TP3: Angular -
réalisé par Ameni Lahouar, ICE-4

PARTIE2 :

GRAND 1:

3. Parcourir ces différents dossiers et fichiers et compléter cette description:

Le dossier `src/` contient : le code source de l'application (composants, services, styles, templates…).

Le dossier `node_modules/` contient : toutes les dépendances et bibliothèques installées via npm.

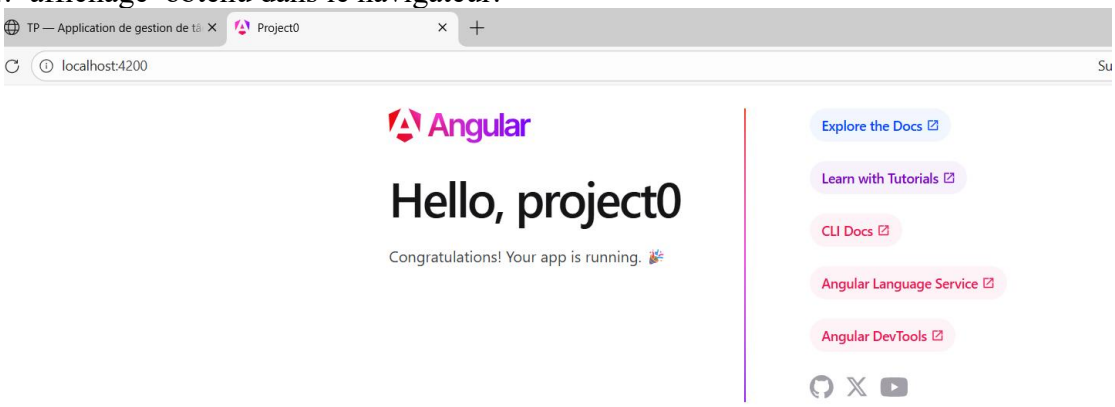
Le fichier `angular.json` contient : la configuration de l'application Angular (build, test, assets, styles…).

Le fichier `package.json` contient : la liste des dépendances, scripts et métadonnées du projet.

Le fichier `package-lock.json` fournit : les versions exactes des dépendances installées pour garantir la cohérence.

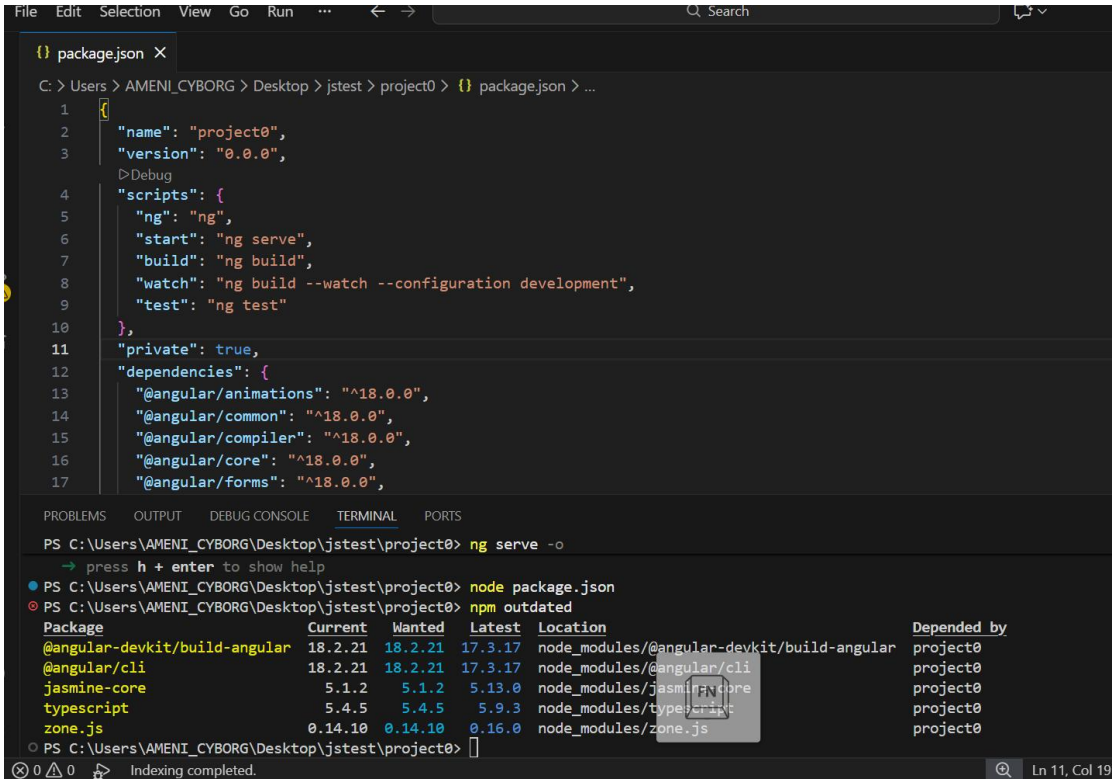
Le fichier `tsconfig.json` contient : la configuration du compilateur TypeScript pour le projet.

4. affichage obtenu dans le navigateur:



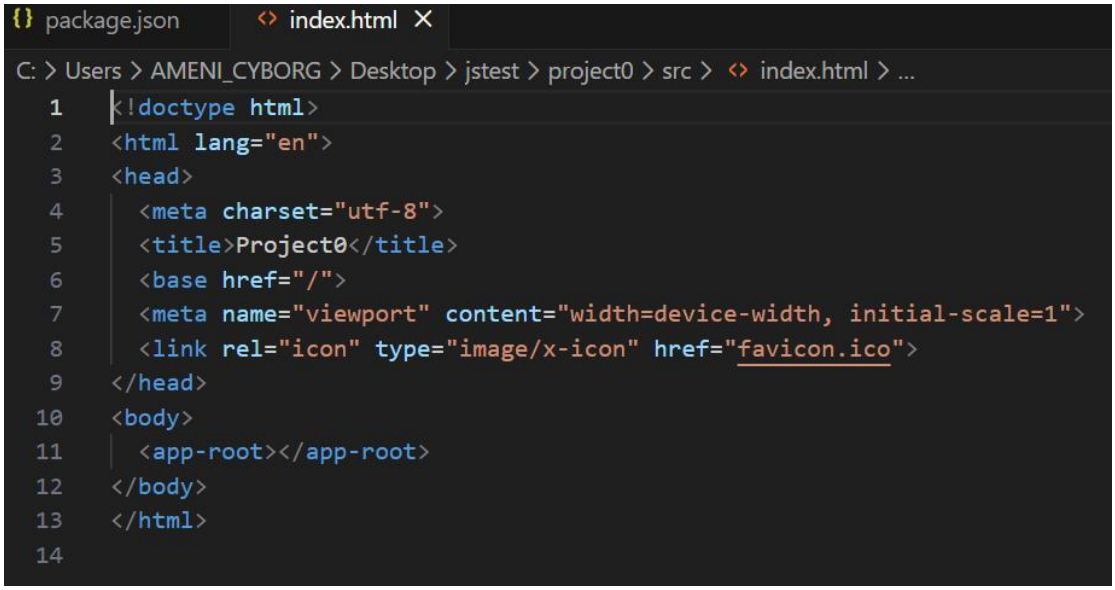
GRAND 2:

2-



->>>Toutes les dépendances peuvent être mises à jour à l'exception de typescript.

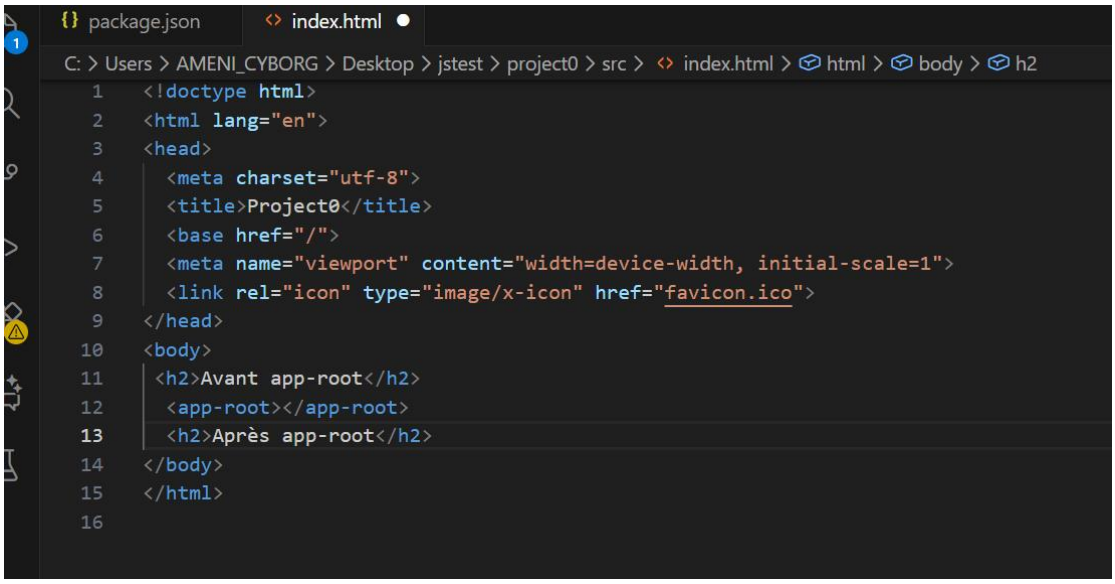
3-index:



GRAND 3:

3-<app-root> est le **sélecteur du composant racine** défini dans app.component.ts; Angular remplace **tout le contenu de** <app-root> par le template de app.component.html.

Comme test, on va modifier index:



Et puis on lance l'application.

->>> Angular ne touche **que la balise** <app-root>.

Tout ce qui est en dehors (<h2> avant et après) reste visible dans le DOM.

Le style et la logique applicative viennent **uniquement du composant** `AppComponent`.

GRAND 6: Validation des acquis:

Partie 1 : QCM sur les outils de développement

1.Quelle est la principale fonction de Node.js dans le développement Angular ?
Réponse : a) Node.js est un environnement d'exécution pour le code JavaScript côté serveur. Il permet d'exécuter JavaScript en dehors du navigateur, ce qui est essentiel pour Angular CLI et la gestion des dépendances.

2.Quelle commande permet d'installer Angular CLI globalement sur votre système ?
Réponse : c) `npm install -g @angular/cli` — Cette commande installe Angular CLI sur tout le système, ce qui permet de créer et gérer des projets Angular.

3. Parmi les éditeurs de code suivants, lequel est recommandé pour le développement Angular en raison de ses extensions et fonctionnalités intégrées ?
Réponse : b) Visual Studio Code — VS Code offre des extensions, le support TypeScript et un débogage intégré, ce qui le rend idéal pour Angular.
4. Quel fichier dans un projet Angular contient les dépendances du projet et est utilisé par npm pour gérer les packages ?
Réponse : b) `package.json` — Ce fichier liste toutes les dépendances et les scripts npm nécessaires au projet.
5. Quelle est la commande pour créer un nouveau projet Angular nommé "ma-app" en utilisant Angular CLI ?
Réponse : c) `ng new ma-app` — Cette commande génère un nouveau projet Angular avec la structure de fichiers de base.
6. Quel est le rôle de la commande ng serve ?
Réponse : b) Lancer un serveur de développement et recharger l'application lors des modifications — Elle compile le projet et ouvre un serveur local pour voir les changements en temps réel.

Partie 2 : QCM sur la création d'un projet Angular

1.Quel répertoire contient les fichiers sources de l'application Angular que vous allez développer ?
Réponse : a) `src/` — C'est là que se trouvent tous les fichiers TypeScript, HTML et CSS de votre application.

2.Dans quel fichier définissez-vous les modules, composants, services et autres éléments qui font partie de votre application Angular ?
Réponse : b) `app.module.ts` — Ce fichier centralise les déclarations et importations des composants et modules nécessaires à l'application.

3.Quel est le rôle du fichier angular.json dans un projet Angular ?
Réponse : b) Définir les configurations de compilation, les entrées et sorties du projet — Il indique comment Angular CLI doit construire et servir l'application.

4.Quelle commande Angular CLI utilisez-vous pour générer un nouveau composant nommé header ?
Réponse : c) `ng generate component header` — Cette commande crée automatiquement le dossier du composant avec les fichiers TypeScript, HTML, CSS et le module associé.

5.Quel est le rôle du fichier tsconfig.json dans un projet Angular ?
Réponse : a) Configurer le compilateur TypeScript pour le projet — Il définit les options de compilation comme le ciblage ES6, le module, et la vérification de types.

Partie 3: Rapport Comparatif : Angular vs React vs Vue.js

1. Angular

Caractéristiques principales : Framework complet, basé sur TypeScript, offre MVC, modules, composants, services et CLI puissante.

Avantages : Très structuré, idéal pour les grandes applications, support officiel Google, outils intégrés (routing, forms, HTTP).

Inconvénients : Courbe d'apprentissage plus raide, plus lourd que les autres frameworks.

2. React

Caractéristiques principales : Bibliothèque JavaScript pour construire des interfaces utilisateur, basé sur les composants, utilise JSX.

Avantages : Léger, flexible, énorme écosystème, facile à intégrer dans des projets existants.

Inconvénients : Nécessite souvent des bibliothèques supplémentaires pour routing ou state management, moins structuré.

3. Vue.js

Caractéristiques principales : Framework progressif, facile à apprendre, basé sur composants, réactif.

Avantages : Simple et intuitif, documentation claire, léger, facile à intégrer.

Inconvénients : Écosystème moins mature que React et Angular, moins de grandes entreprises l'utilisant.

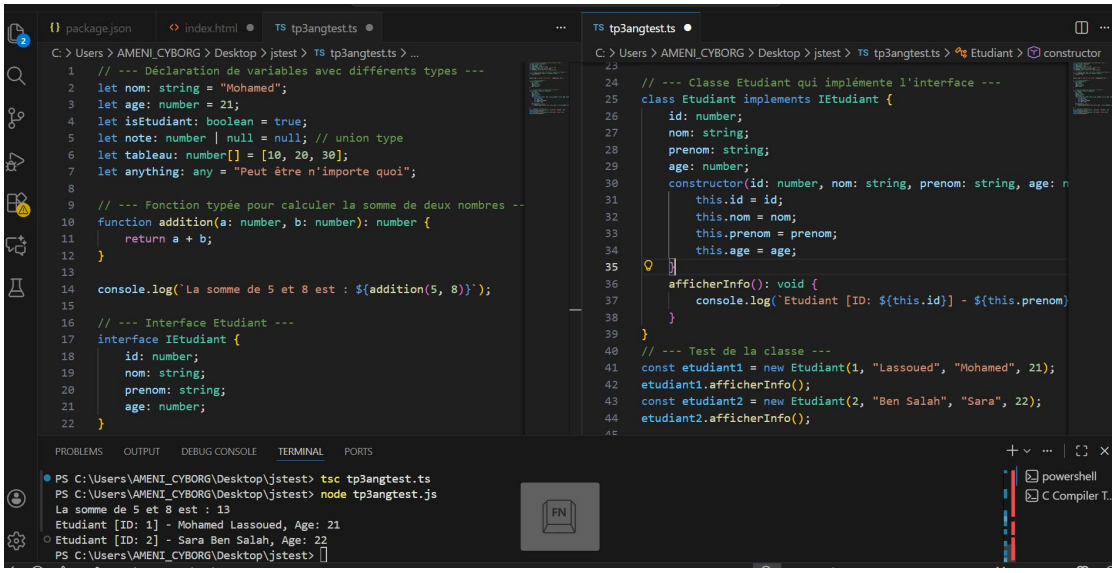
Différences clés :

Angular est un framework complet et strict, React est une bibliothèque centrée UI, Vue est léger et flexible.

Angular utilise TypeScript par défaut, React utilise JavaScript/JSX, Vue permet les deux.

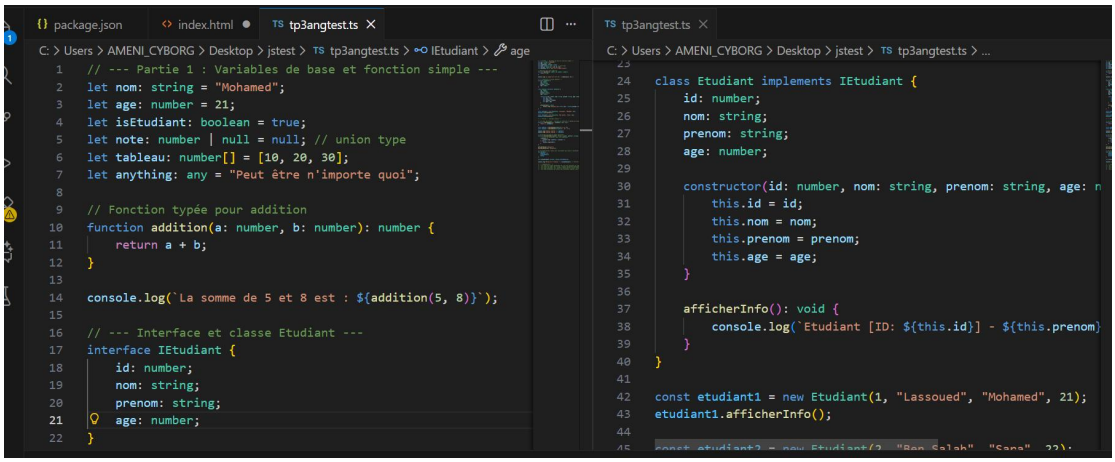
Angular est plus adapté aux grandes applications complexes, Vue et React sont plus adaptés aux projets rapides ou modulaires.

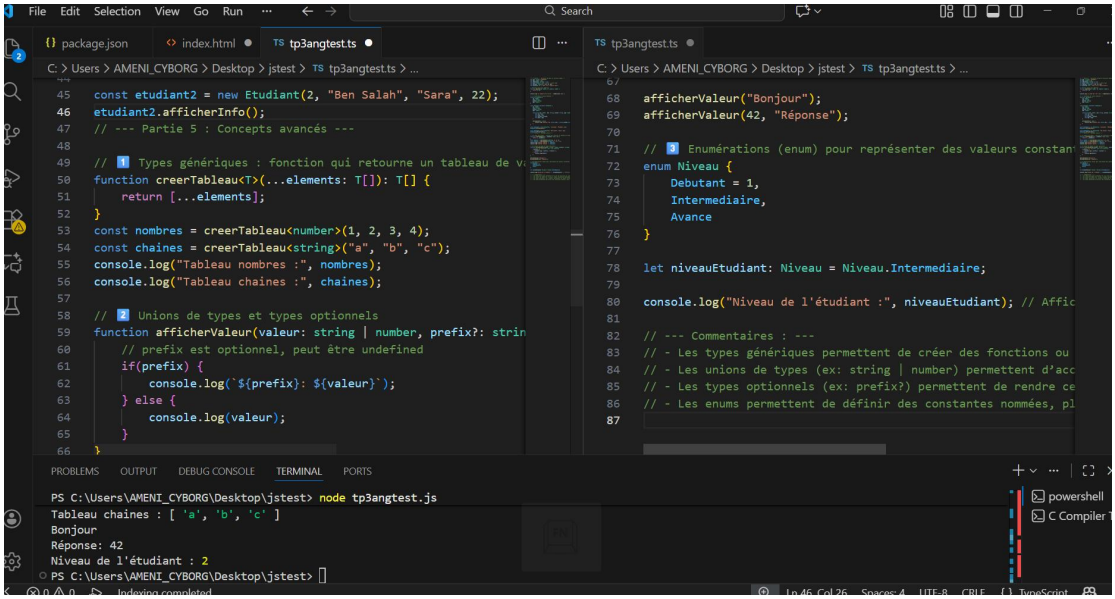
Partie 4:Révision des bases de TypeScript:



On a executé la commande: tsc file.ts qui a genere file.js, qu'on execute enfin pour obtenir les resultats.

Partie5:





- Explications de choix:

`creerTableau<T>` utilise un **type générique** pour créer un tableau de n'importe quel type.

`afficherValeur` montre comment utiliser **unions de types** et **paramètres optionnels**.

`enum Niveau` est une manière propre de représenter des constantes nommées pour le niveau d'un étudiant.