

Matricule ..... Nom ..... Prénom ..... Groupe.....

**Exercice 01:** Compléter le code du projet Angular suivant :

```
<header>GESTION DU RESTAURANT </header>
<nav><a .....="/inscription">Inscription</a>
<a .....="/contact">CONTACT</a></nav>
//Charger le composant qui contient la liste des plats
<section>.....</section>
<div> <router-outlet></router-outlet> </div>
```

**app.component.html**

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ContactComponent } from './contact/contact.component';
import { InscComponent } from './insc/insc.component';
const routes: Routes = [
//Configuration des liens NOS PLATS et CONTACT
(path: " .....", component: ..... },
(path: " .....", component: ..... )];
```

**app-routing.module.ts**

```
<h1>NOS PLATS:</h1>
<table> <tr><th>Plat<th>Prix
<tr
*ngFor="....."
<td>.....
<td>.....<
</tr> </table>
```

**plats.component.html**

```
import { Component, OnInit } from '@angular/core';
@Component({ selector: 'app-plats',
templateUrl: './plats.component.html',
styleUrls: ['./plats.component.css'] })
export class PlatsComponent implements OnInit {
public
plats=[{nom:"palt1",prix:220},{nom:"plat2",prix:400}]
}
```

**plats.component.ts**

```
<form #insc="ngForm"> Login:<input type="text" required name="login" ngModel #.....="ngModel"/>
<div *ngIf=".....errors && (.....touched)">
<small class="text-danger" *ngIf=".....errors?.['.....']">ce
champs est obligatoire</small></div>
Saisir votre mot de passe:<input type="password" name="pwd" ngModel #.....="ngModel"
pattern="[a-z0-9._&+-]+@[a-z0-9.-]"/>
<div *ngIf=".....errors && (.....prestine)">
<small class="text-danger" *ngIf=".....errors?.['.....']">Saisi
un mot de passé valide</small></div>
<input type="submit" [disabled]="!insc.valid"/></form>
```

**insc.component.html**





**Exercice 02 :** On veut construire une SPA permettant de gérer un restaurant. Compléter le script JS suivant :

```
var express=require("express"); var app=express(); var mongoose=require("mongoose");
var db=mongoose.connect("mongodb://localhost:27017/mabase");
var bodyParser = require("body-parser"); var bp=bodyParser.urlencoded({ extended: false });
var schemaClient=mongoose.Schema({nom:String,
plat:{type: mongoose.Schema.Types.ObjectId, ref:.....}});
var schemaPlat=mongoose.Schema({ nom:String, prix:Number});
var client=mongoose.model("clients",.....);
var plat=mongoose.model("plats",.....);
app.....('/client/ajouter',bp, function(req,res){
    var n=req.body.nom; var plat=req.body.plat;
    var c=new .....({.....});
    c.....(function (err, ins) {
        if(err){ console.log("erreur")}else{ res.json(ins)}; });
//Afficher les noms des clients qui ont demandé une pizza vegetarienne
app.get('/q2', function(req,res){
    .....aggregate([
        $lookup: {
            from: ".....",
            localField: ".....",
            foreignField: ".....",
            as: "lp"},
        {$.....: {
            .....//matching
        }},
        {$project: {
            .....//projection}}
    ]).exec(function(err, l) { res.json(l)});
//Modifier les prix des plats spécifiés par l'utilisateur dans l'url paramétrées.
app.get("/plat/:nom/prix/:prix", function(req,res){
    .....find(.....).then (function (p){
        p.prix=.....; //récupérer le prix spécifié dans l'url
        p.....;}); //sauvegarder });
//Supprimer le client spécifié par l'utilisateur dans l'url paramétrées.
app.delete("/client/:id", function(req,res){
    .....then (function (p){
        res.json(p); }); });
app.listen(3000); console.log("serveur lancé sur le port 3000");
```