



RAPPORT DU PROJET D'INITIATION

APPLICATION WEB

APPGROSSING



Réalisé par:

- **FADILI OTHMANE**
- **NAINIA YOUNESS**
- **Bellal M.YASSINE**

IA&GI
2020/2021

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

Remerciements

C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à l'aboutissement de ce travail. Nos remerciements sont adressés à nos enseignants à L'école **ENSAM Casablanca** pour leur contribution à notre formation et pour ses recommandations précieuses et leur disponibilité. Nous tenons aussi à remercier tous les **étudiants IA&GI CI-2**. Enfin nous tenons à remercier également les membres de jury qui me font l'honneur de juger ce projet.

Résumé

Avec le nombre d'applications qui ne font qu'augmenter en nombre et en fonctionnalité, le marché devient de plus en plus compétitif et il devient difficile pour les développeurs d'évaluer avec précision le degré de succès de leurs applications ou de celles de leurs concurrents. Les analyseurs d'Android se sont donc imposés comme une solution efficace et facile à utiliser. Nous proposons ici notre version sous forme d'une application web responsive qui contient l'essentiel de ce qu'on attend d'un tel service. Durant le développement d'APPGROSSING Nous nous sommes tournés vers la pointe des technologies web afin d'optimiser l'expérience des users et nous travaillons à rendre notre base de données la plus fiable en un temps plus au moins réel.

Abstract

With the number of applications increasing in number and functionality, the market becomes more and more competitive and it becomes difficult for developers to accurately assess the degree of success of their applications or those of their concurrent. Android analyzers have therefore established themselves as an efficient and easy-to-use solution. Here we offer our version as a responsive web application that contains most of what is expected of such a service. During the development of APPROSSING We have turned to the cutting edge of web technologies in order to optimize the user experience and we are working to make our database more reliable in a more or less real time.

Plan

► Introduction

I. Présentation & Objectif du Projet

1. Contexte & Objectif
2. Le concept du APPGROSSING
3. L'équipe de projet

II. Études Préliminaires

1. Etude de l'Existant
2. Choix des technologies

III. Analyse et Conception

1. Architecture de l'Application
2. Le modèle conceptuel de données (MCD)
3. Description des entités
4. Stratégie de collecte des données

IV. Réalisation

1. Environnement du travail
2. Structure du Projet
3. Front End de l'application
4. Back End de l'application
5. Gestion des erreurs

► Conclusion

► Webographie

Introduction

Le développement Web désigne de manière générale les tâches associées au développement de sites Web destinés à être hébergés via un intranet ou Internet. Le processus de développement web comprend, entre autres, la conception de sites web, le développement de contenu web, l'élaboration de scripts côté client ou côté serveur et la configuration de la sécurité du réseau.

Le développement Web est le codage ou la programmation qui permet de faire fonctionner un site Web, selon les exigences du propriétaire. Il traite principalement de l'aspect non conceptuel de la création de sites Web, qui comprend le codage et l'écriture du balisage.

Le développement Web va de la création de pages en texte brut à des applications Web complexes, des applications de réseaux sociaux et des applications commerciales électroniques.

❖ **Présentation & Objectif du Projet**

I. Contexte & Objectif :

Dans l'ombre de cette pandémie globale, les gens ont été amenés à rester chez eux, ce qui a considérablement augmenté leur consommation des applications mobiles. Ils ont donc plus que jamais besoin d'un service qui leur permet de retrouver les applications les mieux classées ou qui répondent à certains de leurs critères. Les développeurs eux, cherchent à observer et analyser continuellement le marché, les tendances ainsi que l'avancement de leurs concurrents afin de mieux cibler les clients qui bénéficieront du service qu'ils proposent. Dans le cadre de ce projet de stage, nous avons essayé de répondre à ce besoin pour satisfaire les deux parties sous forme d'un site web.

II. Le concept du APPGROSSING :

Après une étude des analyseurs de Play store déjà présents (qui sont payants pour la plupart), nous avons tenté de regrouper toutes les fonctionnalités qu'on a jugé utiles de chaque site web. Le concept d 'APPGROSSING est de satisfaire ces clients en leurs transmettant les données du Play store fidèlement et esthétiquement avec une bonne 'user expérience' et un prix raisonnable. Tout en leurs procurant des filtres pour sélectionner les applications qu'il souhaite et des graphes pour visualiser le développement d'une certaine application.

III.L'équipe de projet :

L'équipe qui a réalisé ce travail est composée de 3 étudiants de l'ENSA Casablanca filière A&GI CI-2. Chacun a été amené à réaliser une partie du projet tous en gardant un esprit de collaboration et un climat de communication constant. Cela nous a permis d'avancer plus efficacement et ainsi d'achever le projet plus rapidement :

- NAINIA Youness** : Frontend
- FADILI Othmane** : Backend
- BELLAL Mohammed Yassine** : collecte des données, conception et création de la base de données.

NB : Les technologies et frameworks utilisés dans ce projet ont dû être maîtrisés avant de commencer la réalisation.

❖ Études Préliminaires

I. Etude de l'Existant :

1-Analyse de l'existant

Les entreprises et les développeurs indépendants qui travaillent dans le domaine des applications Android publient principalement leurs applications sur google Play store En raison du grand nombre d'applications disponibles sur ce marché et aussi du grand nombre d'utilisateurs qui le consomment au quotidien. Pour que ces développeurs puissent avoir un avantage concurrentiel, il est nécessaire de publier des applications qui ont à la fois une forte probabilité de succès et qui assure beaucoup d'installations dès leurs débuts. Afin d'arriver à cela, il est nécessaire d'utiliser une plateforme qui permet au développeur d'analyser son concurrent et même d'avoir des informations détaillées sur leur audience.

2-Objectif Ciblé :

- Mettre au point une première version fonctionnelle d'une solution web présentant les services essentiels d'un Android analyser qu'on pourrait améliorer en ajoutant plus de fonctionnalités plus tard.
- Collecter assez de données pour tester le bon fonctionnement de l'application web et tenter d'automatiser l'opération.

II. Choix des technologies :

Choisir la bonne technologie est l'une des étapes les plus importantes afin d'assurer le succès d'un projet. Sachant que le projet Web est assez grand et qu'il sera réalisé en équipe, nous avons veillé à déterminer les Frameworks les plus appropriées. Schématiquement, le développement des applications Web nécessite **un front-end, un back-end et une base de données**. Toutes ces couches forment une pile technologique (Stack).

 Le choix du Framework pour le Front-End :

Le monde **JavaScript** est un environnement riche en outils, bibliothèques et autres frameworks.

Il existe de nombreux Frameworks JavaScript (JS) pour un développement front-end plus rapide et plus interactif.

Actuellement, **Angular, React & Vuejs** sont les derniers frameworks JavaScript frontaux qui vous permettent de concevoir des interfaces utilisateur extrêmement engageantes.

Après une étude comparative précise des 3 framework, on a décidé de choisir le **framework React** qui répond à nos besoins sur le projet et qui présente beaucoup d'avantages par rapport aux autres frameworks.

Le choix du Framework pour le Back-End :

Node JS avec Express est l'un des meilleurs framework pour développer des applications web. D'ailleurs, il s'agit d'un outil extrêmement populaire recommandé par de plus en plus de développeurs web.

La première raison pour laquelle nous utilisons Node Js serait le fait qu'il fonctionne en JavaScript. De plus, une application Web construite avec JavaScript aux deux extrémités se charge très rapidement et en douceur.

La raison suivante serait la pertinence de Node pour les applications en temps réel, évolutives. De plus, les applications développées avec Node.js peuvent bien gérer plusieurs demandes d'utilisateurs.

Le choix du système de base de données :

Notre application Web a besoin de stocker ses données et c'est là qu'interviennent les bases de données.

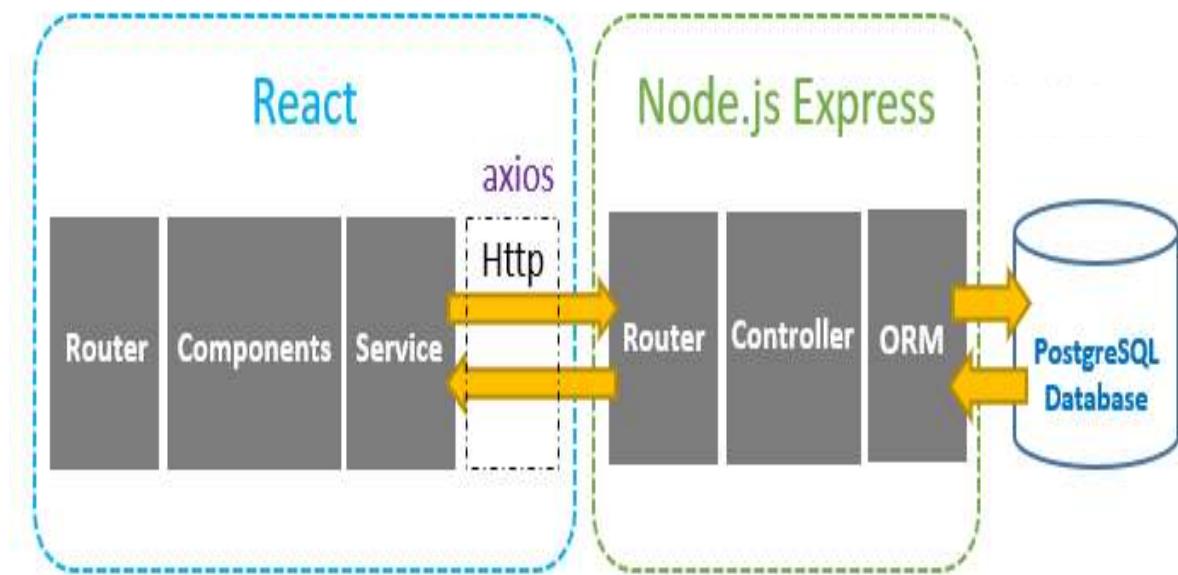
PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données (le cas de notre application), même dans des situations critiques.

La flexibilité de PostgreSQL ne se reflète pas seulement dans sa fonctionnalité et son extensibilité, la base de données offre également de nombreuses possibilités de configuration logicielle et matérielle.

❖ Analyse et Conception

I. Architecture :

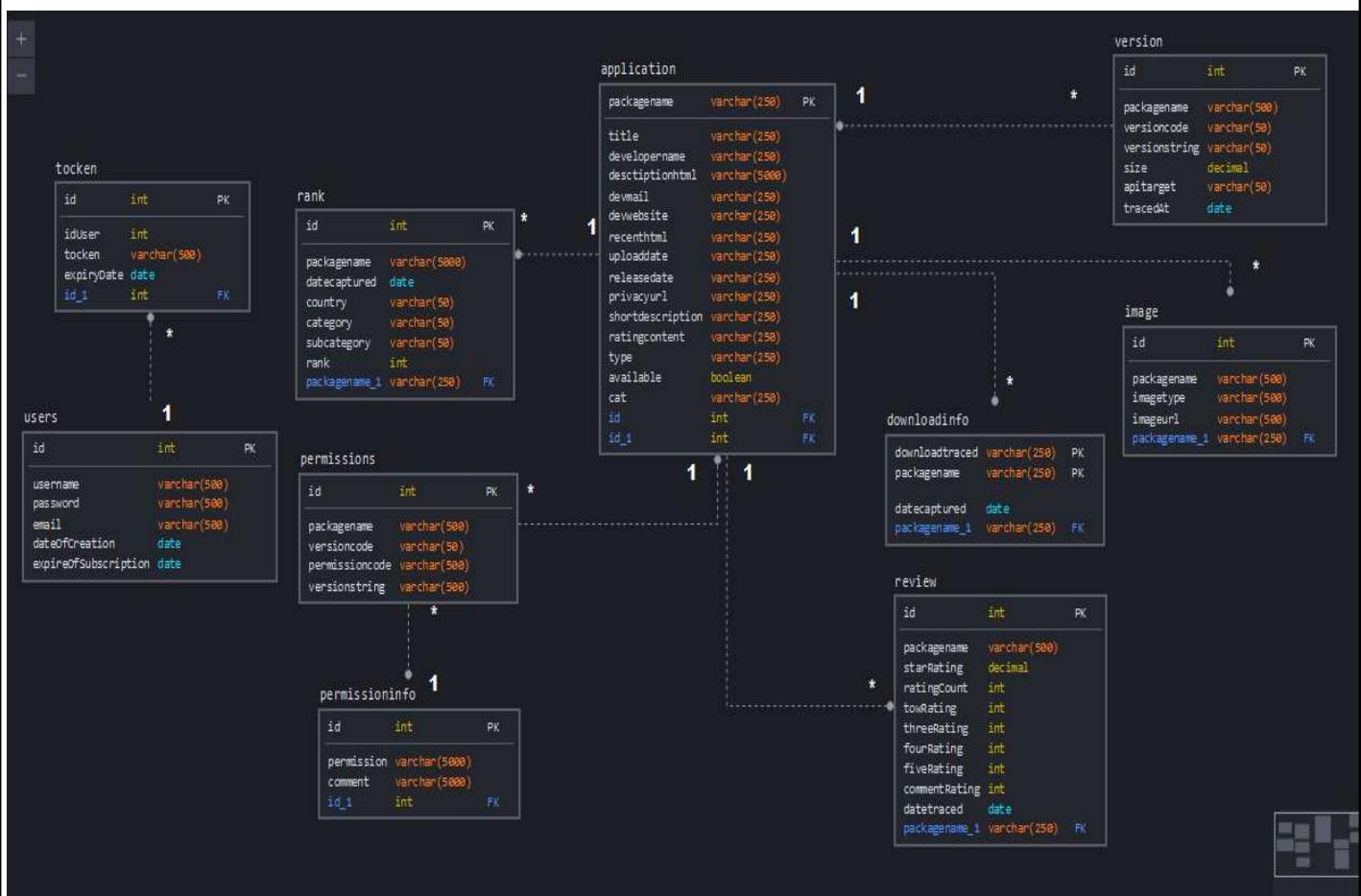
Voici l'architecture de notre système :
React, Node.js Express, PostgreSQL



- Node.js Express exporte les API REST et interagit avec la base de données PostgreSQL à l'aide de Sequelize ORM.
- React Client envoie des requêtes HTTP et récupère les réponses HTTP à l'aide d'Axios , consomme des données sur les composants.
- React Router est utilisé pour naviguer vers les pages.

II. Le modèle Conceptuel de Données :

Voici le modèle conceptuel de données (MCD) de notre Application Web :



III. Description des entités :

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre elles. De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens ; les objets sont des instances de classes et les liens sont des instances de relations.

Les principales classes (entités) de notre application sont :

Application :

Il contient toutes les informations basic sur l'application. Il est inséré pour la premier fois quand on détecte l'application, puis on le met à jours quand l'application devient introuvable.

id [PK] intégrée	packagename [PK] character varying	title character varying (111)	developername character varying (1)	descriptionhtml character varying (8255)	uploaddate date	releasedate date	shortdescription character varying (555)	available boolean	cat character varying
13373551	com.qiwi.wia...	Words in a Pic 2	QIWI Games AB	This game is for a true word geni...	2018-07-06	2015-08-14	Try the game that everyone love...	true	GAME
13952731	com.oup_elt.olt	Oxford Learner's Thesau...	Oxford University ...	[null]	2015-12-02	[null]	Oxford Learner's Thesaurus: A D...	true	APPLICATION
12827967	air.abcklubben...	ABC-klubben 2	Natur & Kultur	I ABC-klubben, Diamantjakten får...	2020-08-29	2020-08-29	At the ABC club you practice wo...	true	APPLICATION
13169084	air.bahraniapp...	Draw Expressive Comics	BahraniApps	Easily create expressive, artistic ...	2020-07-31	2017-09-05	Easily create expressive, artistic ...	true	APPLICATION
12973932	air.br.com.pok...	Poker 5 Card Draw - 5CD	MegaJogos	Poker Five Card Draw Online wit...	2020-09-03	2017-02-02	Poker 5 Card Draw Online with t...	true	GAME
12900132	air.com.bmap...	Rainbow Ice Cream Co...	bmapps	It is time to make some more a...	2020-08-10	2016-06-18	Deliver some amazing snow con...	true	APPLICATION
12979523	air.com.clever...	Gold Strike	CleverMedia, Inc.	(Note: This is the real official Gol...	2017-02-06	2011-12-13	Play Gold Strike, the popular acti...	true	GAME
12934638	air.com.fernus...	Fen Bilimleri Mobil Kütü...	FERNUS	Fen Bilimleri Mobil Kütüphane	2020-10-01	2020-10-01	Science Mobile Library	true	APPLICATION
12895863	air.com.functu...	Adam and Eve 3	Functu	Adam is again on a journey to fin...	2020-04-29	2017-07-31	Adam is again on a journey to fi...	true	GAME
12879725	air.com.gamb...	Gambino Poker	Gamesko Limited	There are almost 3 million Gamb...	2020-07-28	2013-01-04	Gambino Poker - free texas hold...	true	GAME
12892120	air.com.midjw...	Battle of Polytopia - A C...	Midjwan AB	The Battle of Polytopia has begu...	2020-06-25	2016-11-30	Fast 4X strategy war game. Buil...	true	GAME
12934595	m.mesoturqisht	Meso Turqisht	Muzik Shqip Falas	Meso turqisht eshte nje aplikaci...	2020-10-12	2020-09-27	Meso turqisht falas nga aplikaci...	true	APPLICATION
129002766	air.com.routa	Dirty Airplane Cleaning	routaless	Having a company that owns air	2020-08-28	2018-01-01	More than cleaning develop your	true	APPLICATION

Rank :

Contient les informations sur le classement de l'application dans les différentes listes (top free, top new free, top grosssing, ect....).

On actualise chaque jour le classement de (top free, top new free, top grosssing, ect...)

id [PK] integer	packagename character varying (955)	datecaptured date	country character varying (955)	category character varying (955)	subcategory character varying (955)	rank integer
12733	com.mojang.minecraftpe	2020-09-13	MA	GAME	apps_topselling_paid	1
12734	com.ubisoft.assassinscreed.i...	2020-09-13	MA	GAME	apps_topselling_paid	2
12735	it.rortos.realflightsimulator	2020-09-13	MA	GAME	apps_topselling_paid	3
12736	com.rockstargames.gtasa	2020-09-13	MA	GAME	apps_topselling_paid	4
12737	com.ninjakiwi.bloonstd6	2020-09-13	MA	GAME	apps_topselling_paid	5
12738	com.robtopx.geometryjump	2020-09-13	MA	GAME	apps_topselling_paid	6
12739	com.marmalade.monopoly	2020-09-13	MA	GAME	apps_topselling_paid	7
12740	com.squareenixmontreal.hitm...	2020-09-13	MA	GAME	apps_topselling_paid	8
12741	com.rockstargames.gtavc	2020-09-13	MA	GAME	apps_topselling_paid	9
12742	com.handelabra.OneDeckDun...	2020-09-13	MA	GAME	apps_topselling_paid	10
12743	nz.co.codepoint.minimetro	2020-09-13	MA	GAME	apps_topselling_paid	11
12744	com.and.games505.TerrariaP...	2020-09-13	MA	GAME	apps_topselling_paid	12
12745	com.chucklefish.stardewvalley	2020-09-13	MA	GAME	apps_topselling_paid	13

Review :

Contient les détails sur les reviews et les ratings des applications. On ajoute chaque jour les nouveaux détails de review tout en gardant les enregistrements des jours précédents dans la base de données.

id integer	packagename character varying (955)	starrating double precision	ratingcount bigint	oneringrating bigint	towrrating bigint	threerating bigint	fourrating bigint	five rating bigint	commentrating bigint	datetraced date
23036	com.mojang.minecraftpe	4.59945821762085	3620079	198423	53406	112772	270533	2984942	1864903	2020-09-13
23037	com.ubisoft.assassinscreed.i...	3.6732611656188965	64883	13297	3539	6658	8955	32431	36772	2020-09-13
23038	it.rortos.realflightsimulator	4.2067108154296875	48485	4820	1727	3093	7813	31031	26136	2020-09-13
23039	com.rockstargames.gtasa	4.527487754821777	538444	38652	8370	16229	42241	432950	244960	2020-09-13
23040	com.ninjakiwi.bloonstd6	4.771256446838379	163322	3218	1163	2947	15099	140892	70153	2020-09-13
23041	com.robtopx.geometryjump	4.59792423248291	738604	35573	9905	25577	73809	593738	337798	2020-09-13
23042	com.marmalade.monopoly	4.243827819824219	36171	3280	1354	2177	5809	23548	15555	2020-09-13
23043	com.squareenixmontreal.hitm...	4.4073967933654785	823922	53635	19605	49645	115609	585425	324599	2020-09-13
23044	com.rockstargames.gtavc	4.244309425354004	153845	18200	3963	7826	15913	107941	63106	2020-09-13
23045	com.handelabra.OneDeckDun...	4.21999979019165	390	31	15	23	85	234	204	2020-09-13
23046	nz.co.codepoint.minimetro	4.623510360717773	43895	1106	593	1770	6779	33645	13310	2020-09-13
23047	com.and.games505.TerrariaP...	4.431775093078613	292893	21012	8426	14041	29017	220395	154742	2020-09-13
23048	com.chucklefish.stardewvalley	4.710607685211600	50000	1203	601	1474	1572	50138	26874	2020-09-13

DownloadInfo :

On suit le nombre de downloads qui appartient à cette série (+1 +5 +10 +50 +100 +500 +1000 +5000 +10000 +50k +100k +500k +1m +5m +10m + 50m + 100m...).

A chaque changement du nombre de downloads, on ajoute un nouvel enregistrement.

id [PK] integer	packagename character varying (955)	datecaptured character varying (955)	downloadt character v...
28859	com.mojang.minecraftpe	2020-09-13	10,000,0...
28860	com.ubisoft.assassinscreed.i...	2020-09-13	1,000,00...
28861	it.rortos.realflightsimulator	2020-09-13	1,000,00...
28862	com.rockstargames.gtasa	2020-09-13	1,000,00...
28863	com.ninjakiwi.bloonstd6	2020-09-13	1,000,00...
28864	com.robtopx.geometryjump	2020-09-13	1,000,00...
28865	com.marmalade.monopoly	2020-09-13	500,000+...
28866	com.squareenixmontreal.hitm...	2020-09-13	10,000,0...
28867	com.rockstargames.gtavc	2020-09-13	1,000,00...
28868	com.handelabra.OneDeckDun...	2020-09-13	10,000+ ...

Image :

Contient les lien des images (screenshot et icon)

id [PK] integer	packagename [PK] character varying (955)	imagetype character varying (555)	imageurl character varying (555)
7017567	com.Markin.FallMen	1	https://play-lh.googleusercontent.com/FRbzGCVvREr603KP44iyFhx83F8pvw...
7017568	com.Markin.FallMen	1	https://play-lh.googleusercontent.com/1XZjaFSUVuu96A-Ln8mhYZddLquRsl...
7017569	com.Markin.FallMen	1	https://play-lh.googleusercontent.com/OC9wnXh00wYbOoNWlxTr-PxDNHK...
7017570	com.freecallmeetings.tyguide	2	https://play-lh.googleusercontent.com/Hrqp88H3H7Q_qSIBuBZmYlqcY8_mE...
7017571	com.freecallmeetings.tyguide	4	https://play-lh.googleusercontent.com/0wzld-Kk5tA8LE0tJtA1ntVj4nKXLyC...
7017572	com.freecallmeetings.tyguide	1	https://play-lh.googleusercontent.com/sR0zvyHSuUJsQD7lv4GuPXPlq07P9...
7017573	com.freecallmeetings.tyguide	1	https://play-lh.googleusercontent.com/EJBYI5U-PZiHW54vosMSmWJE5t8F0...
7017574	com.freecallmeetings.tyguide	1	https://play-lh.googleusercontent.com/lh05566WeOPApzxy-3cEnuu_cG0E8ph...

PermissionInfo :

Table static contient des infos sur chaque permission.

Permissions :

Contient les permissions de chaque application.

Version :

Contient les informations sur la version de l'application. On ajoute un nouvel enregistrement sur l'application à chaque nouvelle version.

IV. Stratégie de Collection des données :

Pour collecter des données, nous avons programmé un script Python(setup) qui collecte des données tous les jours pour que notre base de données reste à jour.

Top new free et top free :

On commence chaque jour par la collecte de la liste des applications de top new free, top free, top new paid, top paid et top grossing pour chaque catégorie. Ce qui nous donne $540 * 4 * 30$ applications. Soit un total de 64800 applications par pays.

Autre apps de même developer :

Puisqu'il y a une grande partie des applications n'est pas classée dans top new free, top free, top grossing etc... On collecte toutes les apps créés par chaque dévelloper. A la fin de cette étape, il est possible de trouver jusqu'au double des applications déjà existantes.

Apps par title/keyword

A partir des deux étapes précédentes, on arrive déjà à construire une base de données importante. Mais nous ajoutons à cela une nouvelle collecte d'application en se basant cette fois sur les titres des apps déjà existantes dans notre base de données. Cette étape nous donne pour chaque titre unique une cinquantaine d'applications supplémentaires.

En répétant en boucle les trois étapes précédentes, on finit facilement avec 3 millions d'applications. Soit presque toutes les apps qui reçoivent de l'engagement et des installations.

❖ Réalisation

I. Environnement de travail :

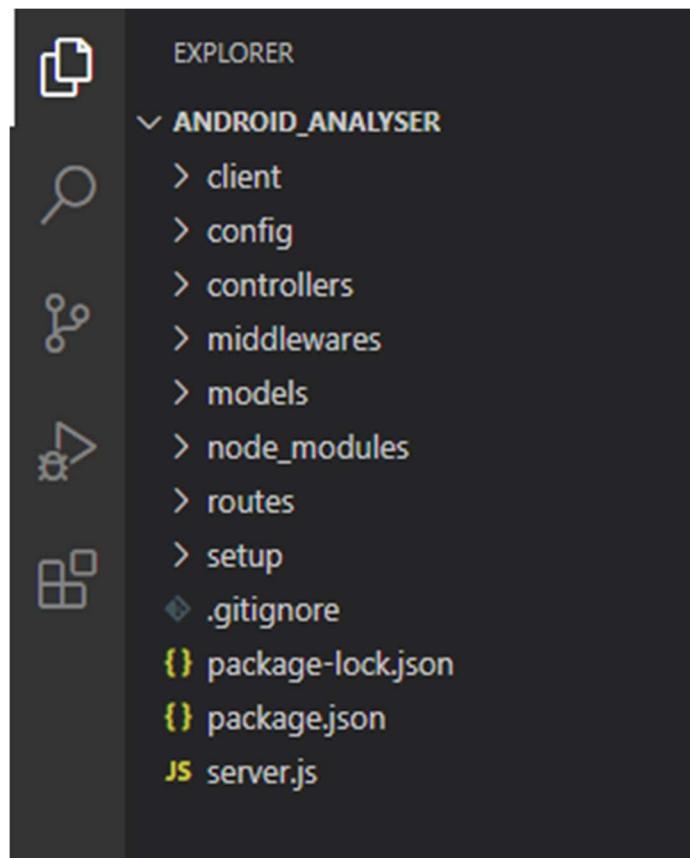
Pour la réalisation de ce projet nous avons utilisé les outils et les technologies suivants :

- [**React**](#) - React est une bibliothèque JavaScript pour créer des interfaces utilisateurs (UI pour la partie frontend).
- [**NodeJS**](#) - Node.js est un environnement d'exécution JavaScript multiplateforme open source basée sur le moteur V8 de Google Chrome qui exécute du code JavaScript en dehors d'un navigateur. Fondamentalement, nous utiliserons ceci pour exécuter notre code javascript sur le serveur.
- [**PostgreSQL**](#) - Il s'agit d'un système de base de données relationnelle objet open source que nous utiliserons pour sauvegarder le contenu de notre Application.
- [**React-Bootstrap**](#) : est une réimplémentation complète des composants Bootstrap à l'aide de React. React-Bootstrap remplace le JavaScript Bootstrap. Chaque composant a été construit à partir de zéro comme un véritable composant React, sans dépendances inutiles comme jQuery. En tant que l'une des plus anciennes bibliothèques React, React-Bootstrap a évolué et grandi avec React, ce qui en fait un excellent choix comme base de votre interface utilisateur.
- [**Express**](#) - Il s'agit d'un cadre d'application Web pour Node.js que nous utiliserons comme serveur pour nos API.

- [Sequelize](#) - Ceci est un ORM (Object Relational Mapper) pour communiquer avec notre base de données PostgreSQL.
- [Postman](#) - Une application Chrome que nous utiliserons pour tester pratiquement notre Application.
- [CORS](#) (Cross Origin Resource Sharing) - Il s'agit d'un package node.js qui permet aux demandes d'ignorer la politique de même origine et d'accéder aux ressources à partir d'hôtes distants.
- [Bcryptjs](#) - Il est utilisé pour le hachage et la comparaison des mots de passe.
- [nodemon](#): Il conduit au redémarrage automatique de notre application de nœud lorsque des changements se produisent dans notre application.
- [PgAdmin4](#) : pgAdmin est le principal outil de gestion Open Source pour Postgres. PgAdmin4 fournit une interface graphique puissante qui simplifie la création, la maintenance et l'utilisation des objets de base de données.

II. Structure du Projet :

Voici la structure de notre projet :



-client : dossier contient le frontend de notre application web.

-config : contient les fichiers .env et config.js,

le fichier **.env** est un endroit où vous stockez votre variable d'environnement, variables qui sont sensibles à votre application, et que vous ne voudriez pas qu'elle tombe entre de mauvaises mains. Il exporte les paramètres de configuration pour la connexion PostgreSQL & Sequelize.
Le fichier **config.js** contient les clés secrètes du tokens d'authentification pour la vérification jwt.

- .gitignore** : on crée le fichier **.gitignore** et on l'ajoute le **.env** et **node_modules** à l'intérieur, cela vous permet de ne pas valider ces fichiers dans votre référentiel git.
- node_modules** : dossier contient toutes les dépendances qu'on a besoin d'importer dans notre projet.
- routes** : contient le fichier **routes.js** pour gestion des routes.
- controllers** : il contient tout le code nécessaire pour implémenter les points de terminaison qu'on va utiliser dans notre App.
- middlewares** : est placé entre le serveur et notre routeur. Il permet d'analyser les corps de requête entrants (vérifier si l'utilisateur est authentifié ou non) aussi de validé les données.
- models** : contient les deux modèles de base de données ORM PostgreSQL dans le projet : **App** et **User**, et deux fichiers pour la connexion à la base données PostgreSQL.
- package.json** : contiens les informations relatives au notre projet : -Name, version, Licences, l'autheur,...
 - Dependencies : liste des modules dont le projet dépend.
- server.js** : Serveur Web Express où nous configurons CORS, initialisons et exécutons les API REST Express de notre application web.

III.Front End de l'Application:

1-Introduction

Pour créer notre GUI nous avons opté pour Reactjs, qui est une librairie open source de javascript créé par Facebook.



Nous citerons ci-dessous quelque avantage que nous procure Reactjs :

-basé sur les composants c'est à dire que chaque élément de UI possède son propre code javascript avec ses propre états (state) ce qui fait qu'ils sont réutilisables, chose qui économise un temps précieux pour les développeurs. De plus, la gestion des erreurs et des changements est considérablement facilitée.

-le concept de DOM virtuelle : bien que cette notion ne soit pas spécifique à Reactjs (Vuejs), le développeur n'est en contact qu'avec ce VDOM et chaque changement qui est fait est comparé avec le DOM réel et donc la balise exacte ou le changement a été effectué est visée pour le changer. Ceci nous évite le fait de parcourir le DOM réel, une opération qui prend beaucoup du temps et de l'énergie.

-Bien que tout le site soit en javascript, ReactJS utilise une syntaxe spéciale appelée JSX qui s'apparente au traditionnel HTML en mixant l'HTML et le Javascript, le développement reste donc familier au développeur web.

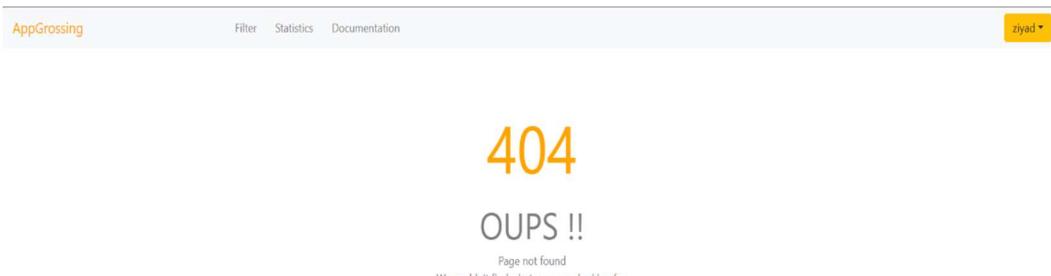
Nous avons décidé de nous passer des librairies qui traitent les états comme Redux car nous avons jugé que le site web n'est pas particulièrement volumineux. On se contentera donc d'utiliser les props ou contexte de Reactjs selon notre besoin. En outre, de ne pas utiliser la nouvelle fonctionnalité hooks qui permet aux composantes fonctionnelles d'accéder aux états et aux propriétés life time. Mais plutôt de rester fidèle au composant fonctionnel et classes normales pour bénéficier d'une approche orientée objet.

2-Description des Pages de l'application APPGROSSING

Nous disposons pour le moment de 4 pages :

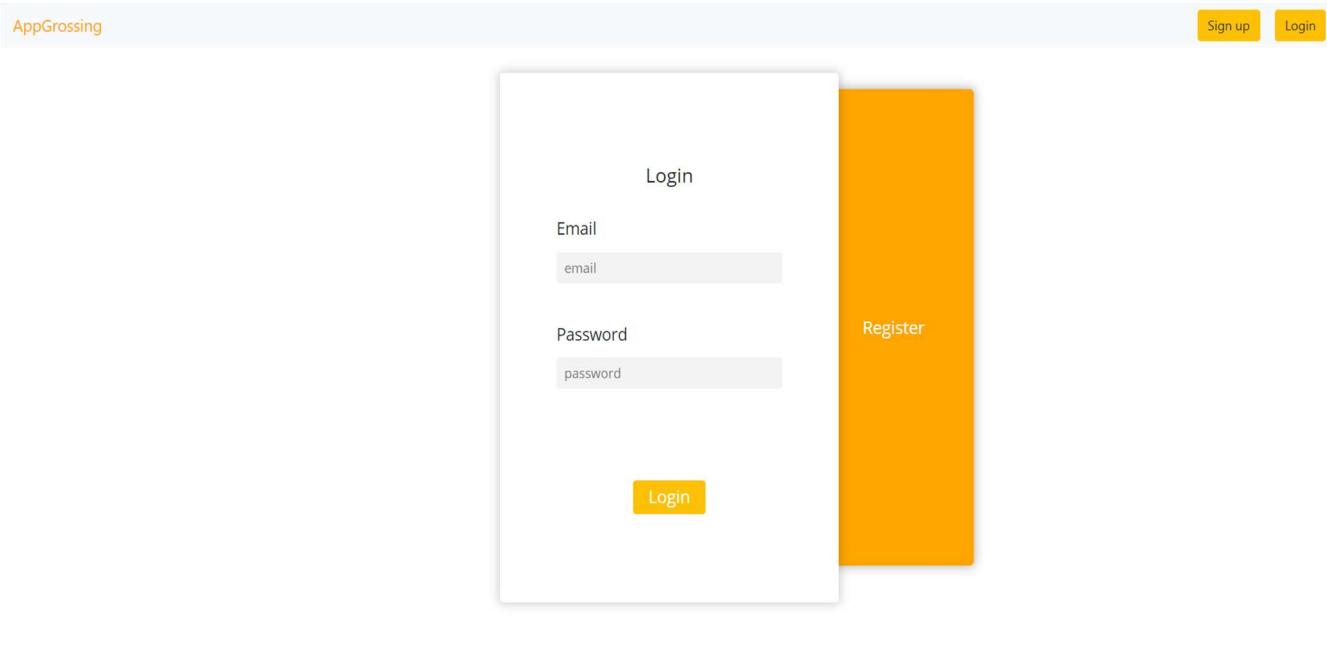
-default Page :

Si l'url n'est pas correcte ou une erreur interne dans le web site arrive ou dans le serveur cette page va être automatiquement afficher.

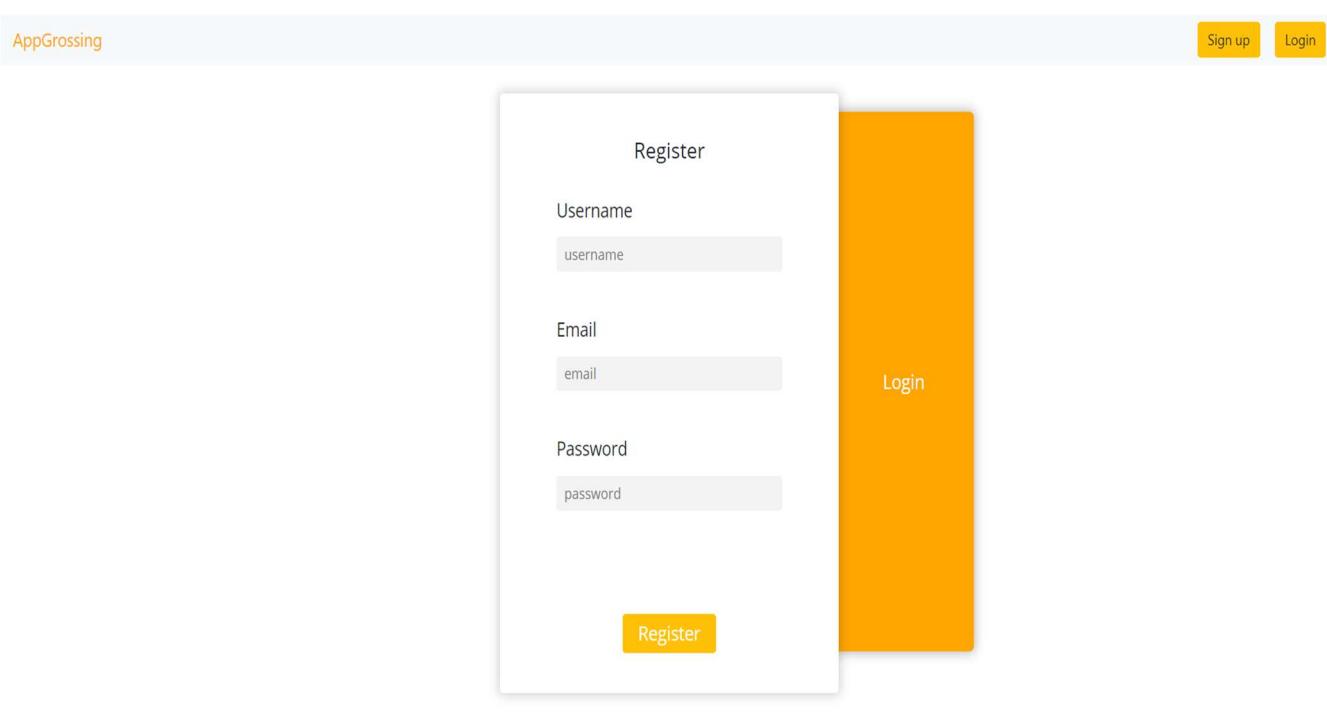


-Login et Register :

email/password authentification, tous les détails concernant l'authentification seront précisés dans la partie backend.



The image shows a login form interface. At the top left is the text "AppGrossing". At the top right are two buttons: "Sign up" and "Login". The main area contains a white rectangular box labeled "Login". Inside this box, there are two input fields: one for "Email" containing the placeholder "email" and another for "Password" containing the placeholder "password". Below these fields is a yellow "Login" button. To the right of the input fields is a vertical orange bar with the word "Register" written on it.



The image shows a register form interface. At the top left is the text "AppGrossing". At the top right are two buttons: "Sign up" and "Login". The main area contains a white rectangular box labeled "Register". Inside this box, there are three input fields: one for "Username" containing the placeholder "username", one for "Email" containing the placeholder "email", and one for "Password" containing the placeholder "password". Below these fields is a yellow "Register" button. To the right of the input fields is a vertical orange bar with the word "Login" written on it.

Le header a deux états selon si l'utilisateur est connecté ou pas : Une fois connecté, l'utilisateur est automatiquement redirigé vers la page Statistics.



-Statistics Page :

Qui représente notre accueil, vous pourrez y trouver toutes les applications avec trois filtres sur le pays, la catégorie des applications ainsi que dans quelle liste ils appartiennent c'est-à-dire (top new free, top free, etc....). Les applications seront présentées sous forme d'un tableau ordonné par leurs classement (Rank).

Playstore rank: Top New Free, categorie:Game, country:Morocco

Ranking	Logo	Title	Downloads	Review	Rating
1		Emoji Puzzle! By Supersonic Studios LTD	5,000,000+ downloads	8039	★★★★★
2		Color Roll 3D By Good Job Games	5,000,000+ downloads	9621	★★★★★
3		Swing Loops - Grapple Hook Race By SayGames	1,000,000+ downloads	3344	★★★★★
4		Lucky Toss 3D By MiniJoy Group	1,000,000+ downloads	9115	★★★★★
5		VIP Guard By Gismart	1,000,000+ downloads	1503	★★★★★
6		My Town : ICEME Amusement Park Free By My Town Games Ltd	100,000+ downloads	2956	★★★★★
7		My Town : Cinema. Fun Movistar Kids Movie Nigh... By My Town Games Ltd	1,000,000+ downloads	7345	★★★★★

Pour chaque application nous présentons son Logo, le titre de l'application et celui du développeur, une estimation du nombre d'installations ainsi que le nombre de reviews (le nombre de personnes qui ont évalué l'application) et le Rating moyen(star rating).

13		Beat EDM : Tiles Beat Hop By KingNote	50,000+ downloads	548	
----	--	--	-------------------	-----	--

Dans chaque page Statistics on retrouve 100 applications.

1 Next

Prev 2 Next

Prev 3 Next

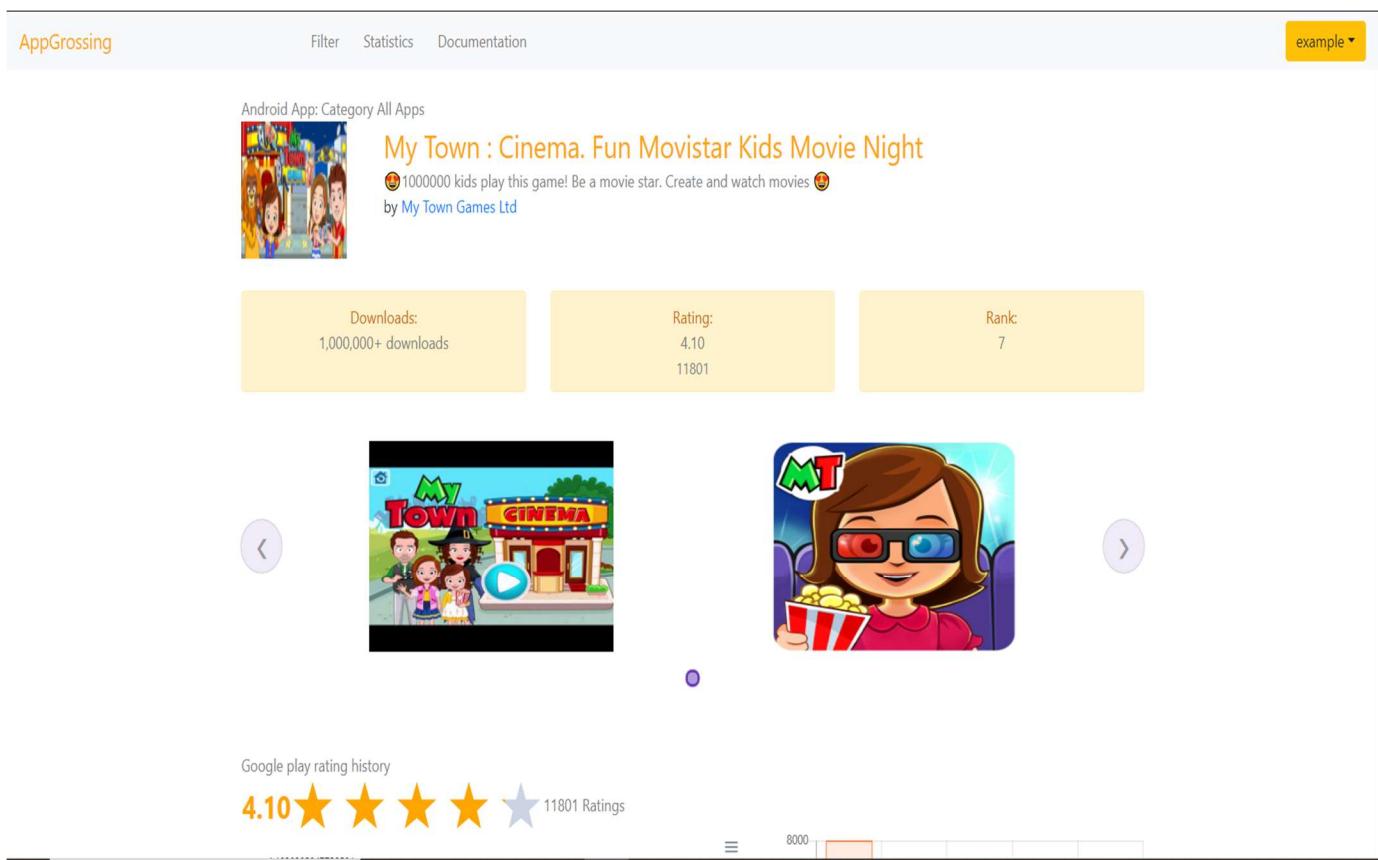
AppGrossing	Filter	Statistics	Documentation	example ▾
333		Merge Space Ships: Cyber Future Merger By NOXGAMES - free big head puppet sports	1,000+ downloads	18
337		Stickman Shooter By AppArtman	1,000+ downloads	69
339		Cooking Cosmetic Box Cake By vstarapps	5,000+ downloads	5
340		Advices Fall Guys Ultimate Knockout By AshCreation	5,000+ downloads	45
341		Brick Builder! By Coda Platform	1,000,000+ downloads	5319
342		New Piggy Scary Roblx's Mod granny By Story Game One	1,000+ downloads	5
344		Love Rescue : Pull Pins and Brain Wash By Hoba Soft	5,000+ downloads	7
345		Real Driving School By Qizz	1,000+ downloads	46
346		Crime Traffic Casual Racing By Oppana Games	5,000+ downloads	9

Prev 3 Next

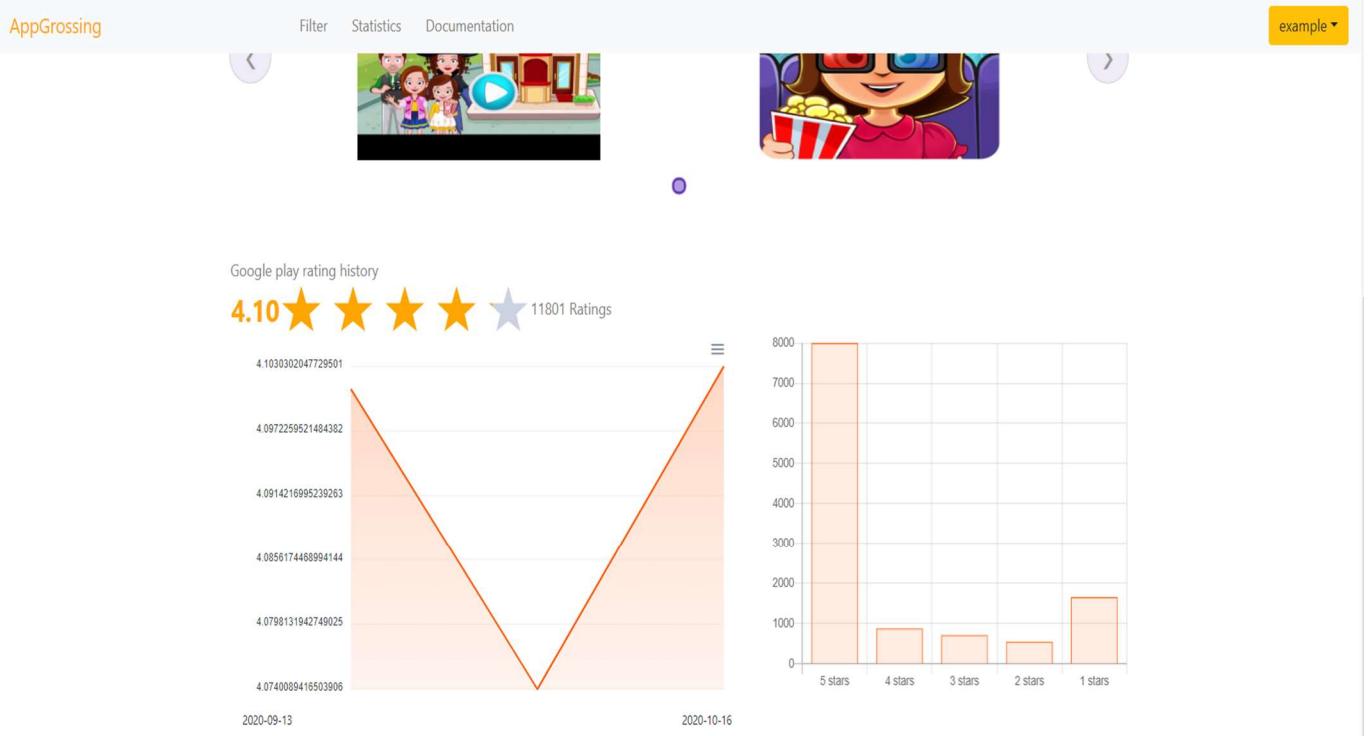
-applications Page :

Cette page vous offre toutes les informations nécessaires sur l'application choisie, accompagnée de graphes qui montrent l'avancement de cette application au fil du temps. Des fonctionnalités de prédictions (en utilisant les enregistrements précédent) peuvent éventuellement être ajoutées à cette page dans des versions à venir.

On y accède en appuyant sur l'application en question sur la page Statistics, Filter ou developper.



Vous y trouverez toutes les informations concernant l'application qui se trouve dans la base de données et qui peuvent se montrer utile pour le développeur de l'application mais aussi pour un potentiel concurrent ou un simple utilisateur. Donc l'amélioration de cette page sera liée au données utile de la base de données sur l'application.



Le graphe à droite est un bar graph qui montre le nombre de personnes qui ont évalué l'application en fonction de leurs choix. Celui à gauche montre l'historique des ratings pendant les 3 derniers jours.

-Developper Page :

Si vous êtes intéressés par le créateur d'une application et voulez savoir des informations à son sujet ainsi que toutes les applications qu'il a développées. Cette page est pour vous.

AppGrossing Filter Statistics Documentation example ▾

Developer Supersonic Studios LTD

App number: 11	Rating Count: 4000	Avr Rating: 4,3
-------------------	-----------------------	--------------------

Apps

Logo	Title	Category	Downloads	Review	Rating
	Wheel Scale!	GAME	5,000,000+ downloads	7391	★★★★★ ★
	Web Hero	GAME_ADVENTURE	1,000,000+ downloads	599	★★★★★ ★
	Emoji Puzzle!	GAME	5,000,000+ downloads	8054	★★★★★ ★
	Join Clash 3D	GAME	50,000,000+ downloads	176403	★★★★★ ★
	Sort It 3D	GAME_PUZZLE	10,000,000+ downloads	57311	★★★★★ ★
	Skater Race	GAME_ARCADE	500,000+	1253	★★★★★

-Filter Page :

Ici nous proposons à nos utilisateurs une série de filtres à leurs dispositions pour ceux qui cherchent des applications spécifiques.

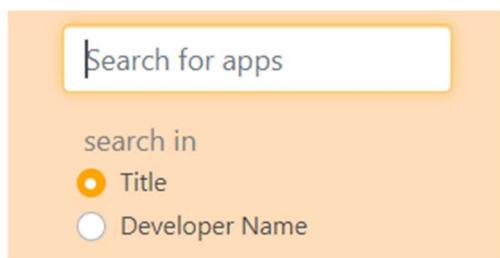
AppGrossing Filter Statistics Documentation example ▾

Search for apps Confirm 100 Apps match

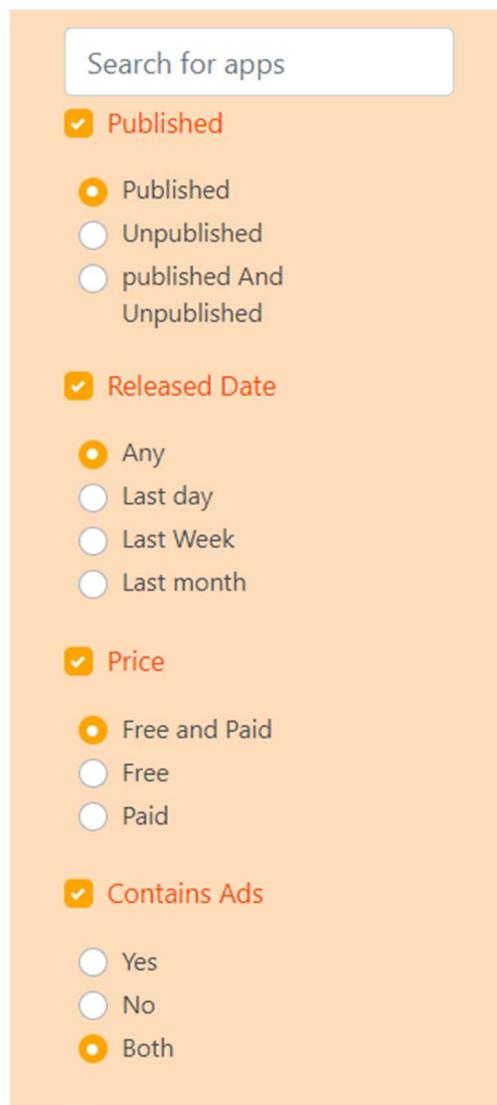
Logo	Title	Category	Downloads	Review	Rating
	Agent Z! By MBC Group	GAME	500+ downloads	15	★★★★★ ★
	Connex : Line Connect Brain Puzzle By Applicants	GAME	500+ downloads	20	★★★★★
	Vacation Family Picnic Adventure By Toon Kids Studio	GAME	500+ downloads	9	★★★★★
	البيوت القاتل By MOORD ;	GAME	500+ downloads	6	★★★★★ ★
	تارجح - مع عاملوں By MOORD ;	GAME	500+ downloads	6	★★★★★ ★
	Perfect Barber shop Hair salon Game By HAK3 Games	GAME	500+ downloads	5	★★★★★ ★
	Find My Spaceship By AM Company	GAME	500+ downloads	5	★★★★★
	royale run 3d! fall guys By Limo Appgms	GAME	500+ downloads	6	★★★★★ ★

Les filtres proposés dans cette version sont :

- Recherche par titre, ou par le dévelloper name de l'application.



- Recherche par le temps de publication, la date de sortie, le prix et si elle contient des pubs ou pas.



- Puis la recherche par le star rating et le nombres de personne qui ont évalué l'application (Le filtre avancé donne plus de liberté et inclut la gestion d'erreur)

Rating
0 < Rating < 5

min= 0 max= 5

0 < Number of Rating < 100M

min= 0 max= 100000000

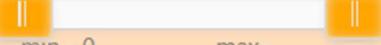
Advanced Filter

Rating
0 < Rating < 5
R> 0
R< 5

0 < Number of Rating < 100M
NR> 0
NR< 100000000

Advanced Filter

- Filtre pour le nombre d'installation de l'app

Downloads
0 < Downloads < 10M

min= 0 max= 10000000

Advanced Filter

Downloads
0 < Downloads < 10M
D> 0
D< 10000000

Advanced Filter

- Et finalement nous retrouvons les filtres qui sont présent dans la page Statistics avec en plus le filtre Rank

Top Charts
List
Top New Free ▾
Category
All Apps ▾
Country
Morocco ▾
Rank

min= 0 max= 600

Advanced Filter

*Chaque filtre peut être soit activer soit désactiver (les filtres désactivés ne seront pas pris en considération dans la requête SQL)

*Toutes les pages ont un design qui se rapprochent en se basant sur l'orange, gris, noire et blanc.

3-Flexibilité & Responsivité

Un site web professionnel se doit d'être flexible afin de s'adapter à toutes les tailles d'écran. Pour s'assurer que notre site web soit responsive, nous avons eu recours au react-bootstrap qui est un front-end Framework très populaire. Ce dernier non seulement propose une panoplie de composants à personnaliser, dont on a fait usage à bon escient, mais aussi offre un système de grid qui facilite la maîtrise de l'emplacement des composants en fonction de la taille de l'écran en question.

Voici un exemple des deux pages Statistics et Filter :

The screenshot shows a web application interface titled "AppGrossing". At the top, there is a navigation bar with a menu icon. Below it, a search bar displays the query: "Playstore rank: Top New Free, categorie:All Apps, country:Morocco". Underneath the search bar are three dropdown filters: "Top New Free", "Morocco", and "All Apps". The main content area features a table with five columns: "Ranking", "Logo", "Title", "Downloads", and "Review". The table lists four apps:

Ranking	Logo	Title	Downloads	Review
1		Emoji Puzzle! By Supersonic Studios LTD	5,000,000+ downloads	8038
1		My Town : Cinema. Fun Movistar Kids Movie Nigh...	1,000,000+ downloads	7345
1		PROCREATE PAINT By Procreated Apps LLC	50,000+ downloads	530
1		Big Boy Toyz By Big Boy Toyz	10,000+ downloads	206

AppGrossing

Search for apps

- Published
- Released Date
- Price
- Contains Ads
- Rating
- Downloads
- Top Charts

Confirm 100 Apps match

Logo	Title	Category	Downloads
	Agent Z! By MBC Group	GAME	500+ download
	Connex : Line Connect Brain Puzzle By Applicans	GAME	500+ download
	Vacation Family Picnic Adventure	GAME	500+ download

4-Le Routage

Les routes sont les url qui font qu'on navigue vers chaque page. Pour cela nous disposons de React-Router qui est la bibliothèque la plus populaire et la plus utilisée pour le routage dans les applications React.

Pour notre cas nous avons utilisé :

- /statistics
- /filter
- /apps/<packageName>
- /developer/<developperName>
- /login
- /register

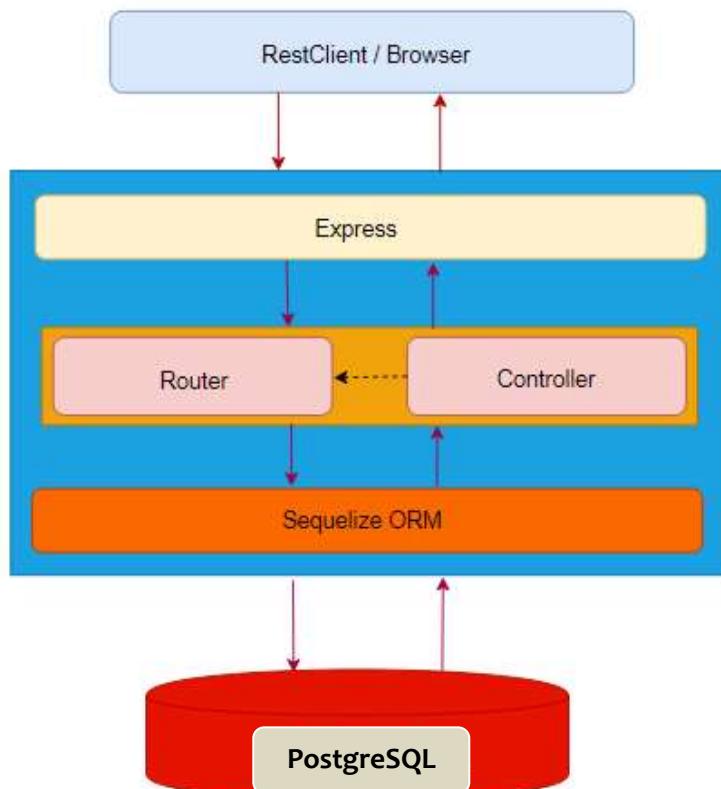
Pour nous assurer qu'aucun utilisateur n'accède à aucune des pages principales avant de se connecter, les routes des pages Statistics, Filter, Developer et Apps sont privées. C'est à dire si on essaie d'y accéder sans être connecté, on est automatiquement redirigé vers la page login, et vice versa.

```
return (
  <div className="App">
    <ErrorBoundary>
      <BrowserRouter>
        <Switch>
          <PublicRoute path="/login" isLoggedIn={true} component={LoginRegister} />
          <PublicRoute path="/register" isLoggedIn={false} component={LoginRegister}/>
          <Route path='/statistics' component={Statistics} exact/>
          <Route path='/statistics/:Plus' component={NotFoundPage}/>
          <Route path='/filter' component={Filter} />
          <Route path='/filter/:Plus' component={NotFoundPage}/>
          <PrivateRoute path='/apps/:title' component={Apps} />
          <PrivateRoute path='/developer/:devName' component={Developer} />
          <Route component={NotFoundPage}/>
        </Switch>
      </BrowserRouter>
    </ErrorBoundary>
  </div>
);
```

IV. Back End de l'Application :

1-Introduction

Depuis son introduction en 2009, Node JS a explosé en popularité et est maintenant la norme de l'industrie pour les systèmes de serveurs basés sur JavaScript. En conjonction avec le Framework Web populaire Express JS, Node a permis aux développeurs de créer des applications Web évolutives, rapides et sophistiquées en utilisant un langage principal à la fois sur le front-end et le back-end: JavaScript . Express est un cadre d'application Web Node.js minimal et flexible qui fournit un ensemble robuste de fonctionnalités pour les applications Web et mobiles.



2-Authentification

Nous avons souvent besoin d'identifier l'utilisateur qui envoie une requête au serveur, et nous devons également assurer la sécurité de la transmission des informations client-serveur. C'est là que les jetons d'accès nous aident, en particulier JWT.

JSON Web Token (JWT) est une norme ouverte (RFC 7519) pour la création de jetons d'accès basés sur le format JSON. En règle générale, il est utilisé pour transférer les données d'autorisation dans les applications client-serveur. Les jetons sont créés par le serveur, signés par la clé secrète et transférés au client, qui utilise plus tard ce jeton pour confirmer son identité.

Pour une sécurité totale, notre JWT doit avoir une durée de vie limitée, et pour sa mise à jour, il est nécessaire que l'utilisateur puisse confirmer son authenticité.

Pour bien comprendre, soulignons deux concepts : **l'authentification et l'autorisation**.

l'Authentification est la vérification des informations d'identification de l'utilisateur (dans notre Application la paire login / mot de passe).

l'Autorisation : donner à certaines personnes ou groupes de personnes le droit d'effectuer certaines actions, c'est-à-dire la procédure permettant de vérifier si un utilisateur a accès à une ressource particulière.

Dans notre projet, le processus d'authentification et d'autorisation est le suivant :

- ✓ L'utilisateur envoie ses données au serveur (login / mot de passe). **Processus d'authentification.**
- ✓ Dans le cas de données correctes, le serveur génère un jeton JWT pour l'utilisateur avec les données nécessaires à l'intérieur (autorisations, ID, durée de vie du jeton).
- ✓ L'utilisateur envoie une demande au serveur avec le jeton reçu.
- ✓ Le serveur décide d'accorder ou non l'accès à l'utilisateur en fonction des données du jeton (autorisations, durée de vie). **Le processus d'autorisation.**

```
13 // Login function
14 exports.login = (req, res, next) => {
15     let getUser;
16     console.log(req.body.email)
17     userSchema.findOne({
18         where: {
19             email: req.body.email
20         }
21     })
22     .then(user => [
23         if (!user) {
24             return res.status(401).json({
25                 message: "Authentication failed: User not found!"
26             });
27         }
28         getUser = user;
29         return bcrypt.compare(req.body.password, user.password);
30     ])
31     .then(isValidPass => {
32         if (!isValidPass) {
33             return res.status(401).json({
34                 message: "Authentication failed : Incorrect password !"
35             });
36         }
37         let jwtToken = jwt.sign({
38             userId: getUser.user_id,
39             dlog: Date.now()
40         }, tconfig.jwt_secret_t, {
41             expiresIn: tconfig.tokenExpireIn
42         });
43         res.status(200).json({
44             // -----
45             _id: getUser.user_id,
46             username: getUser.username,
47             token: jwtToken,
48             expiresIn: tconfig.tokenExpireIn,
49             message: 'Authentication was successful'
50         });
51     })
52     .catch(err => {
53         return res.status(401).json({
54             message: "Authentication failed :" + err.message
55         });
56     });
}
```

3- API Routes

Le routage consiste à déterminer ce qui doit se passer lorsqu'une URL est appelée, ou également quelles parties de l'application doivent gérer une demande entrante spécifique.

Pour séparer les routes de notre fichier principal server.js, nous utiliserons Express.Router() de l'objet express.

```
2  const express = require("express");
3  const router = express.Router();
```

On importe le routeur et on le déclare, avec sa « route racine », comme middleware dans l'application.

```
24
25 // Express APIs
26 const api = require('./routes/routes');
27 // API Routes
28 app.use('/', api);
29
```

NB : dans le routeur, les chemins sont relatifs `a la route racine déclarée pour le middleware.

Bonne pratique : Pour séparer la gestion des routes de leur logique de traitement :

- on a défini un contrôleur pour chaque route dans le dossier controllers. Le contrôleur définit exporte les fonctions de traitement.

- le routeur importe le contrôleur de chaque route pour en utiliser les fonctions.

```
11 // Controllers
12 const authController = require('../controllers/authController');
13 const profileController = require('../controllers/profileController');
14 const statsController = require('../controllers/statsController');
15 const filterController = require('../controllers/filterController');
16 const appsController = require('../controllers/appsController');
17 const devController = require('../controllers/devController');
```

Les URL dont nous aurons besoin pour nos pages sont répertoriées ci-dessous :

*Partie Authentication :

```
20  ****Authentication ****
21
22 // Sign-up
23 router.post("/register", checkRequestRegister(), authController.signup);
24
25 // Sign-in
26 router.post("/login", authController.login);
27
28 // log-out
29 router.post("/logout", authController.logout);
```

*Partie d'Administration :

```
42 // Get Users
43 router.route('/users').get(authorize, profileController.getUsers);
44
45 // Get Single User
46 router.route('/user-profile/:email').get(authorize, profileController.getSingleUser)
47
48 // Update User
49 router.route('/update-user/:email').put(authorize,profileController.updateUser)
50
51 // Delete User
52 router.route('/delete-user/:email').delete(authorize,profileController.deleteUser)
53
```

*le corps de notre Application web :

```
59 // Statistics
60 router.post("/statistics", authorize, checkRequestStats(), statsController.stats);
61
62 // Filter
63 router.post("/filter", authorize, checkRequestFilter(), filterController.filter);
64
65 // appContent
66 router.route('/apps/:pckge').get(authorize, checkRequestApps(), appsController.appsContent)
67
68 // developer
69 router.route('/developer/:devname').get(authorize, checkRequestDev(), devController.devApp)
```

NB : toutes les routes sont protégées.

4-Middlewares

Express est une infrastructure web middleware et de routage, qui a des fonctions propres minimales : une application Express n'est ni plus ni moins qu'une succession d'appels de fonctions middleware.

Les fonctions de **middleware** sont des fonctions qui peuvent accéder à l'objet **Request** (**req**), l'objet **response** (**res**) et à la fonction middleware suivant dans le cycle requête-réponse de l'application. La fonction middleware suivant est couramment désignée par une variable nommée **next**.

Les fonctions middleware effectuent les tâches suivantes:

- ✓ Exécuter tout type de code.
- ✓ Apporter des modifications aux objets de demande et de réponse.
- ✓ Terminer le cycle de requête-réponse.
- ✓ Appeler la fonction middleware suivant dans la pile.

Par exemple le fichier qui exporte la fonction `authorize` est un middleware utilisé pour vérifier si un utilisateur a le droit d'accès à une ressource particulière ou non.

```
1 // middlewares/authorize.js
2 const jwt = require("jsonwebtoken");
3 const tconfig = require('../config/config');
4
5
6 module.exports = (req, res, next) => {
7   try {
8     const token = req.headers.token.split(" ")[0];
9     jwt.verify(token, tconfig.jwt_secret_t);
10    next();
11  } catch (error) {
12    res.status(401).json({ message: "No valid token provided or expired" });
13  }
14};
```

-Si la fonction middleware en cours ne termine pas le cycle de **requête-réponse**, elle doit appeler la fonction **next()** pour transmettre le contrôle à la fonction middleware suivant. Sinon, la demande restera bloquée.

5-Validation des Données

Il est impératif d'effectuer une validation côté serveur lors de la création d'applications, en particulier les applications orientées client. La raison en est que l'on ne peut jamais se fier uniquement à l'entrée de l'utilisateur ; car ces entrées contiennent parfois des données fausses / malveillantes.

La validation côté client est un excellent moyen de passer au crible la plupart des entrées, mais vous devez toujours effectuer une validation côté serveur.

La meilleure façon de gérer la validation sur tout type d'entrée (input) provenant de l'extérieur dans Express est d'utiliser le package **express-validator**. Express-validator est une bibliothèque qui s'enroule validator.js et expose ses fonctions comme un ensemble de middlewares.

On a utilisé la validation des données dans tous les routes de notre application. La figure ci-dessous montre un exemple d'utilisation du package express-validator dans la route '/register'.

```

1  const { check } = require('express-validator');
2
3  module.exports = () => [
4    return [
5      check('username')
6        .not()
7        .isEmpty()
8        .withMessage('Username is required')
9        .bail()
10       .isLength({ min: 3 })
11       .withMessage('Username must be contain at least 3 characters long'),
12     check('email')
13       .not()
14       .isEmpty()
15       .withMessage('Email is required')
16       .bail()
17       .isEmail()
18       .withMessage('Email not valid'),
19     check('password', 'Password should contains 8 characters long')
20       .not()
21       .isEmpty()
22       .withMessage('password is required')
23       .bail()
24       .isLength({ min: 8 })
25   ];
26 ]

```

Chaque appel `check()` accepte le nom du paramètre comme argument. Ensuite, il vérifier est ce qu'il est valide au non. En cas d'erreur, le serveur envoie automatiquement une message d'erreur en utilisant `.withMessage()`. Par exemple si le `username` n'est pas valide, le serveur retourne le message « `Username must be contain at least 3 characters long` »

Le package `express-validator` peut également être utilisé de manière pratique pour effectuer une désinfection (`sanitization`) en utilisant les méthodes `.trim()`, `.customSanitizer()`, ... au paramètres.

```

13   .customSanitizer(value => [
14     value = decodeURIComponent(value);
15     return value.replace(/\//g, "\\");
16   ])

```

6-Base de Données & ORM

Chaque projet Node.js exécuté avec Express.js en tant qu'application Web aura besoin d'une base de données. Étant donné que la plupart des applications serveur sont stateless (sans état), afin de les mettre à l'échelle horizontalement avec plusieurs instances de serveur, il n'y a aucun moyen de conserver les données sans un autre tiers (par exemple, une base de données). C'est pourquoi il est bon de développer une application initiale avec des exemples de données, dans laquelle il est possible de lire et d'écrire des données sans base de données, mais à un moment donné, vous souhaitez introduire une base de données pour gérer les données. La base de données conserverait la persistance des données sur les serveurs ou même si l'un des serveurs n'est pas en cours d'exécution.

Notre base de données **PostgreSQL** est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données (le cas de notre application), même dans des situations critiques.

Pour connecter PostgreSQL à notre application Express, nous utiliserons **Sequelize** comme ORM, car il prend en charge plusieurs dialectes, dont l'un est PostgreSQL. Sequelize fournit une API confortable pour travailler avec les bases de données PostgreSQL de la configuration à l'exécution.

Notre projet implémente une base de données pour notre application web avec deux entités de base de données : User et App.

V.Gestion des Erreurs :

Express est livré avec un gestionnaire d'erreurs intégré, qui prend en charge toutes les erreurs restantes qui pourraient être rencontrées dans l'application. Cette fonction middleware de gestion des erreurs par défaut est ajoutée à la fin de la pile de fonctions middleware. Si vous transmettez une erreur à next() et que vous ne la gérez pas dans un gestionnaire d'erreurs, elle sera gérée par le gestionnaire d'erreurs intégré; l'erreur sera écrite sur le client avec la trace de pile.

Les erreurs sont gérées par une ou plusieurs fonctions middleware spéciales qui ont quatre arguments, au lieu des trois habituels : (err, req, res, next). Par exemple dans notre projet :

```
43
44 app.use(function (err, req, res, next) {
45   console.error(err.message);
46   if (!err.statusCode) err.statusCode = 500;
47   res.status(err.statusCode).send(err.message);
48 });
49
```

Ceux-ci peuvent renvoyer n'importe quel contenu requis, mais doivent être appelés **après tous les autres app.use()** et acheminer les appels afin qu'ils soient le dernier middleware dans le processus de traitement des demandes!

Les réponses issues d'une fonction middleware peuvent être au format de votre choix, par exemple une page d'erreur HTML, un simple message ou une chaîne JSON.

Conclusion

Tout au long de la préparation de notre projet, nous avons essayé de mettre en pratique les connaissances acquises durant nos études et cela dans le but de réaliser une application web réussie.

Le résultat de ce travail est donc une application web réalisée dans le respect de plusieurs règles de bonne pratique de la programmation web. Parmi ces bonnes pratiques, on retrouve l'optimisation des données envoyées, l'utilisation des frameworks ou encore l'utilisation d'un design simple et adapté au web. Pour compléter ces bonnes pratiques et autres démarches adoptées, l'accent a été mis sur le choix de plusieurs outils connus pour permettre une maintenance plus efficace. En effet, la maintenance d'une application représentant la plus grande partie de son temps de vie, et les gains pour une application de cette taille n'étant visibles que sur le long terme, il est important de permettre à l'application d'être maintenue facilement et rapidement.

Ce travail est une expérience très épanouissante et enrichissante. Travailler dans une équipe dynamique et créative, pouvoir mettre en pratique le savoir que nous avons acquises sont autant de choses positives que nous en avons retiré.

Webographie

- <https://expressjs.com/en/guide/routing.html>
- <https://www.postgresqltutorial.com/>
- <https://expressjs.com/en/advanced/best-practice-security.html>
- <https://flaviocopes.com/express-validate-input/>
- <https://nodejs.org/en/>
- <https://react-bootstrap.github.io/getting-started/introduction>
- <https://fr.reactjs.org/tutorial/tutorial.html>