-[Comment créer une application Web ReactJS ? (create-react-app) - YouTube](https://www.youtube.com/watch?v=p00FA83KBoU&t=1704s)

2-Introduction à React.js\_TO DO LIST \_Yacine Yaddaden