

---

# Présentation des spécifications techniques

---

# Login

## — Firebase AUTH :

- API proposé par **Firestore** (plateforme de développement appartenant à Google)
- Permet **connexion avec envoi mail** à l'adresse mail renseignée par utilisateur
- **Facilite la gestion de l'authentification** :
  - Stock les informations d'identification
  - Gère toutes les étapes du processus d'authentification
- **Répond le mieux au besoin** défini
- Documentations importantes et bien faite avec exemples de codes

## — VS Google Sign In :

(maintenant : Google Identity Services for Web)

- Cette API proposée par Google permet de connecter l'utilisateur à son compte google directement
- Simple d'utilisation et sécurisé
- Mais on ne passe pas par un envoi de mail tel que le montre la maquette FIGMA
- **CONCLUSION : Ne correspond pas au projet**

## — React-spin :

- Pour l'animation de l'icône d'attente de validation d'authentification
- **Bibliothèque React d'animation de composants**



# Sécurité de l'authentification utilisateur

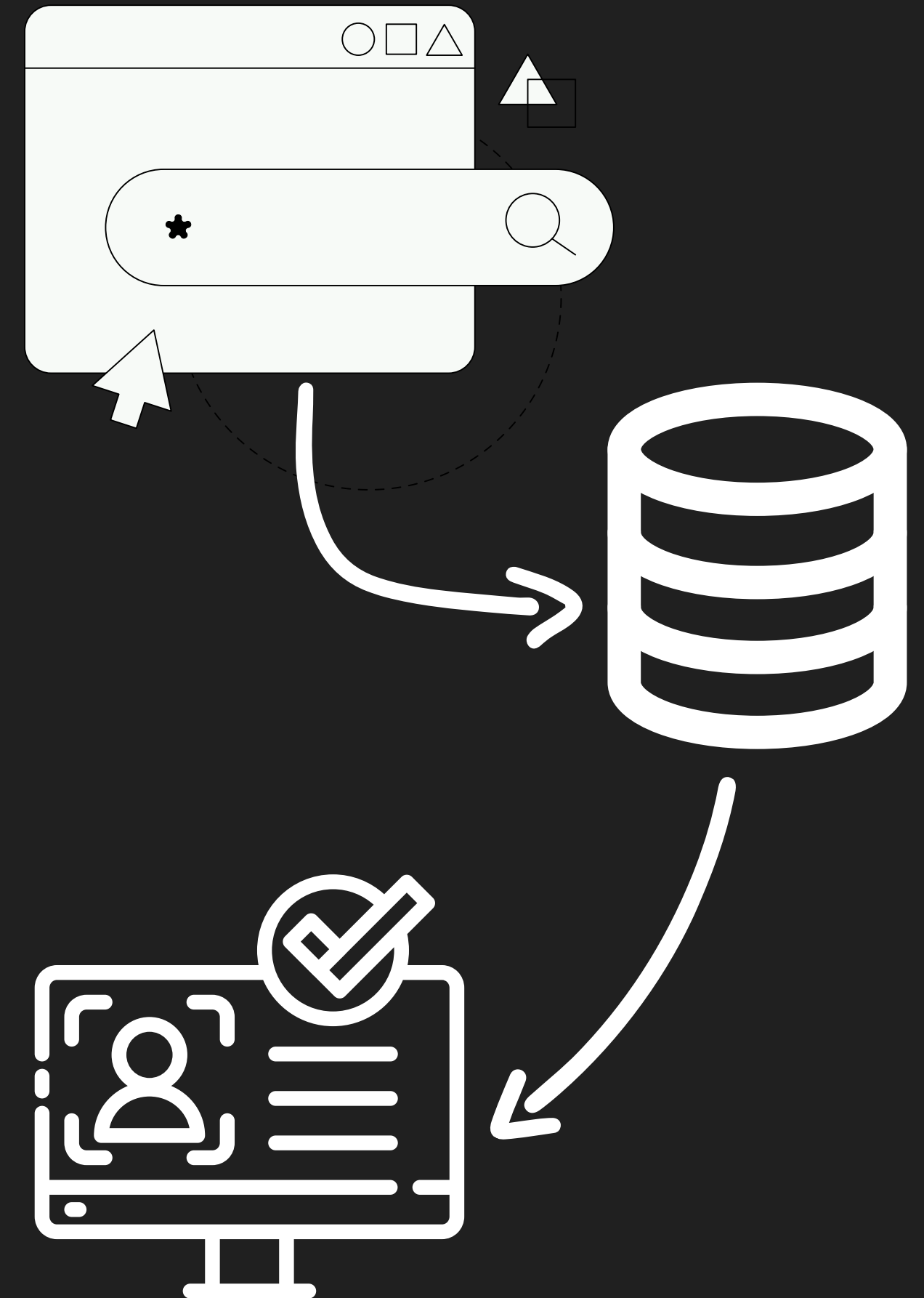
## — JsonWebToken et Bcrypt :

- Permet de **crypter données utilisateurs**
- **Sécurise l'authentification** des utilisateurs
- Outils populaires



## — L'API de Cloudmersive :

- Permet de **vérifier la validité d'une adresse mail** avant de poursuivre le script
- Facile à utiliser grâce à une documentation complète
- Nombreuses fonctions
- MAIS : représente un coût



# Modal et redirection page

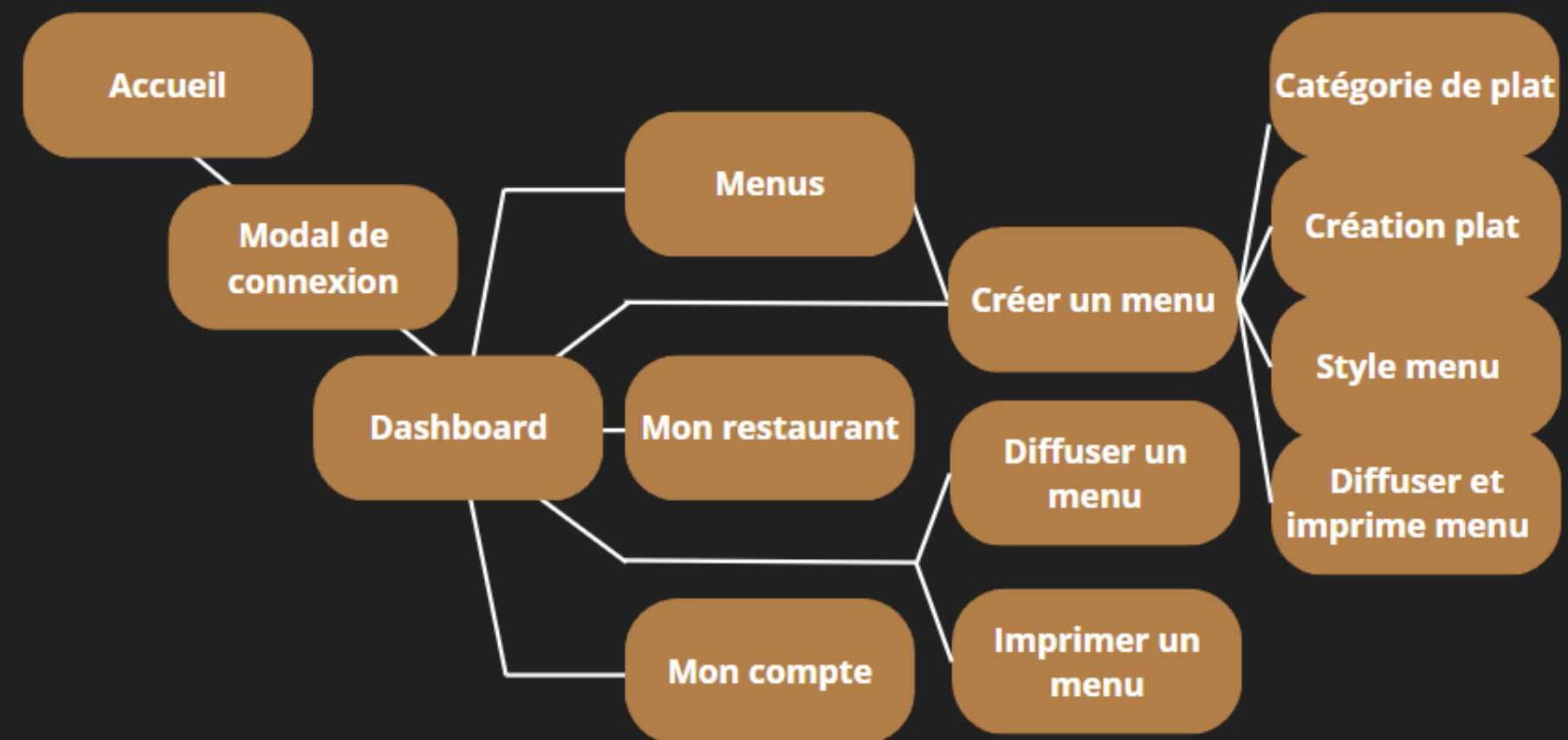
## React Modal :

- bibliothèque simple d'utilisation pour **afficher modale** et personnaliser le style fenêtre (propriétés CSS)



## React Router :

- bibliothèque pour **gérer la navigation** dans le site.
- Redirige l'utilisateur sur une nouvelle route (lors interaction avec éléments)





## — Choix du framework React

1. **Facilité d'utilisation** : facile à apprendre et à utiliser (utilise JavaScript et concepts de programmation orientée objet = familier)
2. **Performance élevée** : rendu des interfaces utilisateur
  - "Virtual DOM" = technique de rendu virtuel qui minimise le nombre de mises à jour nécessaires pour afficher une page



## — VS Vue.js :

- Sensiblement les mêmes points positifs
- MAIS :
  - **REACT plus populaire** : plus grande communauté, plus de nouveaux outils, bibliothèques et extensions
  - **REACT plus personnalisable** : plus de liberté dans la structure de l'application
  - Code **REACT plus prévisible et compréhensible** : grâce à son flux de données unidirectionnel
  - **REACT plus performant**

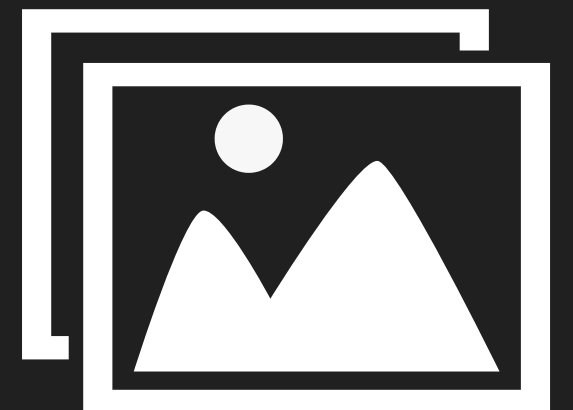
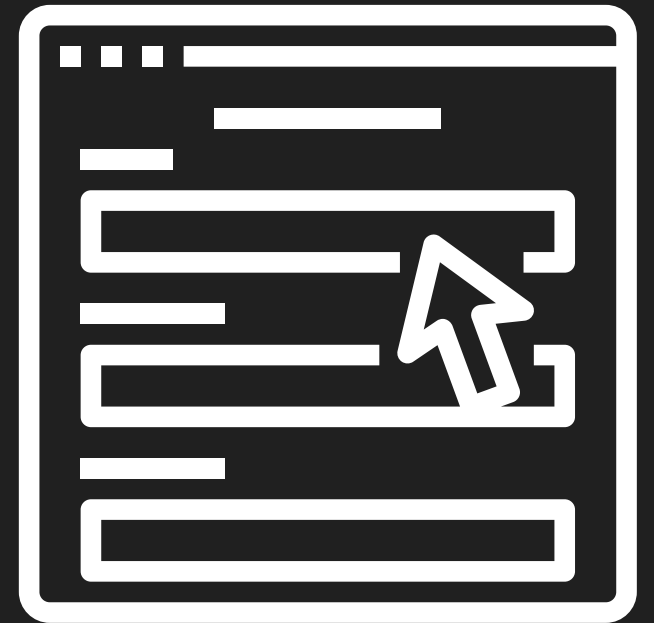
# Ajouter plats et menus

## — React-Hook-Form :

- Bibliothèque de gestion de formulaires (validation des champs, gestions de erreurs...)
- Récente, performante, légère et flexible, syntaxe simple et intuitive
- Se concentre sur la collecte des données de formulaire VS la bibliothèque Formik (gestion des états de formulaire)

## — Module Multer

- Module pour le gestionnaire de packages Node.js
- Gère les fichiers téléchargés dans une application Express
- Traite les données de formulaire de manière multipart/form-data, en extrayant les fichiers téléchargés et en les stockant dans un répertoire spécifié.
- Permet de définir des limites de taille, de types, de noms et autres



# Gestion de la création du PDF

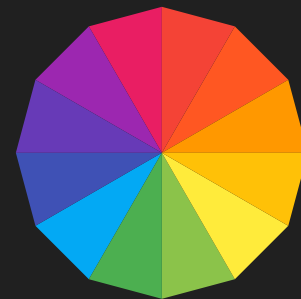
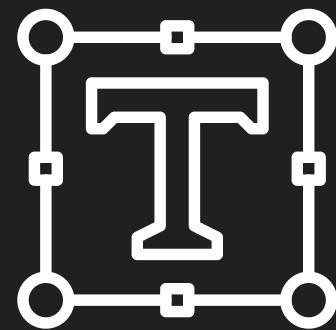
---

## — React-PDF :

- Bibliothèque JavaScript pour générer des fichiers PDF
- Bibliothèque performante et populaire
- Mise à jour très régulière (ajouts nouvelles fonctionnalités)
- **VS l'utilisation de Chromium et la bibliothèque Puppeteer** (bibliothèque Node.js qui permet d'interagir avec Chromium) qui aurait multiplié les outils

## — Google Fonts et React Color

- **Google Font** : propose une large gamme de typographies, simple d'utilisations et populaire
- **React Color** : fournit une gamme de composants de **sélecteurs de couleurs** (de basse, prédéfinies, personnalisés...).



# Gestion du téléchargement et l'impression du PDF

---

## File-saver

- Bibliothèque JavaScript qui fournit une API pour enregistrer des fichiers côté client.
- Facilité d'utilisation, populaire
- Compatibilité avec beaucoup de navigateur
- Pas besoin de base de données car se télécharge directement sur le navigateur de l'utilisateur





# Gestion de la sauvegarde des données

- Menus
- Champs du formulaires
  - catégories de plat
  - plats
- Style du menu
- Infos données utilisateurs

## — Node.js

- Environnement d'exécution côté serveur pour les applications web
- Construit sur le moteur JavaScript
- Facile à apprendre et populaire

## — Basse de données SQL

- Langage informatique utilisé pour gérer des bases de données relationnelles
- **VS** Bases de données SQL non relationnelles

## — Express

- Framework open-source pour le développement d'applications web côté serveur en Node.js



**Pour créer une API BACKEND :**

interagir avec la base de données

# Structure basse de données relationnelle

- **Informations connexions**

- id utilisateur : identifiant unique
- email de l'utilisateur



- **Informations profil**

- Restaurant :
  - id restaurant
  - Nom
  - Adresse, Ville, Code postal...
  - Type de restaurants, logo
  - id utilisateur



- **Menus**

- id menu
- Nom du menu
- Typographie et couleur choisies
- date de création
- id restaurant

- **Catégories**

- id catégorie
- Nom de la catégorie
- id menu

- **Plats**

- id plat : idem
- Nom du plat, description
- Photo, prix
- id catégorie



