Elective module 3

Web Development

PRACTICAL

Name	: -	Prateek Kumar
Registration No	: -	ADIT22AP00152
NSTI Name	: -	NSTI Noida
Course	: -	ADIT (IBM)
Date	: -	25-10-2024
Module	: -	elective module 3
Practical	: -	web development
Requirements/tools	: -	

i) Hardware: -

  i. Working PC
  ii. Internet connection

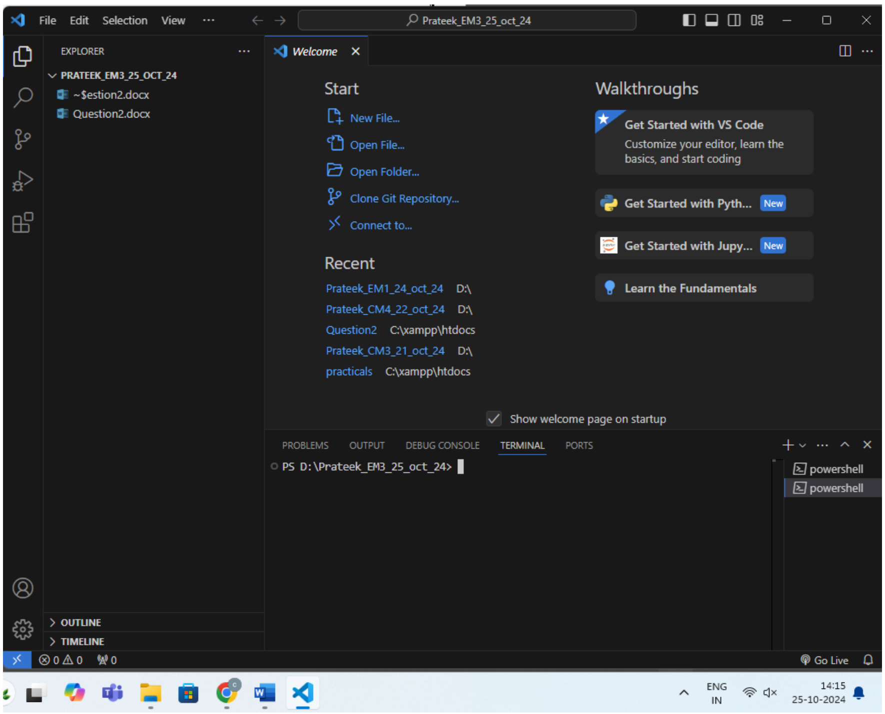
ii) Software: -

  i. Browser
  ii. Code editor(vs code)

Question 1: - Create an angular application that accepts input data using @Input and displays it on page

solution

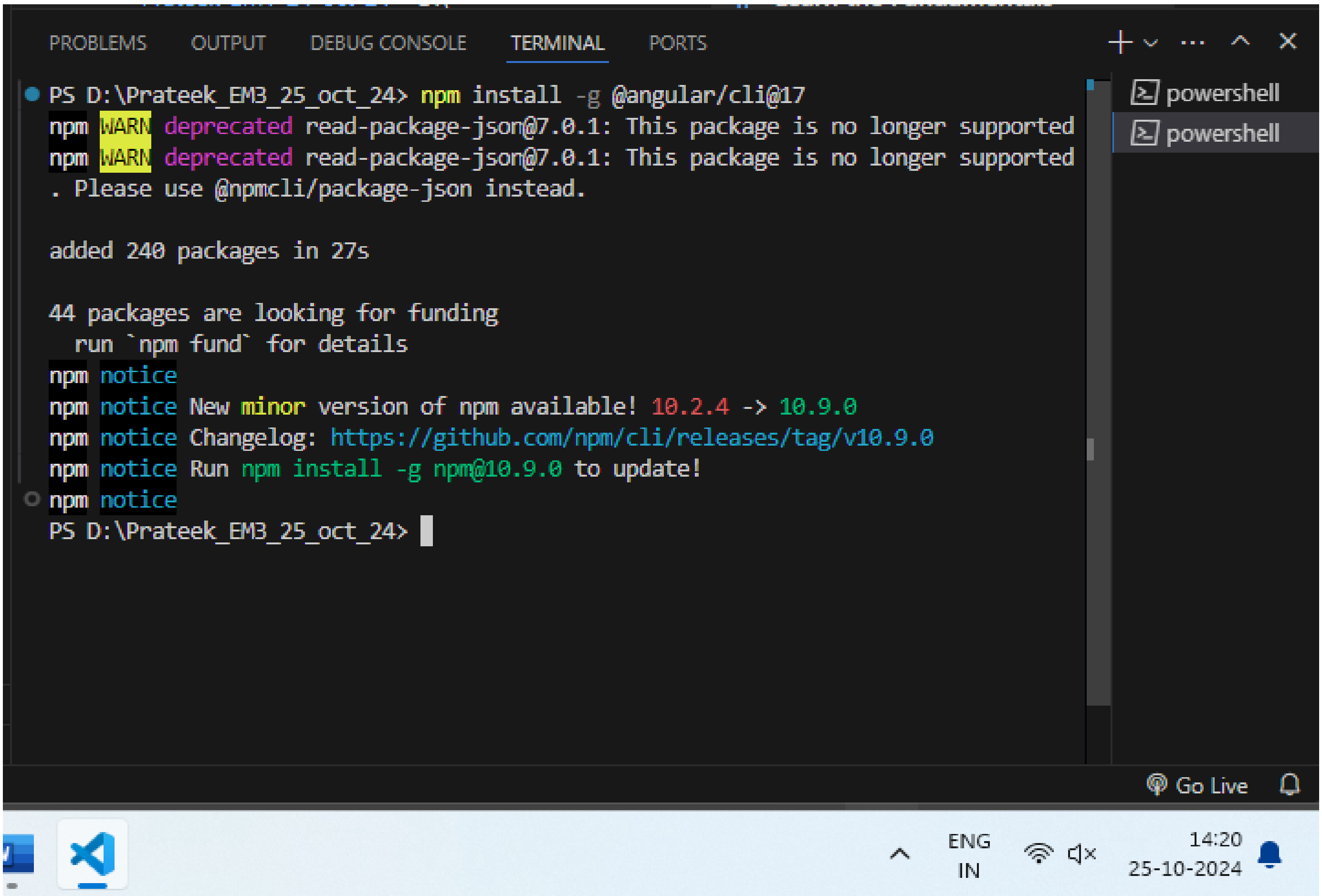
STEP1: - Open visual studio code and open the folder where you want to create the angular project and open the terminal using “ctrl + shift + `”



```
PS D:\Prateek_EM3_25_oct_24>
```

STEP2: - Install latest angular CLI using

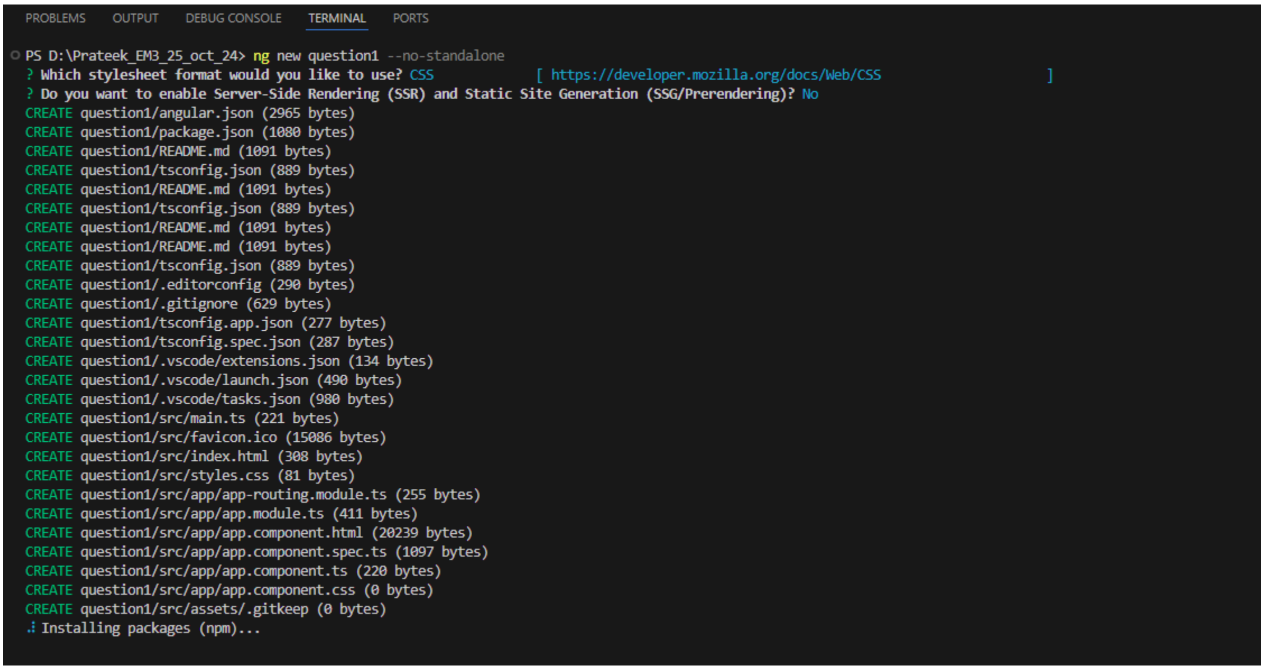
npm install -g @angular/cli@17



```
PS D:\Prateek_EM3_25_oct_24> npm install -g @angular/cli@17
npm WARN deprecated read-package-json@7.0.1: This package is no longer supported
npm WARN deprecated read-package-json@7.0.1: This package is no longer supported
. Please use @npmcli/package-json instead.

added 240 packages in 27s

44 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New minor version of npm available! 10.2.4 -> 10.9.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.9.0
npm notice Run npm install -g npm@10.9.0 to update!
npm notice
PS D:\Prateek_EM3_25_oct_24>
```

STEP3: - Create angular project using ng new question1 and select options to customize your project according to your needs and let it create the application and installing packages

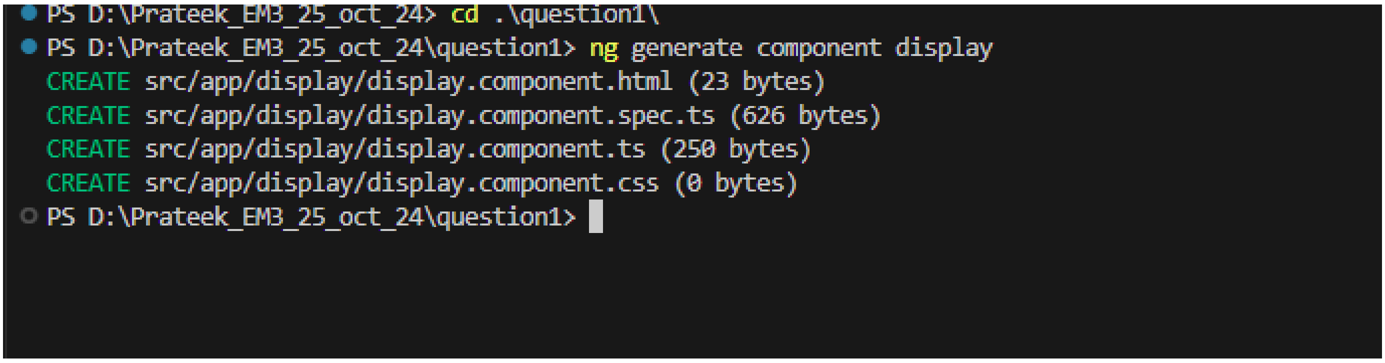
ng new question1 –no-standalone



```
PS D:\Prateek_EM3_25_oct_24> ng new question1 --no-standalone
? Which stylesheet format would you like to use? CSS             [ https://developer.mozilla.org/docs/Web/CSS                    ]
? Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No
CREATE question1/angular.json (2965 bytes)
CREATE question1/package.json (1080 bytes)
CREATE question1/README.md (1091 bytes)
CREATE question1/tsconfig.json (889 bytes)
CREATE question1/README.md (1091 bytes)
CREATE question1/tsconfig.json (889 bytes)
CREATE question1/README.md (1091 bytes)
CREATE question1/README.md (1091 bytes)
CREATE question1/tsconfig.json (889 bytes)
CREATE question1/.editorconfig (290 bytes)
CREATE question1/.gitignore (629 bytes)
CREATE question1/tsconfig.app.json (277 bytes)
CREATE question1/tsconfig.spec.json (287 bytes)
CREATE question1/.vscode/extensions.json (134 bytes)
CREATE question1/.vscode/launch.json (490 bytes)
CREATE question1/.vscode/tasks.json (980 bytes)
CREATE question1/src/main.ts (221 bytes)
CREATE question1/src/favicon.ico (15086 bytes)
CREATE question1/src/index.html (308 bytes)
CREATE question1/src/styles.css (81 bytes)
CREATE question1/src/app/app-routing.module.ts (255 bytes)
CREATE question1/src/app/app.module.ts (411 bytes)
CREATE question1/src/app/app.component.html (20239 bytes)
CREATE question1/src/app/app.component.spec.ts (1097 bytes)
CREATE question1/src/app/app.component.ts (220 bytes)
CREATE question1/src/app/app.component.css (0 bytes)
CREATE question1/src/assets/.gitkeep (0 bytes)
⠼ Installing packages (npm)...
```

STEP4: - Generate the Child Component using

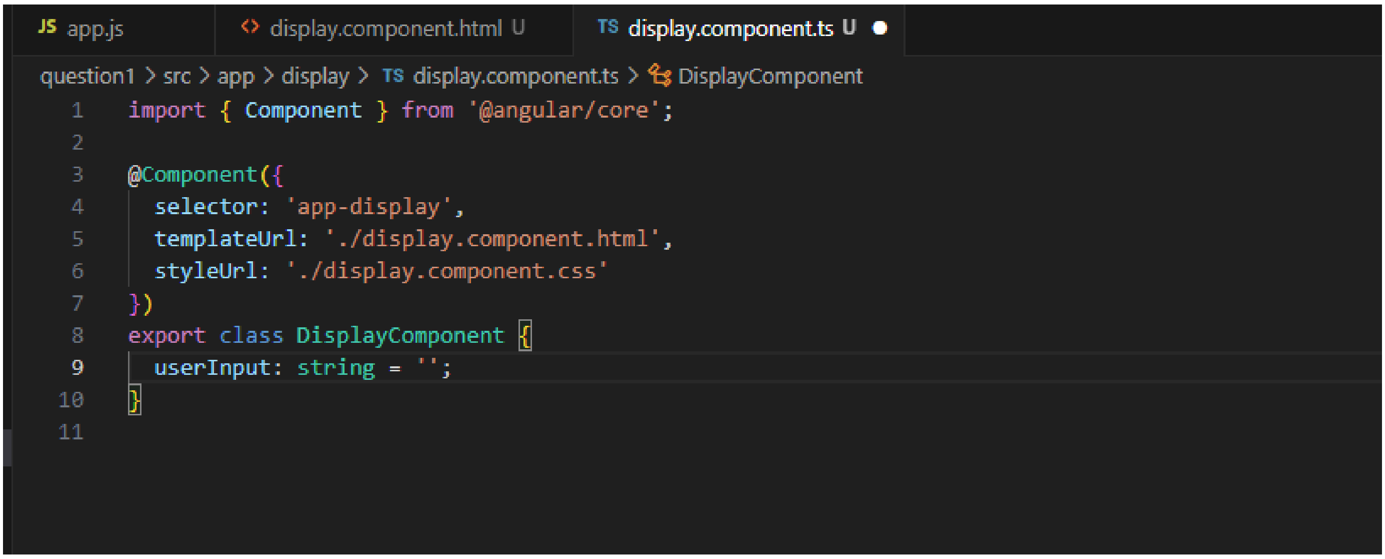
ng generate component display



```
PS D:\Prateek_EM3_25_oct_24> cd .\question1\
PS D:\Prateek_EM3_25_oct_24\question1> ng generate component display
  CREATE src/app/display/display.component.html (23 bytes)
  CREATE src/app/display/display.component.spec.ts (626 bytes)
  CREATE src/app/display/display.component.ts (250 bytes)
  CREATE src/app/display/display.component.css (0 bytes)
PS D:\Prateek_EM3_25_oct_24\question1>
```

STEP5: - configure the Child Component

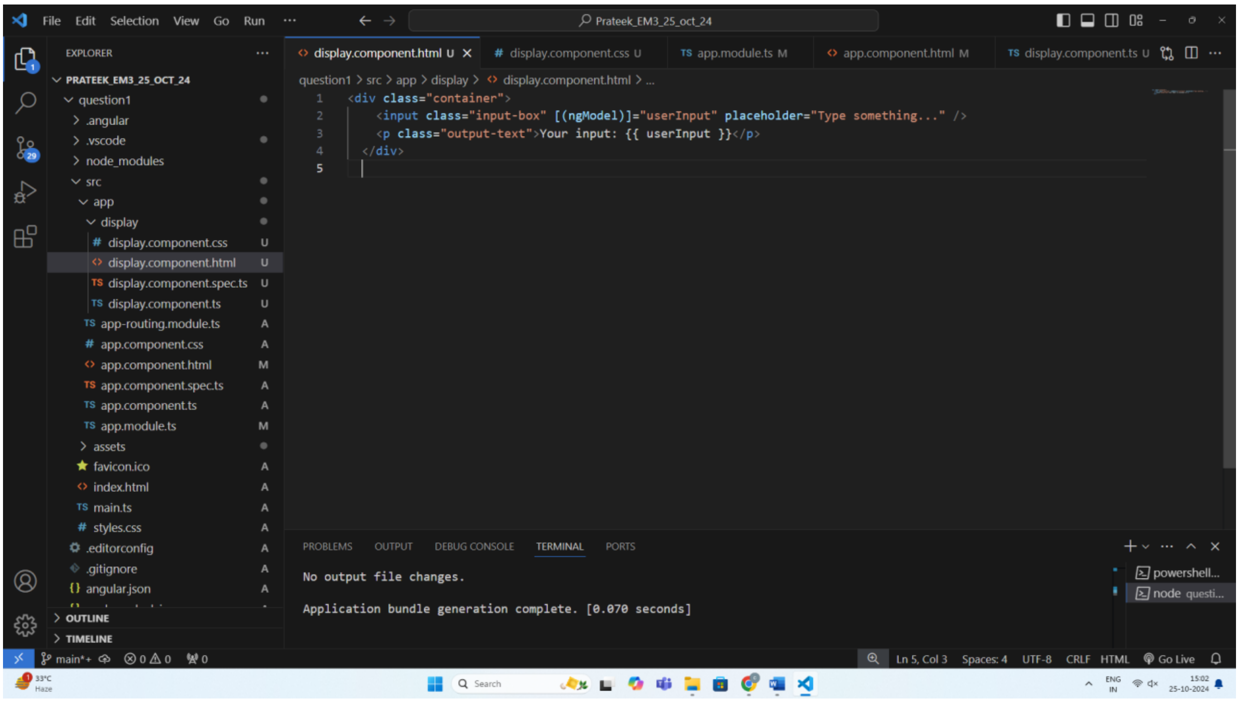
i)create a property to store input



app.js
display.component.html U
display.component.ts U
question1 > src > app > display > display.component.ts > DisplayComponent

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

@Component({
  selector: 'app-display',
  templateUrl: './display.component.html',
  styleUrl: './display.component.css'
})
export class DisplayComponent {
  userInput: string = '';
}
```

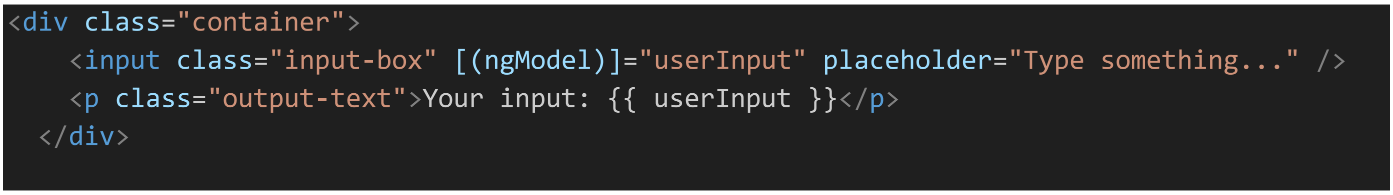
ii)Go to display.component.html and write the code to take user input and show the output on the screen



```
<div class="container">
    <input class="input-box" [(ngModel)]="userInput" placeholder="Type something..." />
    <p class="output-text">Your input: {{ userInput }}</p>
  </div>
```

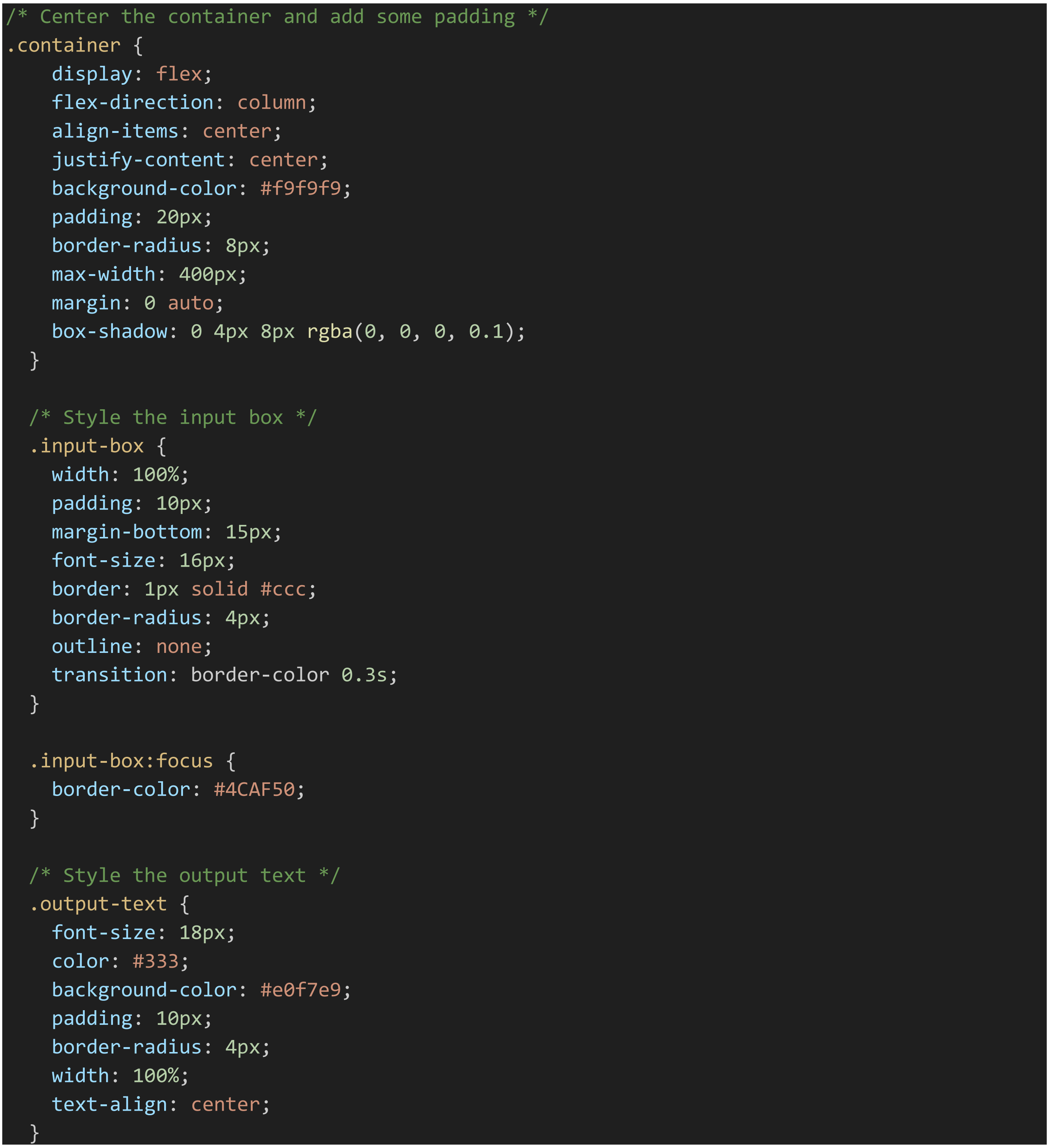
```
No output file changes.

Application bundle generation complete. [0.070 seconds]
```

Here is the code



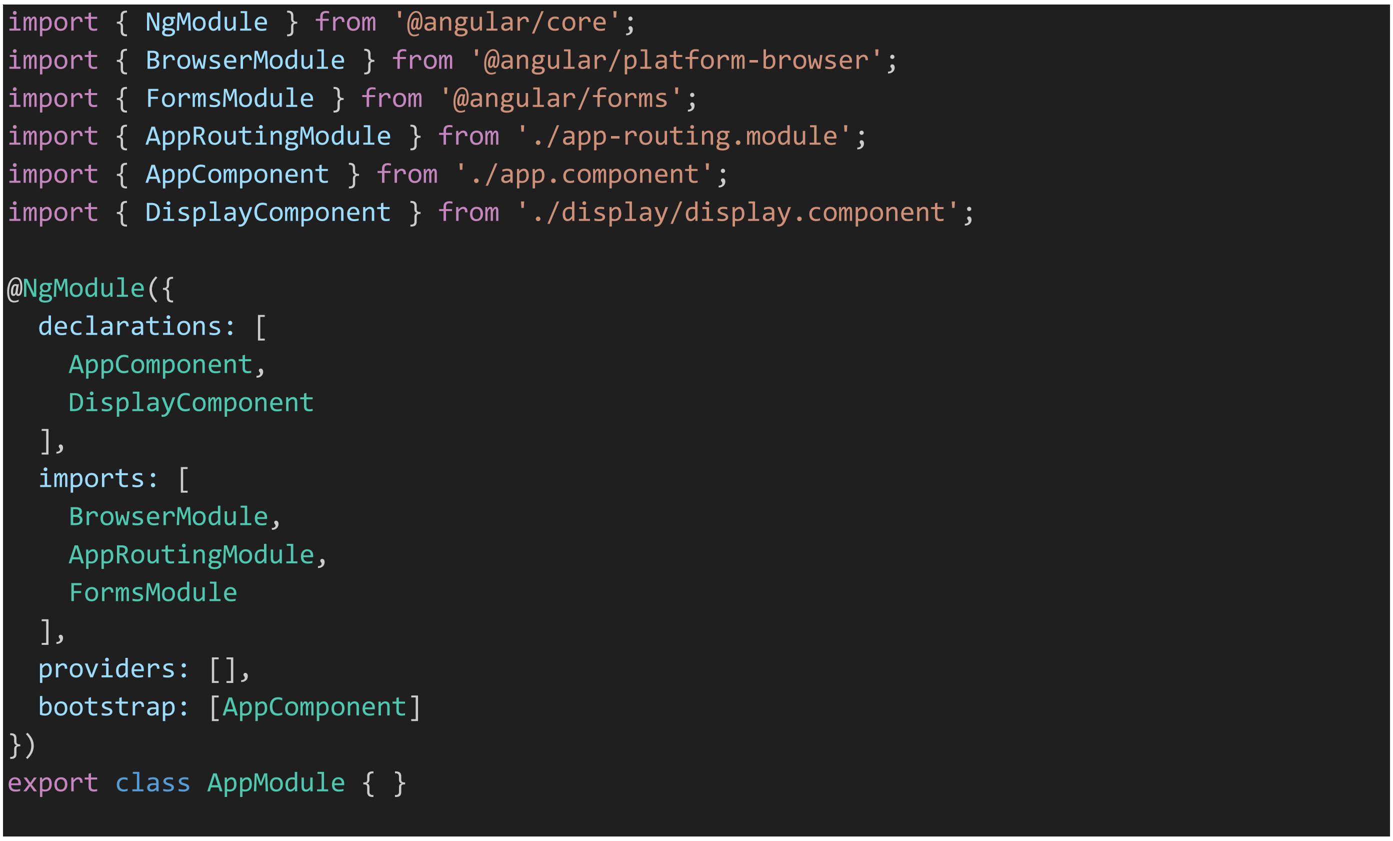
```
<div class="container">
  <input class="input-box" [(ngModel)]="userInput" placeholder="Type something..." />
  <p class="output-text">Your input: {{ userInput }}</p>
 </div>
```

iii).Lets customize it with CSS open display.component.css add css to our html classes



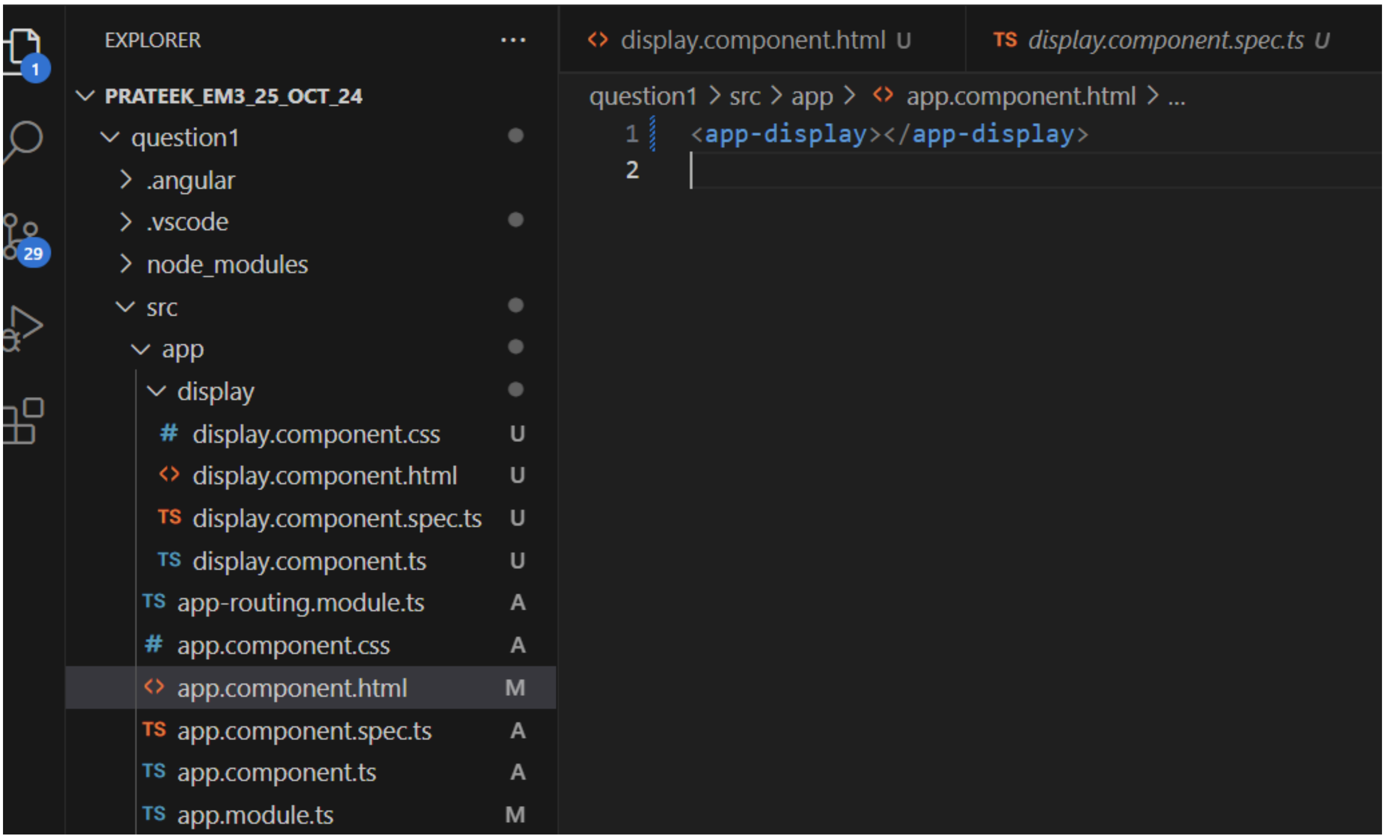
```
/* Center the container and add some padding */
.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  background-color: #f9f9f9;
  padding: 20px;
  border-radius: 8px;
  max-width: 400px;
  margin: 0 auto;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
 }

 /* Style the input box */
 .input-box {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 4px;
  outline: none;
  transition: border-color 0.3s;
 }

 .input-box:focus {
  border-color: #4CAF50;
 }

 /* Style the output text */
 .output-text {
  font-size: 18px;
  color: #333;
  background-color: #e0f7e9;
  padding: 10px;
  border-radius: 4px;
  width: 100%;
  text-align: center;
 }
```

STEP6: - Add FormsModule in app.module.ts file



```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { DisplayComponent } from './display/display.component';

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

STEP7:- Add the display Component to appComponent



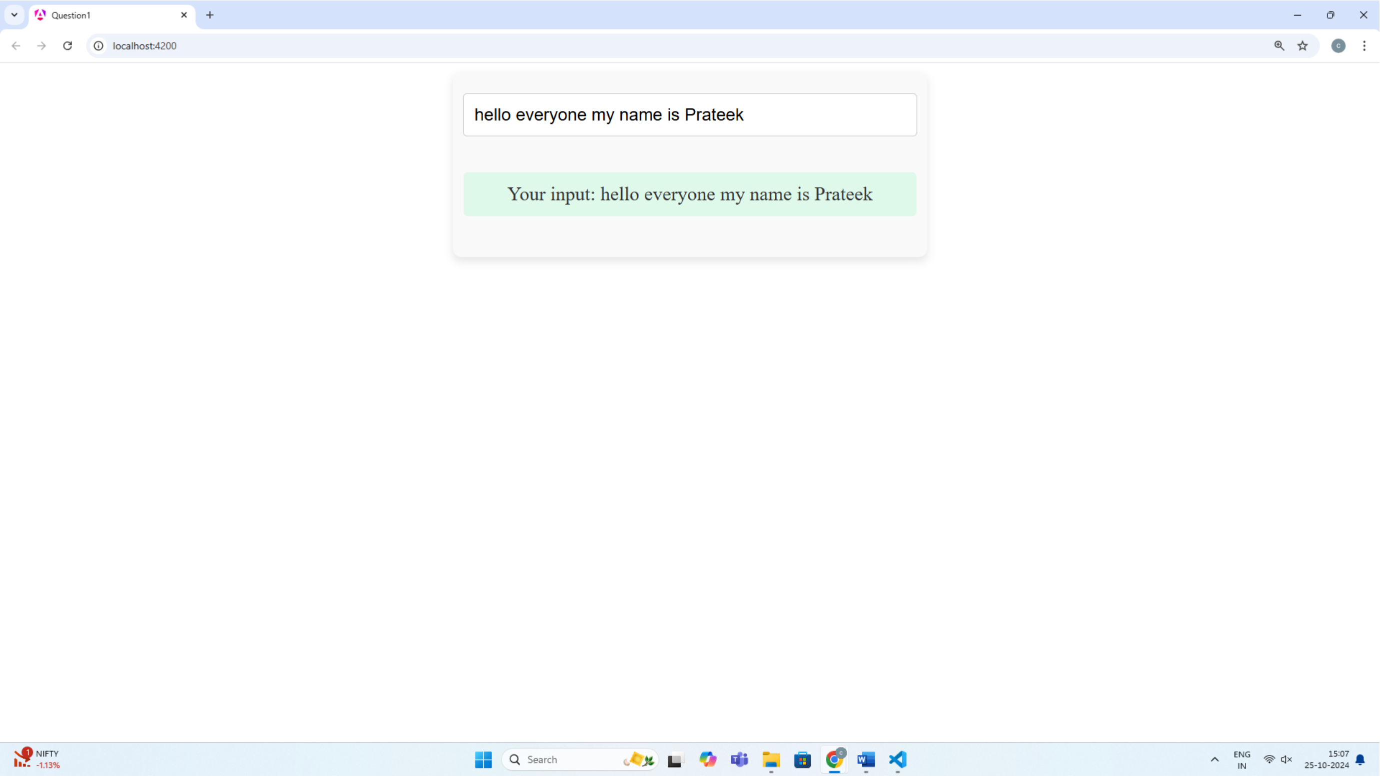
```
<app-display></app-display>
```

STEP8:- run the angular application and test the functionality

i) Run application using

```
ng serve
```

ii) Lets test the output



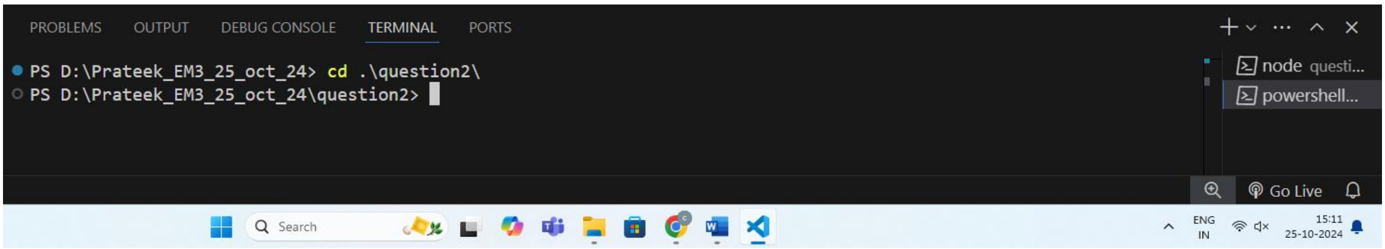
hello everyone my name is Prateek

Your input: hello everyone my name is Prateek

As we can see the two-way data binding is working successfully

Question2:- create a basic express application that listens on port 3000 and responds with “welcome” when accessed.

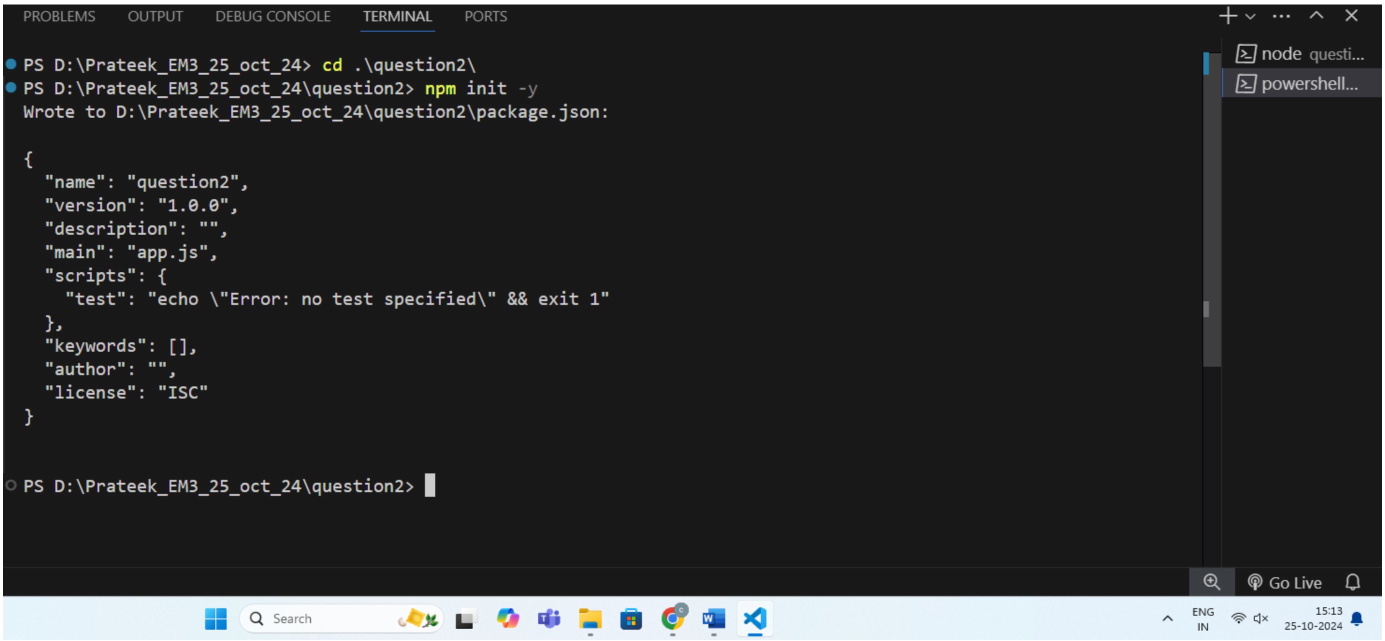
STEP1: - create a folder named question2 in visual studio code and run the terminal using “ctrl + shift + `”



```
● PS D:\Prateek_EM3_25_oct_24> cd .\question2\
○ PS D:\Prateek_EM3_25_oct_24\question2>
```

STEP2: - initialize it as npm directory using

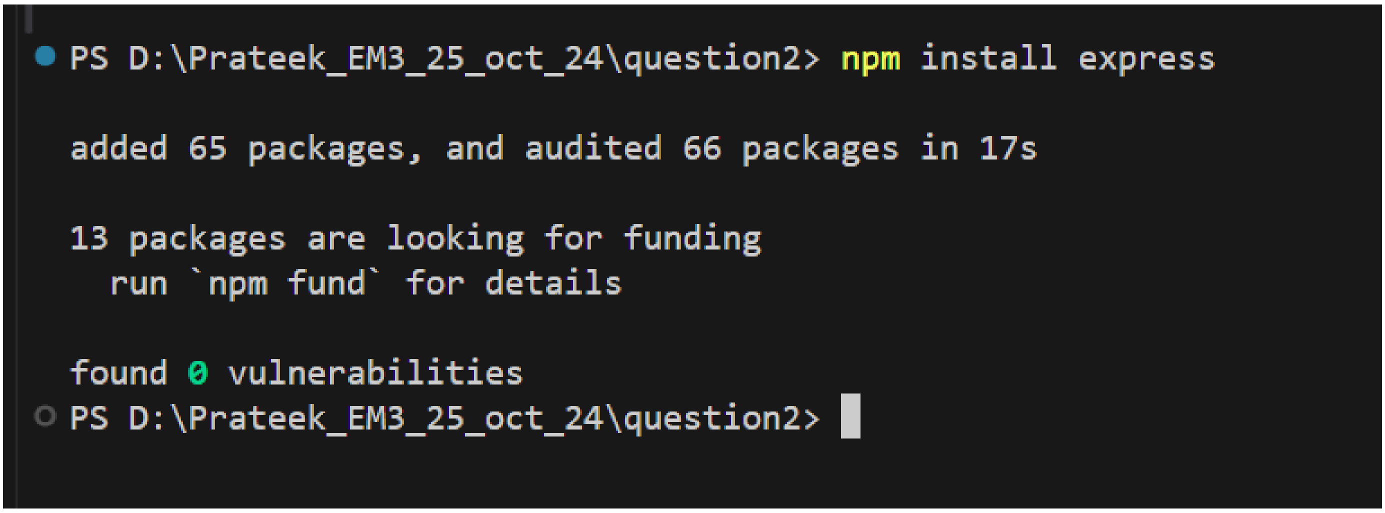
npm init -y



```
● PS D:\Prateek_EM3_25_oct_24> cd .\question2\
● PS D:\Prateek_EM3_25_oct_24\question2> npm init -y
  Wrote to D:\Prateek_EM3_25_oct_24\question2\package.json:

  {
    "name": "question2",
    "version": "1.0.0",
    "description": "",
    "main": "app.js",
    "scripts": {
      "test": "echo \"Error: no test specified\" && exit 1"
    },
    "keywords": [],
    "author": "",
    "license": "ISC"
  }


○ PS D:\Prateek_EM3_25_oct_24\question2>
```

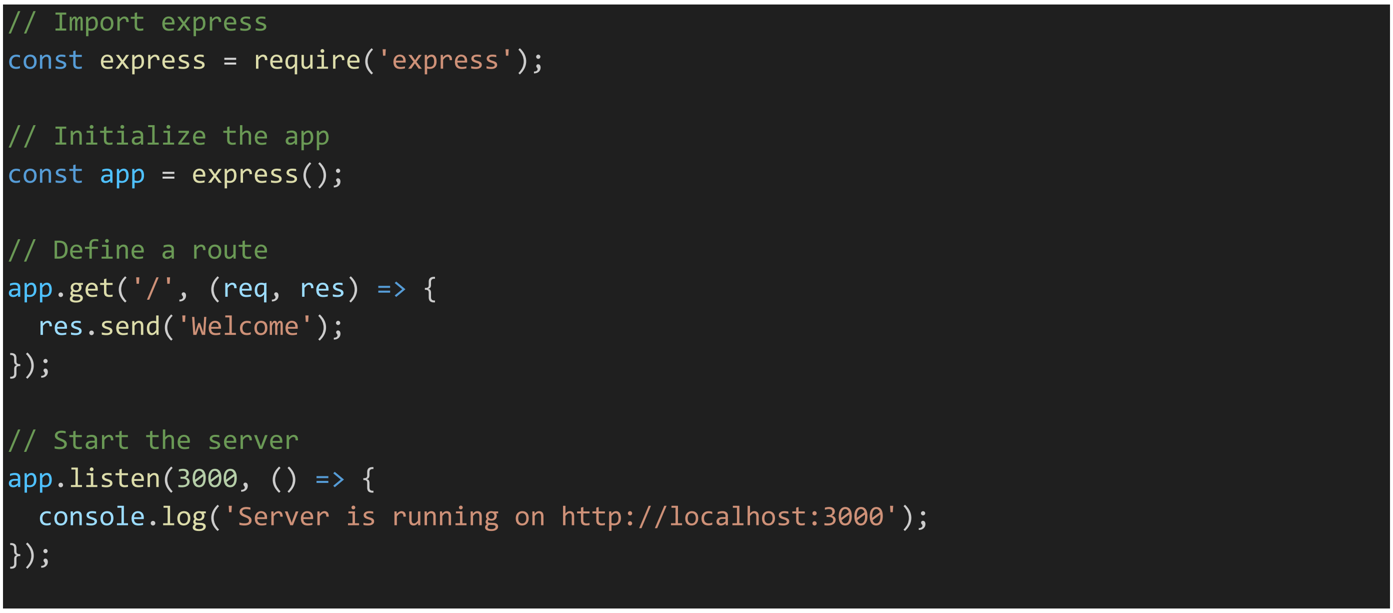
STEP3: - Install express and using

npm install express



```
● PS D:\Prateek_EM3_25_oct_24\question2> npm install express

  added 65 packages, and audited 66 packages in 17s

  13 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
○ PS D:\Prateek_EM3_25_oct_24\question2>
```

STEP4: - create an app.js file and import express and Initialize it and show the output as welcome on the “/” homepage

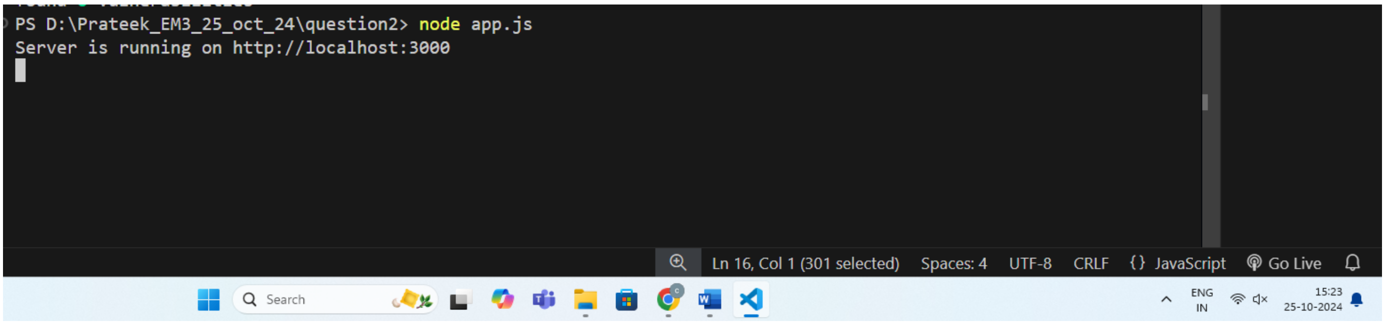


```
// Import express
const express = require('express');

// Initialize the app
const app = express();

// Define a route
app.get('/', (req, res) => {
  res.send('Welcome');
});

// Start the server
app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

STEP5:- run the application and see the output

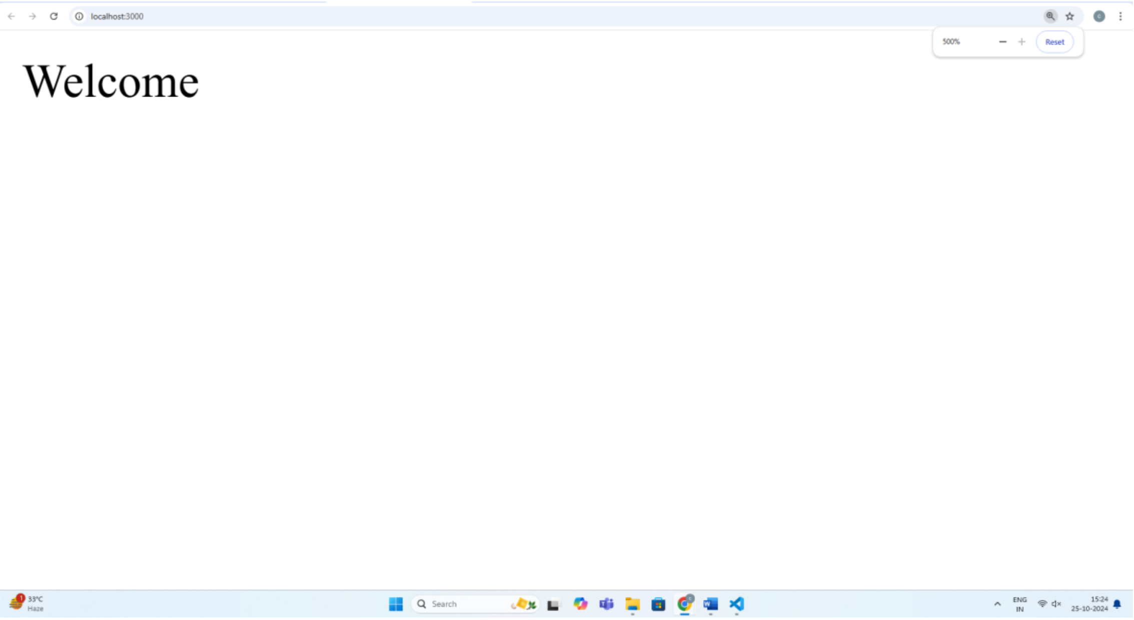
i) run the application using

node app.js



```
PS D:\Prateek_EM3_25_oct_24\question2> node app.js
Server is running on http://localhost:3000
```

ii) check the output



Welcome