Lab 6 – PWA/NodeJs/Angular Part 2
Due: 11:59 CDT, April 23, 2023 (Sunday) (85 points)
Develop a Tic-Tac-Toe game using Angular/Node.js

PWA architecture:
https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures

Description:
The purpose of this lab is to get familiar with PWA and Typescript by developing a simple Tic-Tac-Toe game as a mobile app/PWA using Angular with Visual Studio Code.

Part 2 tasks:
Task 1: Turn in the following files.

1. a screenshot described in step 12 below in a Word doc. Do not zip this Word doc.
2. board.component.ts
3. board.component.html
4. square.component.ts

   You can zip the above 3 files.

Notes:
Follow the instructions in this video: https://www.youtube.com/watch?v=G0bBLvWXBvc
Use the same steps in Lab 6 part1.

Step 1:
Create a web app in Angular using VS Code

a. Create another folder called 'Angular2' or a name you prefer.
b. Inside VS Code, 'open folder' - the 'Angular2' folder in a.
c. Click Terminal and 'new Terminal'.
d. In the Terminal, Type >ng new myapp
e. The above will create an Angular app called myapp.
f. Type Y for – "Would you like to add Angular routing"
g. Type y - yes for 'Add Angular routing' option
h. Important: Choose scss as the style sheet – move your 'ARROW' key to point to SCSS option and hit enter.
i. It will run for a few minutes to compose all the code you need for a web app. Now you have created an angular blank web app with their default components.

j. >cd myapp

k. Type >ng serve -o (to compile your app)

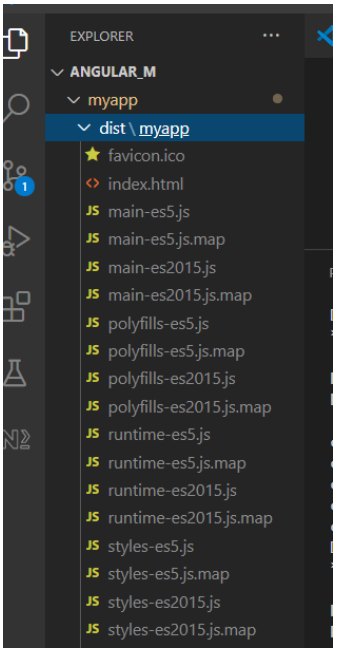
l. You will get an error message if you do not change directory ‘myapp’

m. Type ‘y’ or ‘n’ for Google data sharing question – your choice
n. Compiled successfully
o. The browser will be open with: http://localhost:4200/

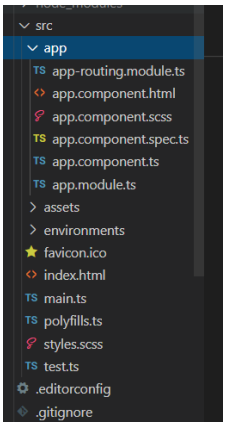
If you get the error below, cd (change directory) to your myapp folder.

“The serve command requires to be run in an Angular project, but a project definition could not be found.”

If you run the build command - > ng build myapp, the files needed for deployment will be stored in the ‘dist’ folder.



We modify code in the source app directory.



Open index.html and take a look.

```
!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Myapp</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

<app-root></app-root> will be replaced by the Angular JavaScript app.

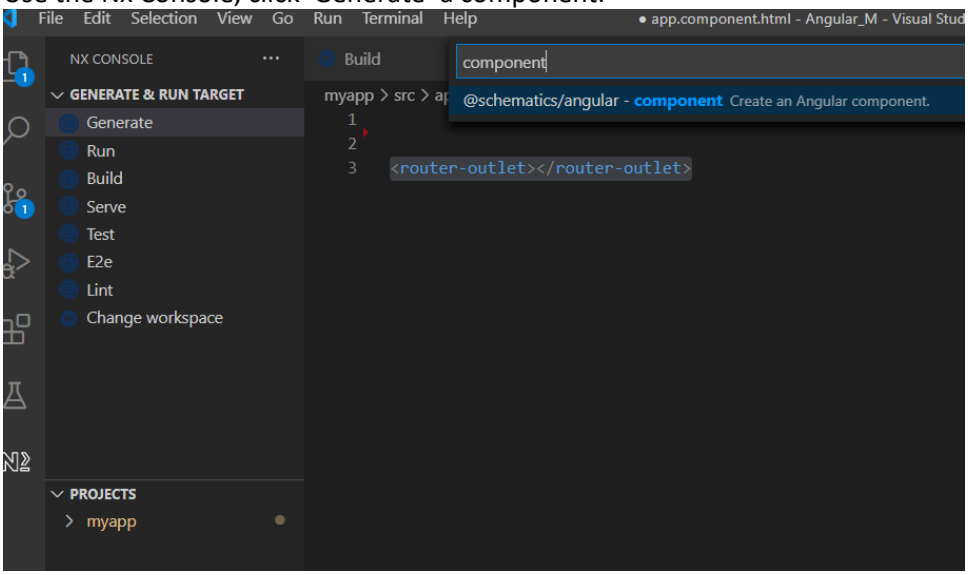
Step 2:

Goto app.component.html (That is the default page for all the web parts there.)
Delete everything except the following:

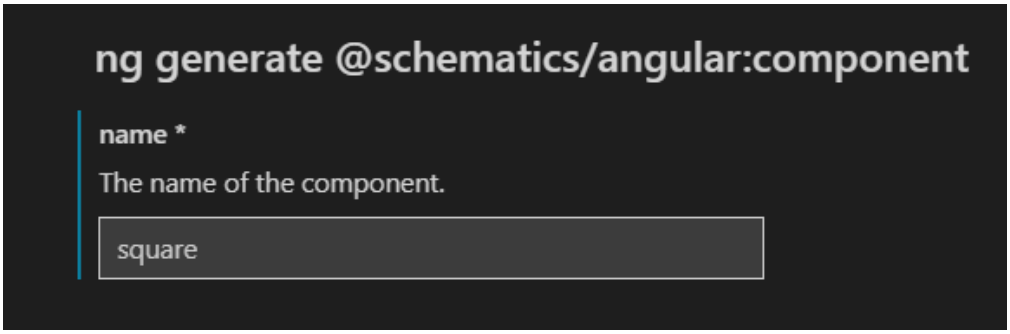
```
<router-outlet></router-outlet>
```

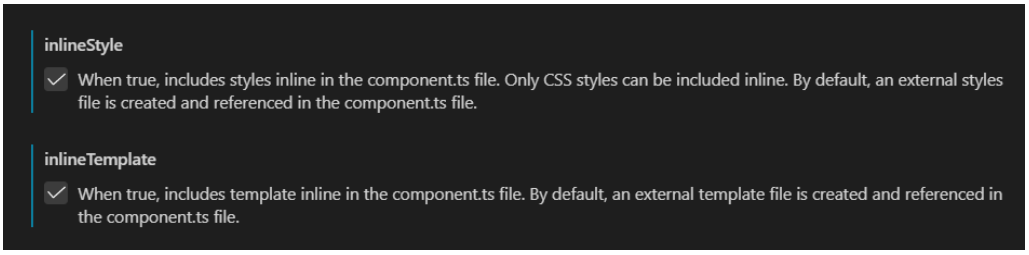
Step 3:

Use the Nx Console, click ‘Generate’ a component.

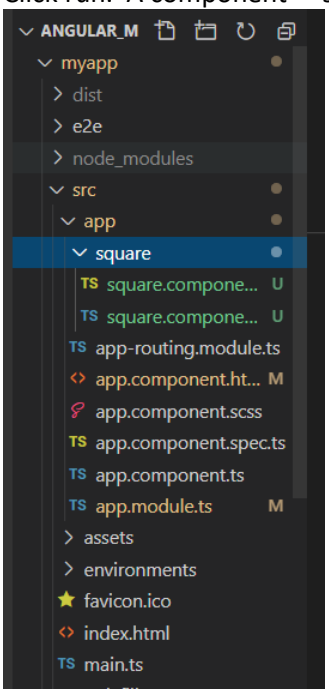


Call it ‘square’ with inline template and style.





Click run. A component – 'square' is created.



Delete this OnInit reference in the following for now.

```
}
export class SquareComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
```

```
  }

}
```

Goto app.component.html:
It has:

```
<router-outlet></router-outlet>
```

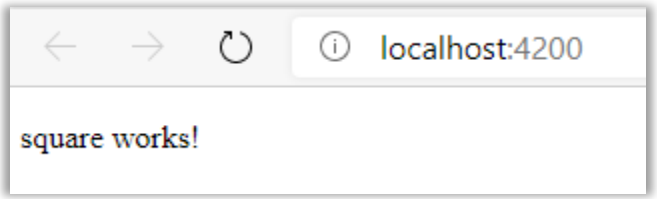
Add:

```
<app-square></app-square>
```

Now the app.component is referencing <app-square></app-square>

If you compile with > ng serve –o

You will see.



Make sure you save all files if you do not see ‘square works!’.

Step 4:

Change ‘square works!’ to {{ rando}}

And add:

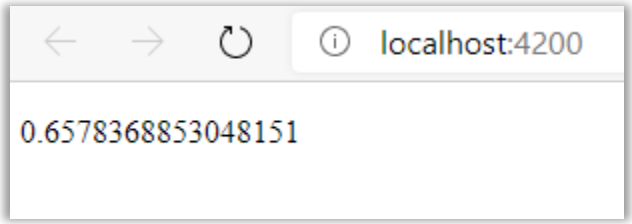
rando = Math.random(); in export class SquareComponent {

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

@Component({
  selector: 'app-square',
  template: `
    <p>
      {{ rando }}
    </p>
  `,
  styles: [
  ]
})
export class SquareComponent  {
  rando = Math.random();
```

```
}
```

Save the file and refresh the web page. It will show a random number:



If you refresh the screen, you will see the number changed.

Step 5:

Now, change the code to this:

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

@Component({
  selector: 'app-square',
  template: `
    <p>
      {{ rando }}
    </p>
  `,
  styles: [
  ]
})
export class SquareComponent  {
  rando;
  constructor(){
    setInterval(() => this.rando = Math.random(), 500);
  }
}
```

Then, the random number will change in an interval of 500 ms.

Step 6:

Add @Input() declarator here:

```
export class SquareComponent  {
 @Input() value: 'X' | 'O';
}
```

```
```

And add

```
<button> {{ value }} </button>
```

To

```
@Component({
  selector: 'app-square',
  template: `
    <p>
      <button> {{ value }} </button>
    </p>
  `,
  styles: [
  ]
})
```

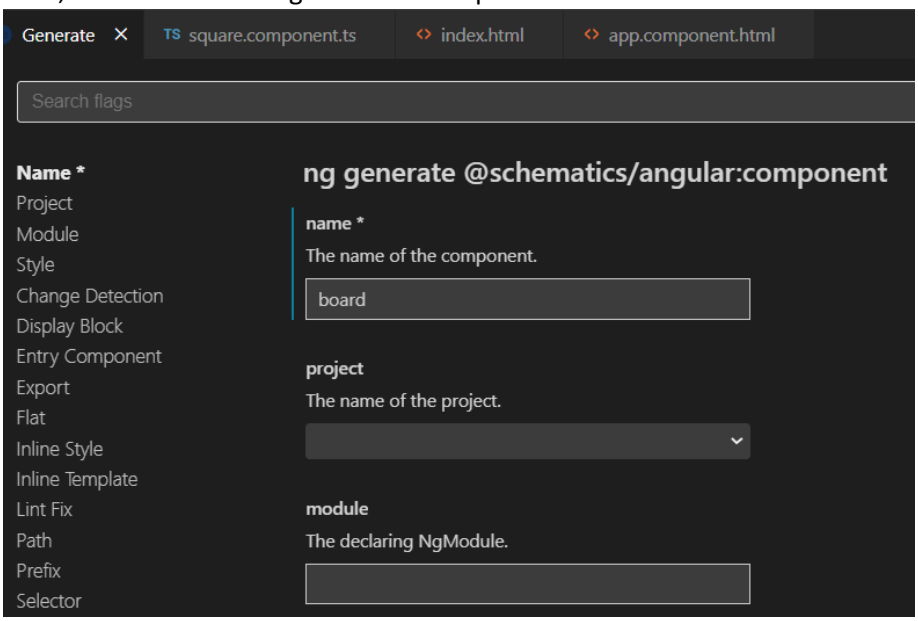
Now, goto app.component.html:

Add: `[value] = "'X'"></app-square>`

```
<app-square [value] = "'X'"></app-square>
<router-outlet></router-outlet>
```

Step 7:

Now, use Nx console to generate a component and name it 'board' with all the default option.



We will make it a smart component- it has internal states that can change.

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

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

  //represent 9 moves:
  squares: any[];
  //determine the current player
  xIsNext: boolean;
  winner : string;  //X or O
  constructor() { }

  ngOnInit(): void {
    this.newGame();  //start a new game
  }
  newGame() {
    this.squares = Array(9).fill(null);
  }
}
```

Step 8:

Add the following logic to board.component.ts:

```
export class BoardComponent implements OnInit {

  //represent 9 moves:
  squares: any[];
  //determine the current player
  xIsNext: boolean;
  winner : string;  //X or O
  constructor() { }

  ngOnInit(): void {
    this.newGame();  //start a new game
  }
  newGame() {
    this.squares = Array(9).fill(null);
```

```
  this.winner = null;
  this.xIsNext = true;
}
get player() {
  return this.xIsNext ? 'X' : 'O ';
}
makeMove(idx: number){
  if (!this.squares [idx]){
    this.squares.splice(idx, 1, this.player);
    this.xIsNext = !this.xIsNext;
  }
  this.winner = this.calculateWinner();
}


calculateWinner() {
  const lines = [
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6]
  ];
  for (let i = 0; i < lines.length; i++) {
    const [a, b, c] = lines[i];
    if (
      this.squares[a] &&
      this.squares[a] === this.squares[b] &&
      this.squares[a] === this.squares[c]
    ) {
      return this.squares[a];
    }
  }
  return null;
}
```

Step 9:

Add this code to board.component.html:

```
<h1>Current Player: {{ player }} </h1>

<button nbButton outline status="danger" (click)="newGame()">Start new Gam
e</button>

<h2 *ngIf="winner">
    Player {{ winner }} won the game!
</h2>

<main>
    <app-square
        *ngFor="let val of squares; let i = index"
        [value]="val"
        (click)="makeMove(i)">

    </app-square>
</main>
```

Step 10: (missing in the video.)

Change code in app.component.html :

```
<app-board></app-board>
<router-outlet></router-outlet>
```

Step 11:

Add nebular for the project:

- ng add @nebular/theme (make sure you are in 'myapp' folder.)
- Choose Comic theme
- And y to customization and animation

Add:

```
NbButtonModule
```

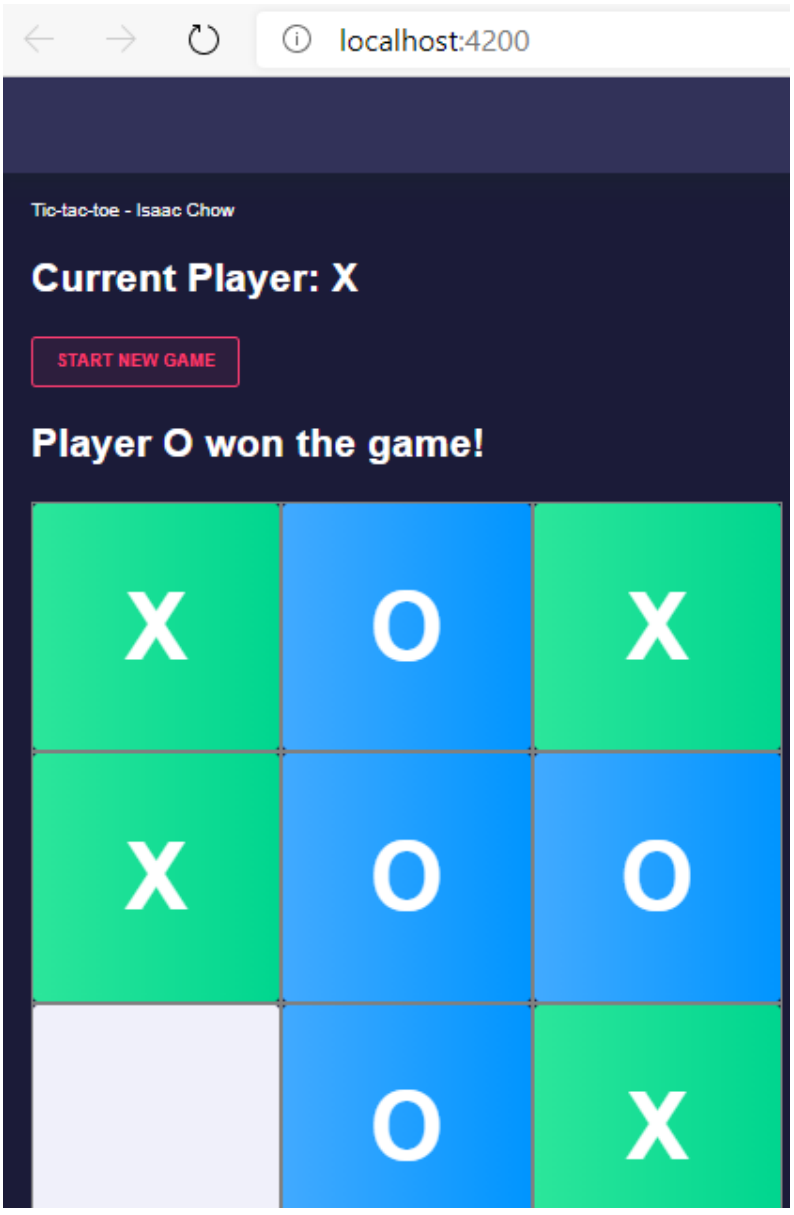
In app.modules.ts

```
@NgModule({
  declarations: [
    AppComponent,
    SquareComponent,
    BoardComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    NbThemeModule.forRoot({ name: 'cosmic' }),
    NbLayoutModule,
    NbEvaIconsModule,
    NbButtonModule
  ],
```

```
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

**Step 12: (Turn in a screen shot of this.)

Put Tic-tac-toe your name at the beginning of the game.
Compile the code again.



Deploying it to Firebase is optional.

Notes:

Potential Issues:

1. You may have an issue about the strict typing of TypeScript with the not-strict typing of the sample code.

Fix: Go into tsconfig.json. For the option "strict," change it to false. This should resolve the issue.

2. You may run into similar errors below when running 'ng serve --o'?

```
C:\Users\kirkt\Angular\my-first-app>ng serve --o
An unhandled exception occurred: Cannot find module '@angular-devkit/build-webpack'
Require stack:
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular-devkit\build-angular\src\dev
-server\index.js
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular-devkit\architect\node\node-m
odules-architect-host.js
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular-devkit\architect\node\index.
js
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular\cli\models\architect-command
.js
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular\cli\commands\serve-impl.js
- C:\Users\kirkt\Angular\my-first-app\node_modules\@angular-devkit\schematics\tools\expo
rt-ref.js
```

Fix: Running 'npm update' beforehand..