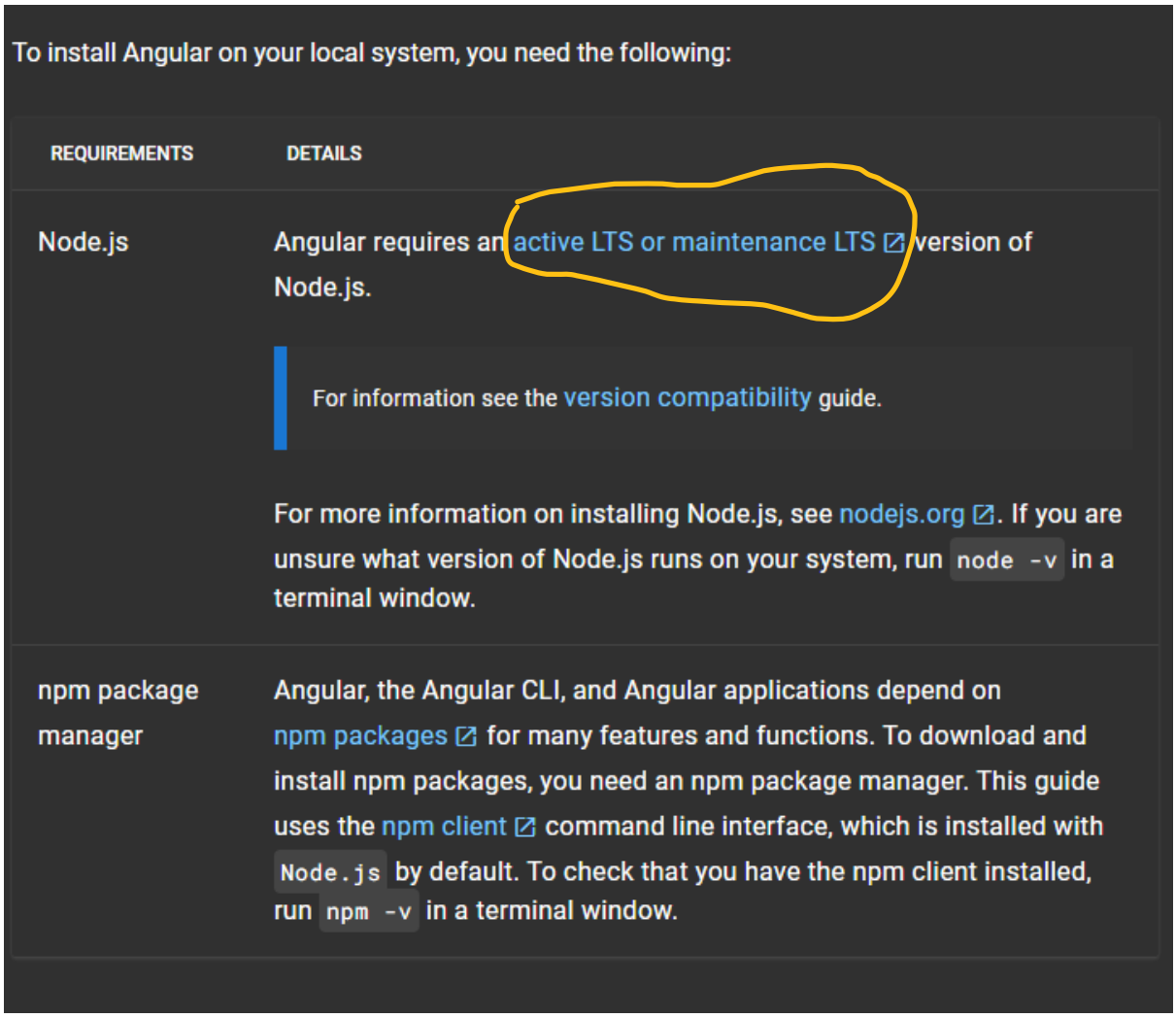
Assignment 2C

Here are the steps on how to create an Angular application which can register a user, log in a user, and show user data:

1. First of all Angular web sit then go to menu then click to getting start then again go to setup and after scroll down then you see this and click on it and download node.js and after that (To install the Angular CLI, open a terminal window and run the following command: > npm install -g @angular/cli)



2. Create a new Angular application using the Angular CLI:

   >ng new ass_2_c

3. Install the Angular Material UI library:

   > npm install @angular/material @angular/cdk @angular/animations

   his command installs Angular Material, along with its dependencies such as Angular CDK (Component Dev Kit) and Angular Animations. Angular Material is a UI component library for Angular applications. It provides a set of reusable and accessible UI components based on the Material Design principles. These components include things like buttons, cards, menus, dialogs, forms, and more. By using Angular Material, developers can quickly build attractive and consistent user interfaces for their Angular applications without having to write CSS or JavaScript from scratch. It promotes a consistent design language and helps maintain a cohesive look and feel across different parts of the application.

4. Go to index.html inside src folder add bootstrap link in header element

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/boo
tstrap.min.css" rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW
3" crossorigin="anonymous">
```

5. Go to director inside project folder using cd command

   >cd ass_2_c

6. Create a new component for the user form:

   >ng g component user

7. Go to app-routing.module.ts inside app folder import UserComponent and routers path add code as it is

```
import { UserComponent } from './user/user.component';
const routes: Routes = [
  {path:'user',component:UserComponent}
];
```

note : delete this line : `const routes: Routes = [];`

8. Go to app.component.html inside app folder delete all code and write following code

```
<h1 style="text-align: center;">{{title}}</h1>
<app-user></app-user>
```

9. Go to app.component.ts inside app folder and edit title variable which in the AppComponent function write as following code

```
title = 'User Login Dashboard';
```

10. Go to app.mmodule.ts inside app folder and import FormModel and add FormModel in imports array which in @NgModel()

Before import and add code :

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { UserComponent } from './user/user.component';

@NgModule({
  declarations: [
    AppComponent,
    UserComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

After import and add code :

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { UserComponent } from './user/user.component';
// for ng modal
import { FormsModule } from '@angular/forms'; // <-- NgModel lives here
@NgModule({
  declarations: [
    AppComponent,
    UserComponent ],
  imports: [
    BrowserModule,
    AppRoutingModule,
FormsModule ],
providers: [],bootstrap: [AppComponent]})
export class AppModule { }
```

11. Go to user.component.html inside user folder delete all code and write following code

```
<div class="user__container">
    <div class="mb-3 row input__container">
        <div class="mb-3 row">
            <label class="col-sm-2 col-form-label">Username:
            </label>
            <div class="col-sm-10">
                     <input type="text" class="form-control"
                    [(ngModel)]="username"
                    placeholder="username">
            </div>
        </div>

        <div class="mb-3 row">
            <label class="col-sm-2 col-form-label">Password:
            </label>
            <div class="col-sm-10">
                <input type="password" class="form-control"
                [(ngModel)]="password" placeholder="password">
            </div>
        </div>
        <div class="btn_grp">
            <button class="btn-primary" (click)="AddData()">
            Add User </button>
            <button class="btn-danger"
            (click)="userList.pop()">Remove User</button>
        </div>
    </div>

    <!-- </form> -->


    <hr />
```

```
    <h2>Users List</h2>

    <ul class="list-group list-group-horizontal">
        <h3 class="list-group-item flex-fill">Sr. No</h3>
        <h3 class="list-group-item flex-fill">Usernames</h3>
        <h3 class="list-group-item flex-fill">Passwords</h3>
    </ul>

    <!-- <ul class="list-group list-group-horizontal"> -->

        <div *ngFor="let user of userList; index as
        indexOfelement;">


            <ul class="row justify-content-around">
                <span class="col-3">{{indexOfelement}}</span>
                <span class="col-3">{{user.name}}</span>
                <span class="col-3">{{user.pass}}</span>
            </ul>
            </div>

    <!-- </ol> -->

</div>
```

12. Go to user.component.css inside user folder delete all code and write following code

```
.user__container{
    border: 2px solid black;
    margin: 2rem;
    border-radius: 12px;
    background-color:rgb(244, 234, 161);
    font-size: 30px;
    padding: 1rem;
    text-align: center;
}

.input__container button{
    display: block;
    width: 200px;
    margin: 1rem;
    height: 60px;
    border-radius: 10px;
    text-align: center;
}

.input__container div{
    align-items: center;
}

.btn_grp{
    display: flex;
    justify-content: center;
    gap: 20%;
}
```

13. Go to user.component.ts inside user folder import OnInit and implements on UserComponent function then add inside the UserComponent one more function which ngOnItit() , this NgOnInit function is empty and compulsory for OnInit

    Before import, add and implemention code :

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent {

}
```

    After import, add and implemention code :

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

14. Go to user.component.ts inside user folder then add main logic and code inside UserCompent function which gives output

Before add code :

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

After add code :

```
import { Component, OnInit } from '@angular/core';
// import {User} from '../user';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
```

```
export class UserComponent implements OnInit {
  username = '';
  password = '';

  userList = [{
    name:"Shivendra",
    pass:"123"
  },
   {
     name:"Shiv",
    pass:"345"
  }];

  AddData = () => {
    console.log("clicked");
    let temp = {name:this.username, pass:this.password}
    this.userList.push(temp);
  }

  constructor() { }

  ngOnInit(): void {
  }

}
```

15. Last step which run the angular application. So first go to cmd and the go up to folder which store angular application folder after that run following cmd command for run application

    > ng serve --o