Day 1

1. Agile
2. Git

Git setup

* You must install git in your machine
* You must have a git-hub account with your personal mail-id
* Verify that you are able to login to git-hub account from your personal mail-id

What is Git

Distributed Version Control System

Git terminologies

1. Repository - It is kind of folder that keeps track of teams work, it can be created either locally or in the remote location, it will have information about the project version and remote location

Git basic commands

init, add, commit, push

Right solution when you are working in a team

1. you must create a feature branch that is having all the versions of master/main branch
2. work in the local feature branch and push that to the remote repository
3. In remote repository review & integrate the new feature branch with the master branch, in case there’s a conflict don’t integrate inform the user that the feature branch is not up to date with the remote master

What needs to be done if there’s a merge conflict

1. Pull the changes from remote to local master
2. Create a new branch from the local master & then do the changes [or] switch to the existing branch and merge it with the master and then do the changes

Activity:

1. Update a.txt in the user02 feature branch (issue50)
2. Push the feature branch to the remote
3. In remote repository merge the branch if there’s no conflict & delete the feature branch.

Activity

1. Delete the global config that are
   1. user.name
   2. user.email
   3. credentials.helper
2. create a local repository & add that to the remote
3. Try out with 2 users with feature branch and understand what needs to be done if there’s a merge conflict.
4. Note down all the commands in your notes

Summary of GIT commands

* Git configuration - to setup user.name and user.email

git config --global user.name “your-user-name in git”

git config --global user.email “your-email-id”

* Setting up the credentials so that it doesn’t ask you to enter password each time you push the changes to the remote repository

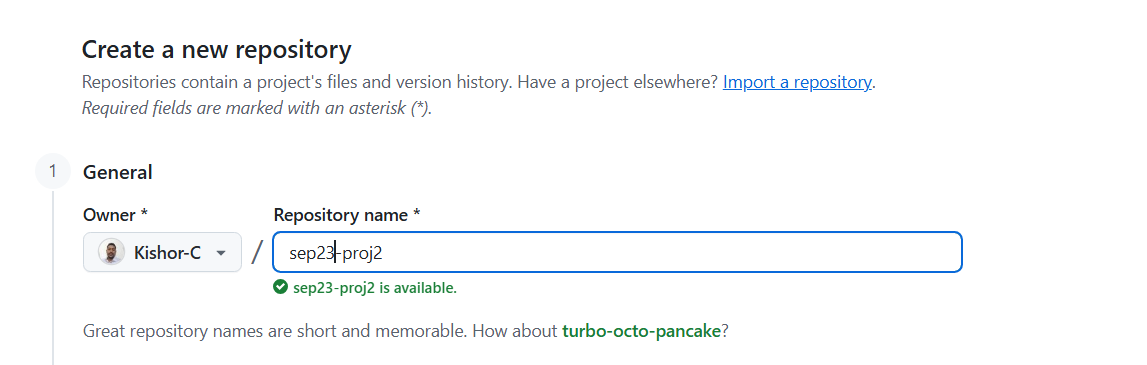
git config --global credential.helper token-id

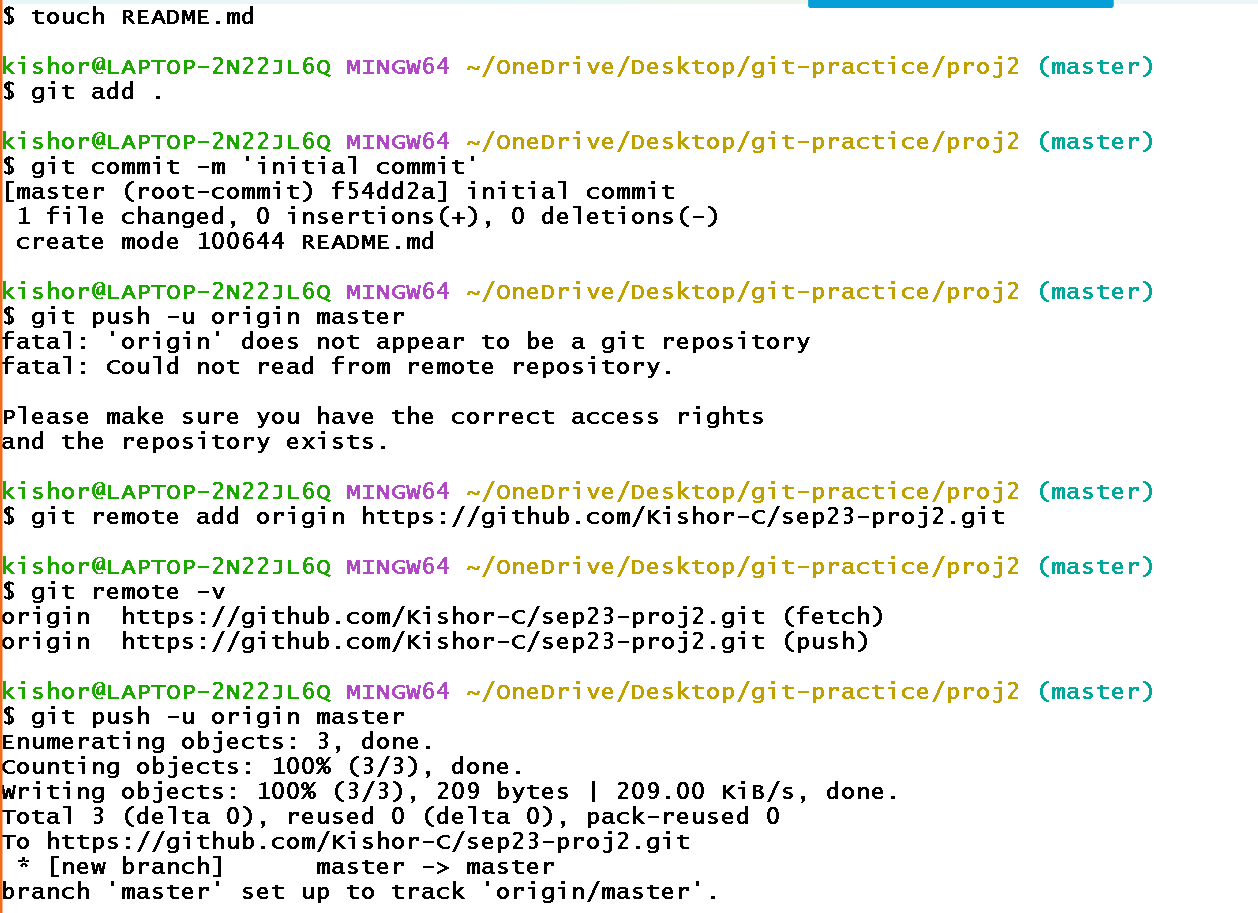
Note: You can go to developer setting to get the token-id (classic)

* init, add, commit, push, status, log, pull, clone, checkout, branch, restore commands

Day 2

Create a new repository in the git-hub with the name sep23-proj2





Commands entered

git init  
touch README.md  
git add .  
git commit -m 'initial commit'  
git remote add origin <https://github.com/Kishor-C/sep23-proj2.git>  
git push -u origin master

How to format the README.md file so that it will be easier for users to read the content

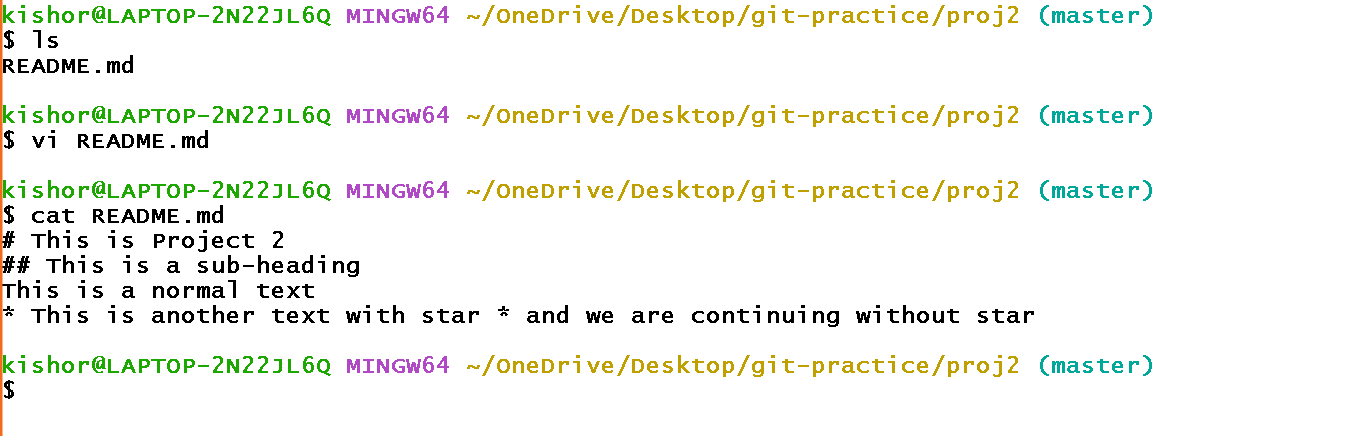
# this is for main heading like heading 1

## this is for heading 2

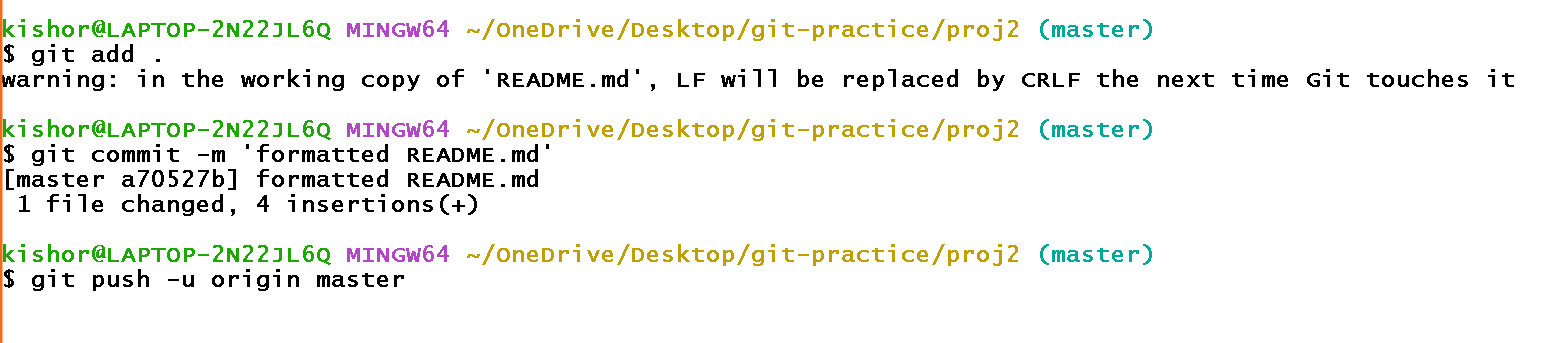
You can use maximum 6 different headings, for sixth heading you must use # 6 times.

‘\*’ is used for bullet point if you give space after the \*.

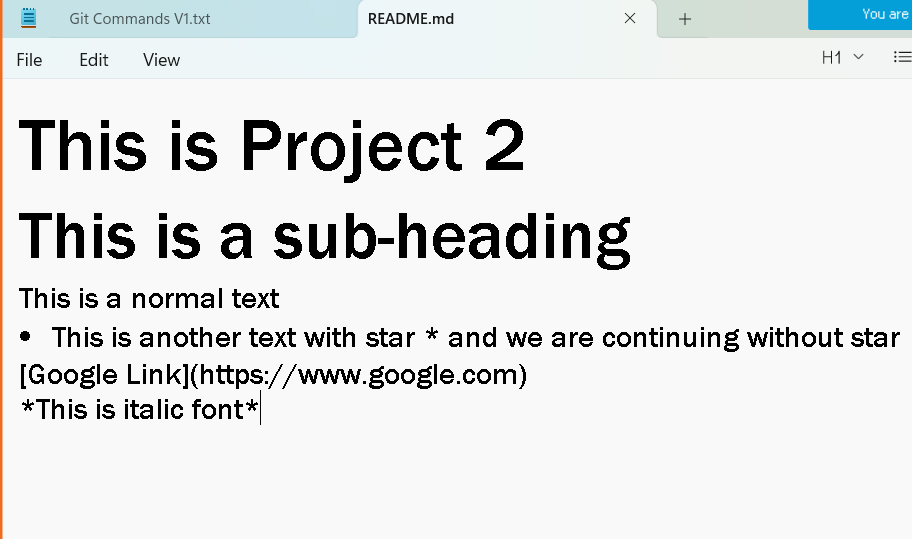
Following are the commands and the content of README.md file



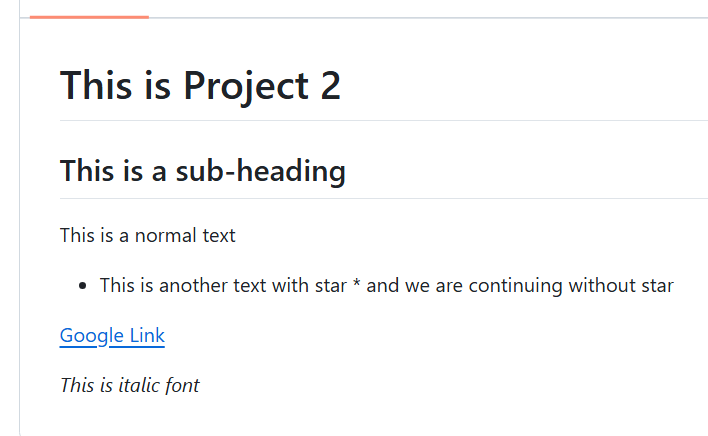
Add & Push the changes to the remote



Formatting the README.md file with some cheatsheet markup



Output:



Simple activity: With the help of cheat-sheet format the README.md file that will have bold-fonts, italic-fonts, lists, links

For new lines you can use two spaces and hit enter [or] you can also use <br> tag which is an HTML tag [or] add 2 extra new lines

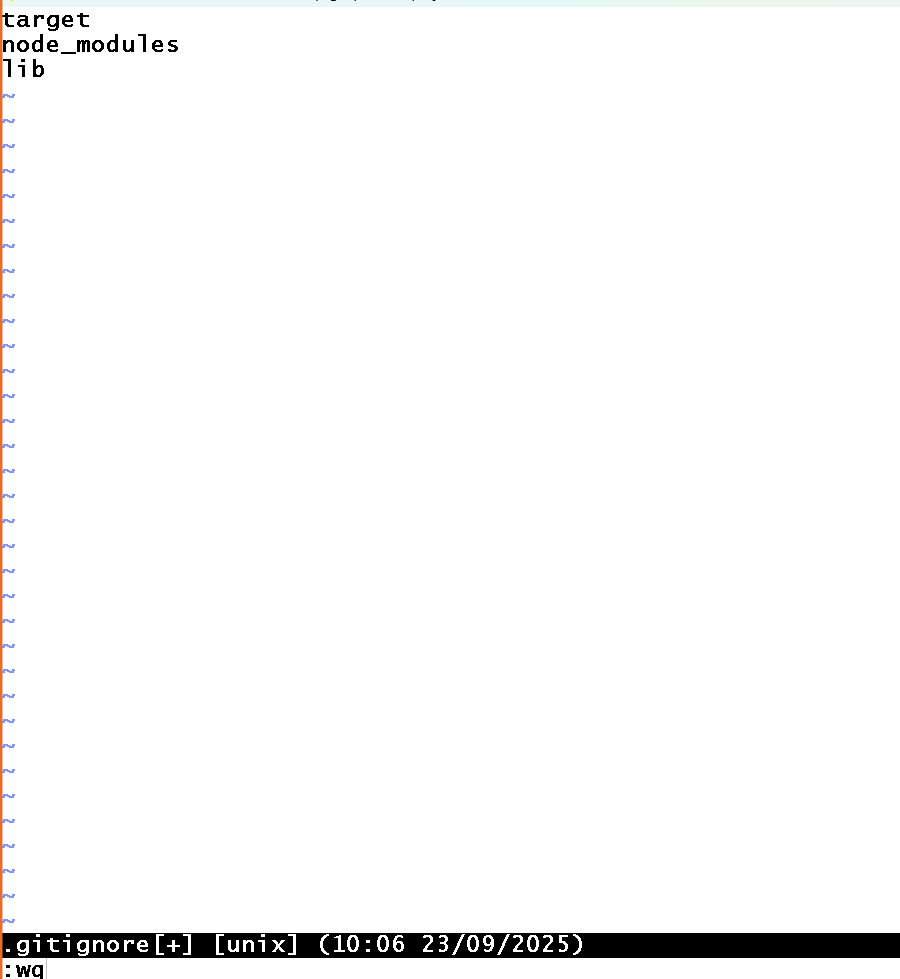
git ignore files

Sometimes you don’t have push project related libraries into git or IDE related files into git ex: node\_modules, target, .settings, .metadata

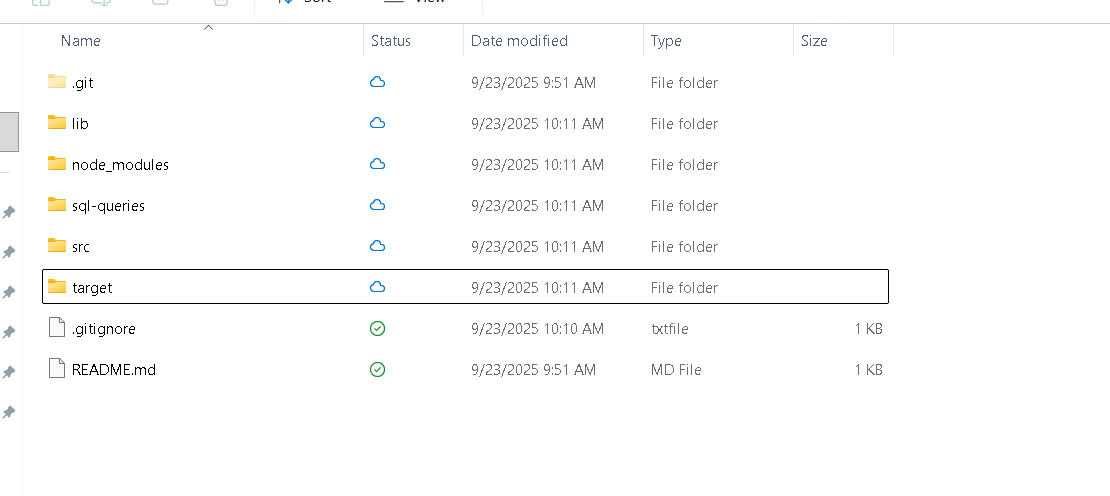
You must create a .gitignore file and mention list of files that shouldn’t be tracked.

touch .gitignore

Edit and add below list of files/folders name in the .gitignore

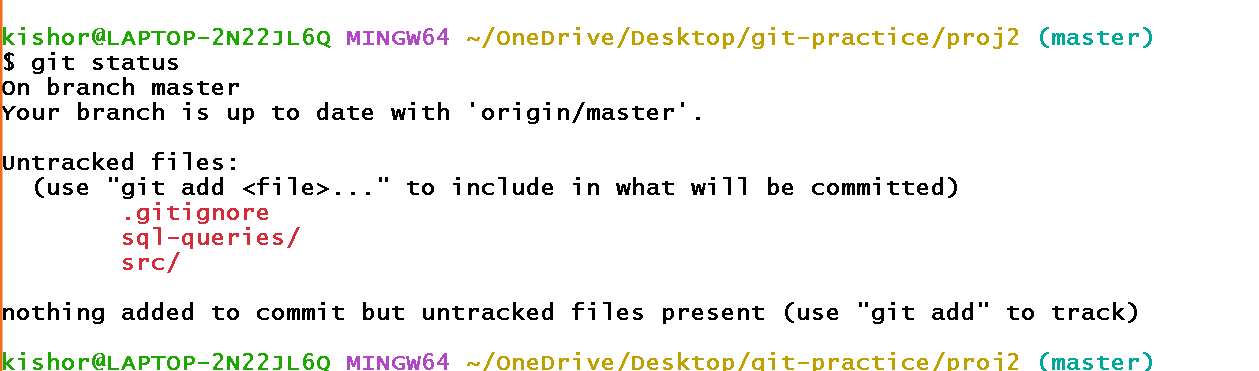


Create src, lib, target, node\_modules, sql-queries folders



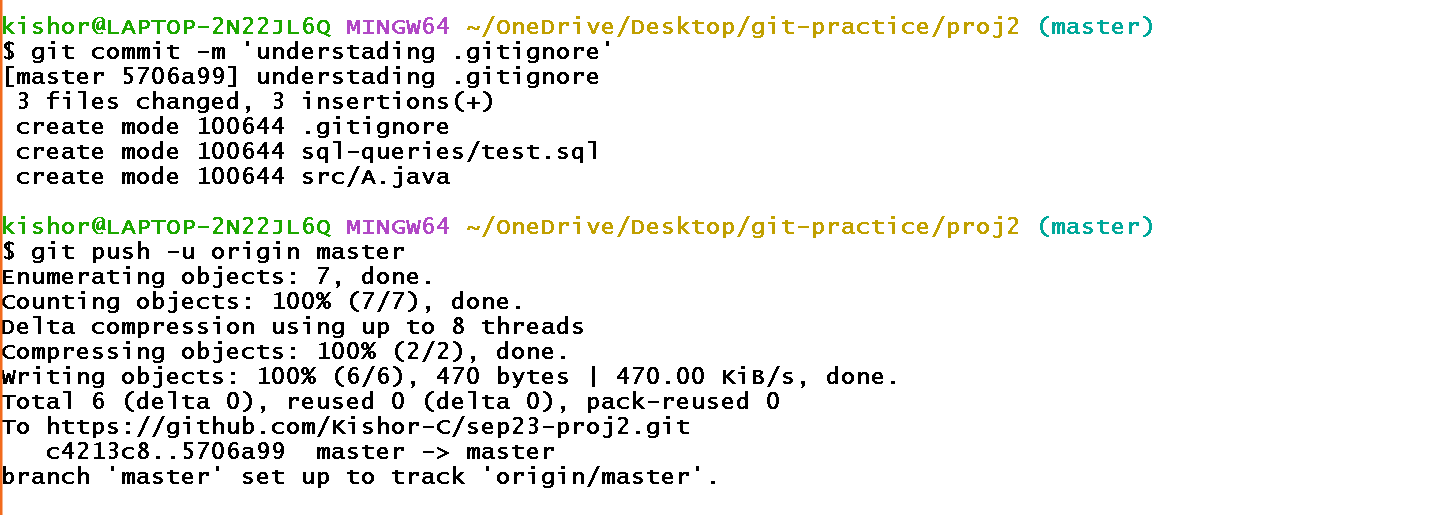


Enter git status to know all the tracked files

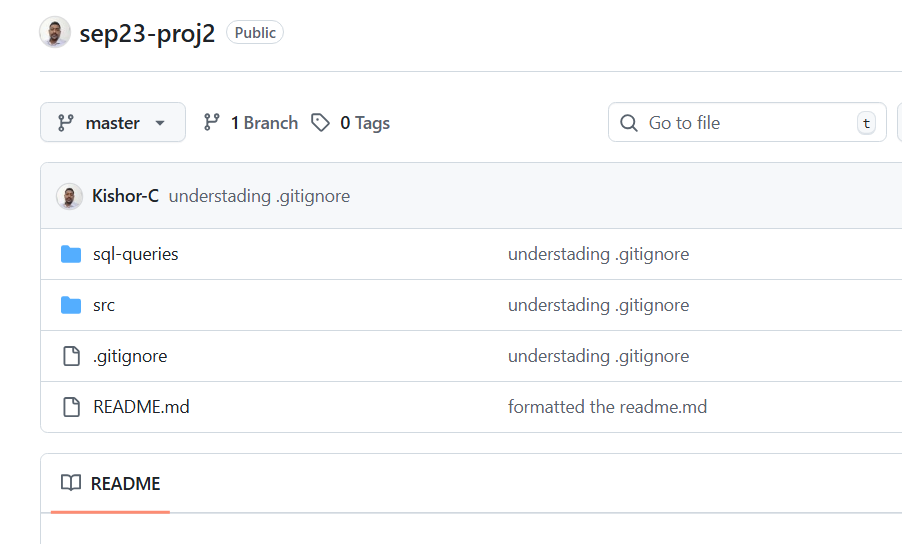


Notic only sql-queries & src are tracked and other folders like lib, target & node\_modules are not tracked, it means they are not pushed to the remote repository.

Try to push this project to the remote



Output:



Git works in many cloud platforms that don’t explicitly use GIT hub repository which are

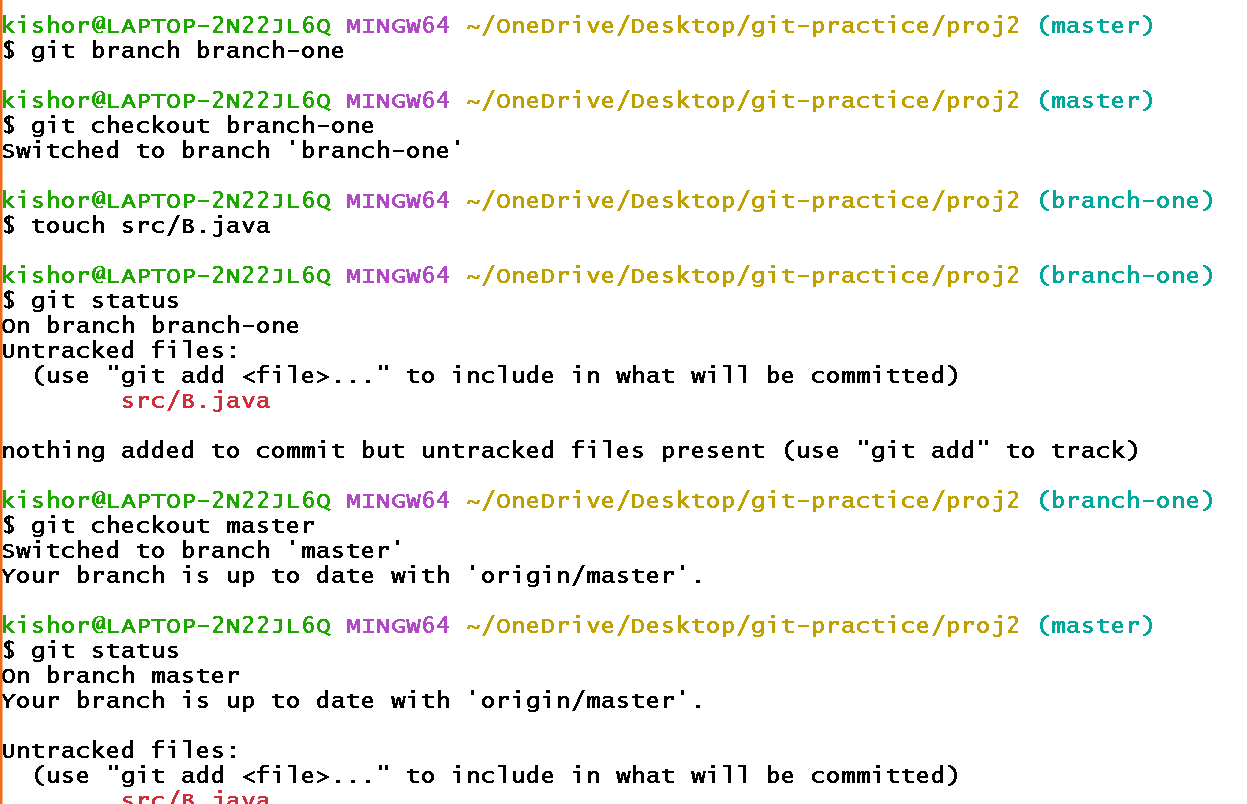
* Bit Bucket
* AWS code commit
* Azure

All these platforms have repositories which are private it means specific to the organization, they are not public for any public contribution. However Git-Hub is public others can also contribute to any project

All these platforms need a sign-in and you can create private repository and through GIT commands you can still update these private repositories

Git stash command

Sometimes you might be in the middle some work and want to switch to a different branch, then you can save using stash so that it won’t appear in other branch.



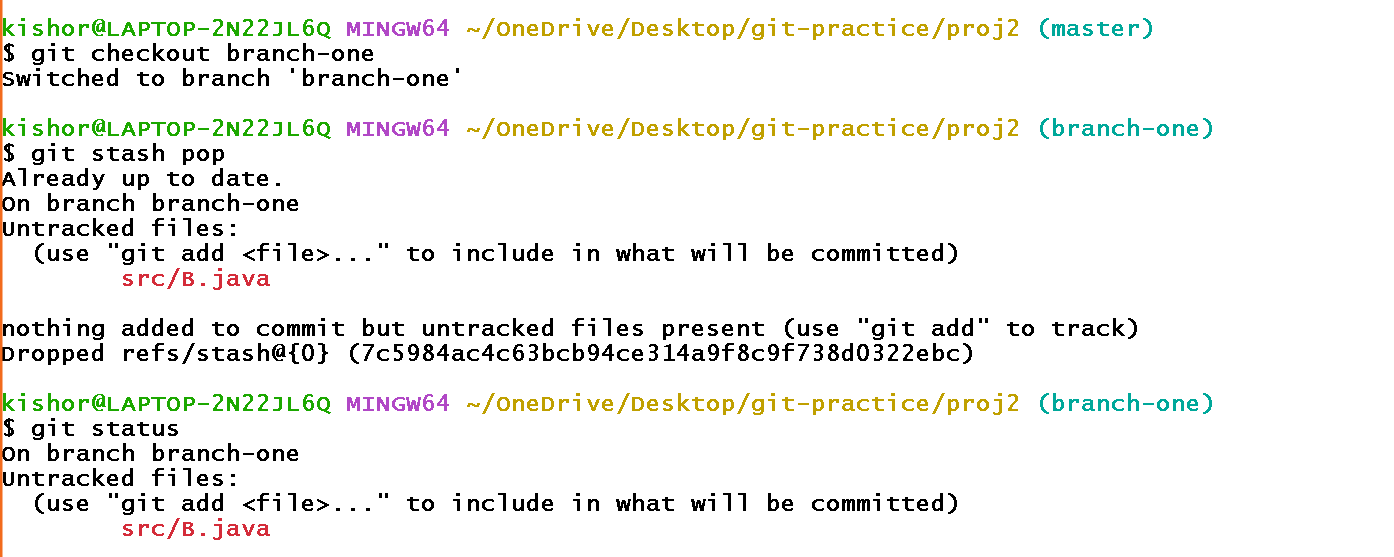
So here branch1 created some file, but when you switch to some other branch that untracked files appear, what we need to do is we must stash it so that when you switch the branch they don’t appear.



You will not see the untracked files in different branch ex: master branch



Now you can get those stashed files if you want to work on it, for that you need to again switch to the branch-one



Assuming that B.java file is completed you can push that to the remote

Listing the branch

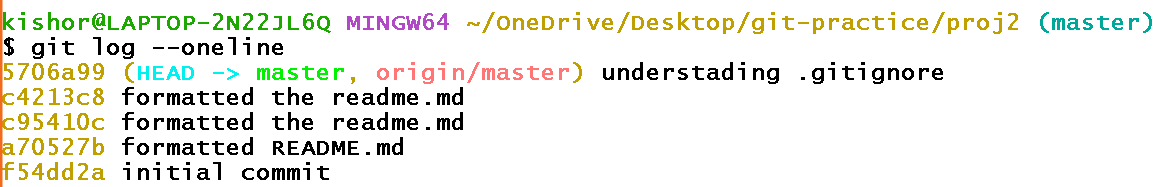
git branch



delete the branch

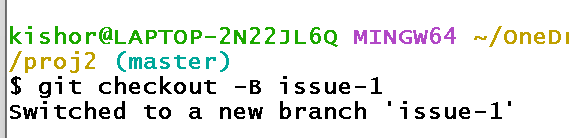


git log --oneline

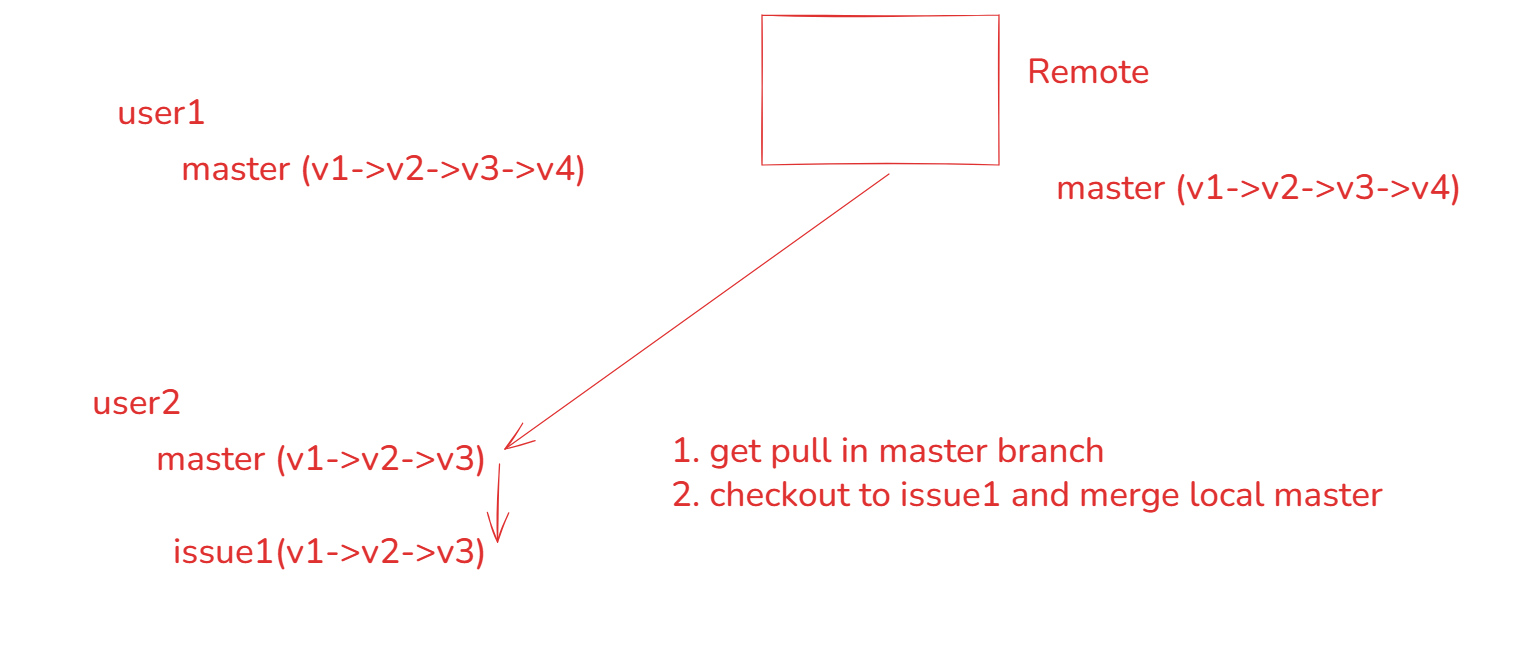


Sometimes you may have many branch in your local repository and you need to know your branch is having which version then these git log will be helpful, using which you can merge in case you want the latest version.

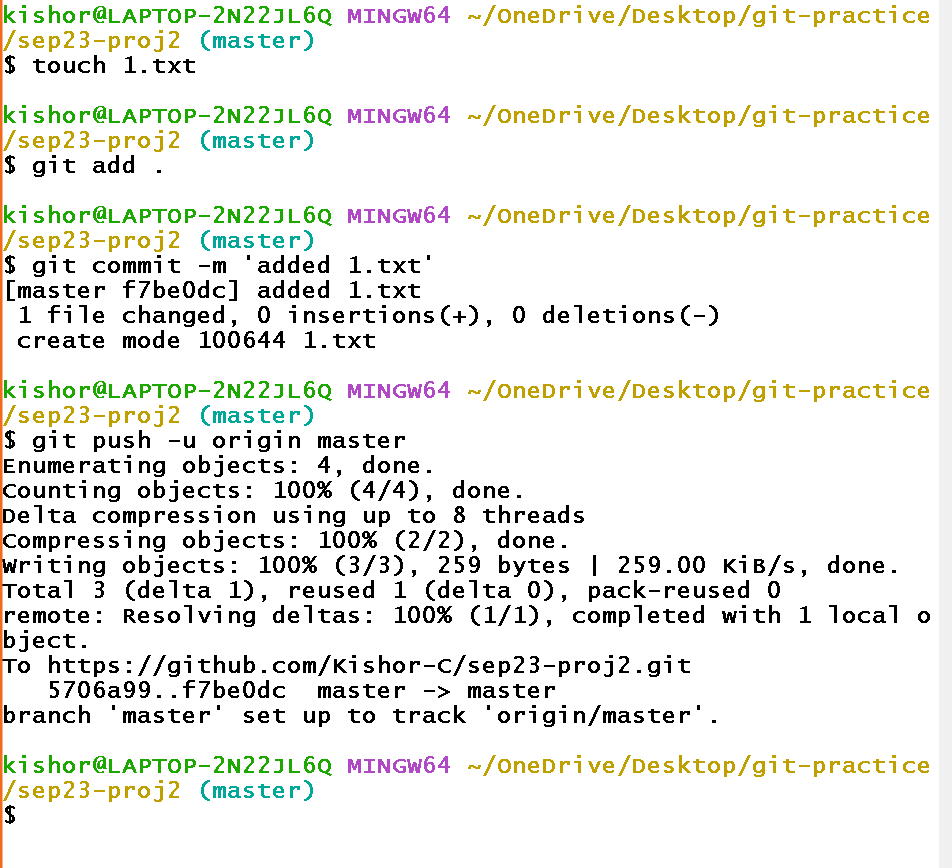
Merging a branch in another branch locally



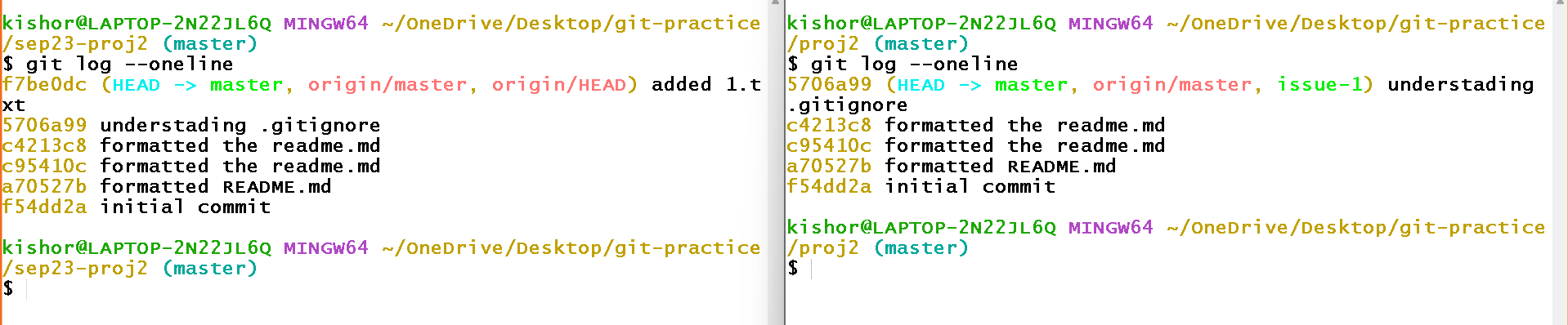
Above command creates a branch if not present



User1 does some changes to the remote master

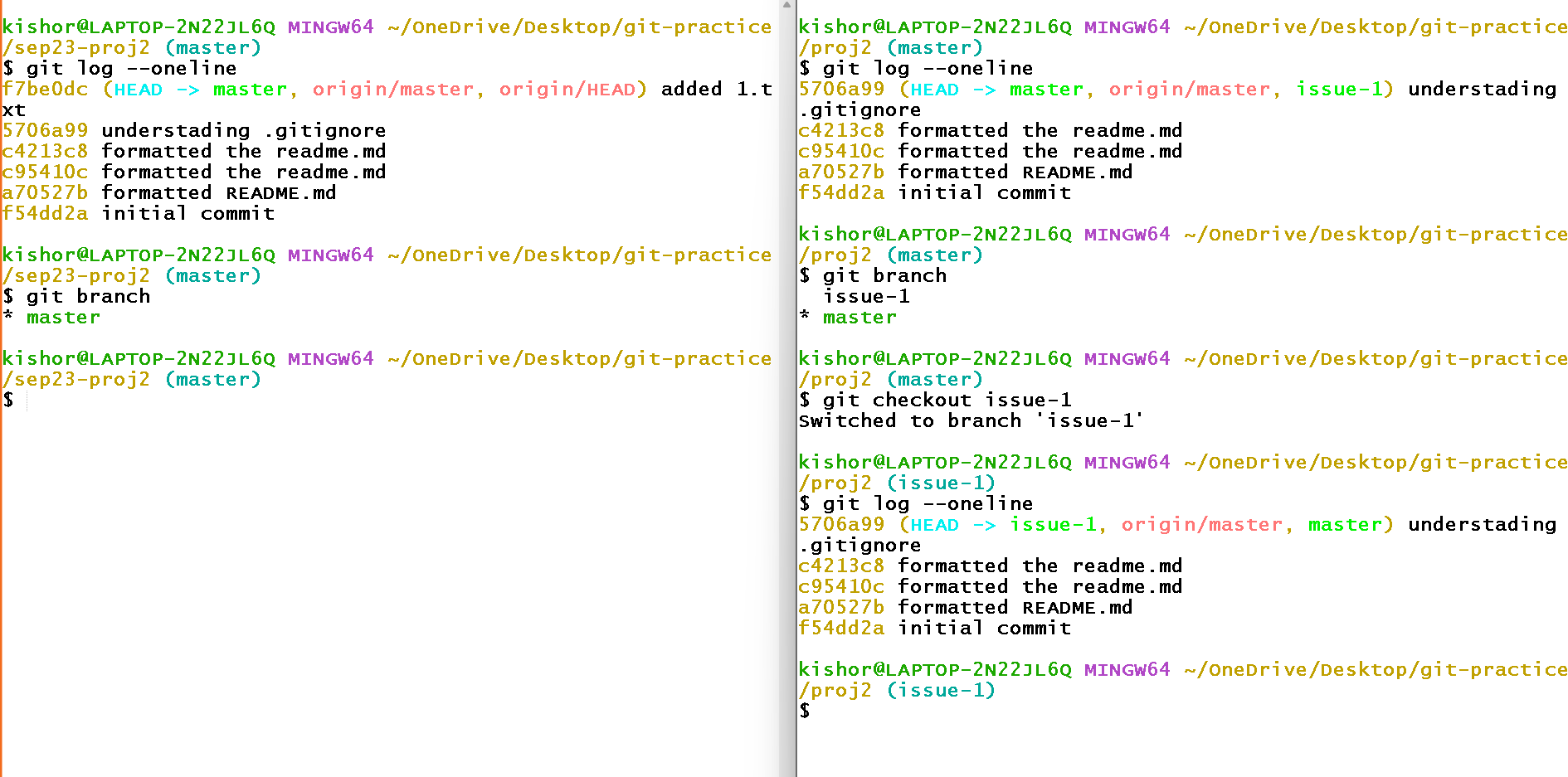


Log the history in both the terminals

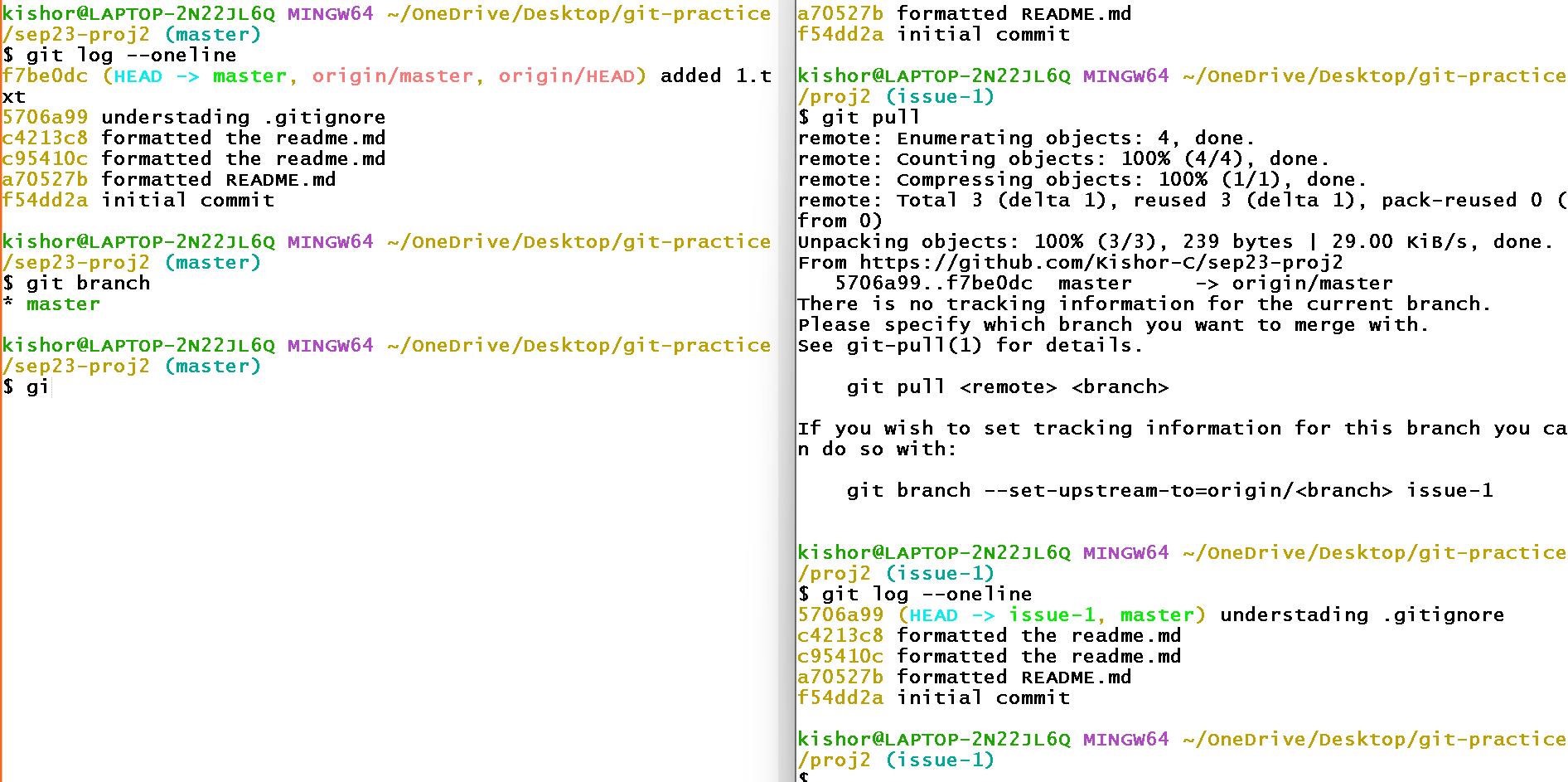


Notice the HEAD which is pointing to the recent version at both left & right terminal, you can observe left terminal is having the latest update, however the right terminal has old updates.

Checkout to issue-1 in the right terminal and log the updates



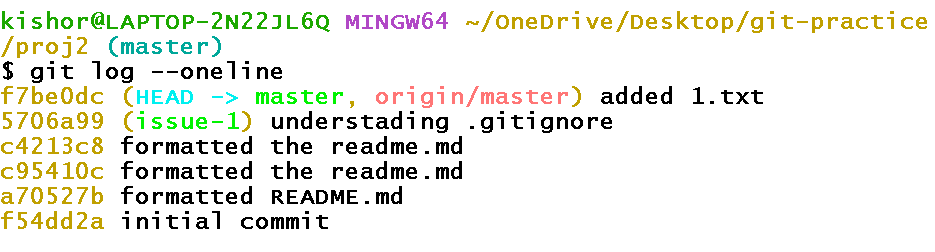
If you use git pull in the issue-1 you don’t see any changes coming from the remote, because there’s no branch with the name issue-1 in the remote



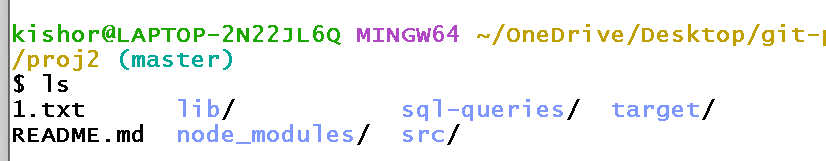
Observe in the 2nd terminal we still have old updates, what we can do is we can checkout to the master branch, pull the remote master & then checkout to the issue-1 branch and then merge so that issue-1 will have the latest update.



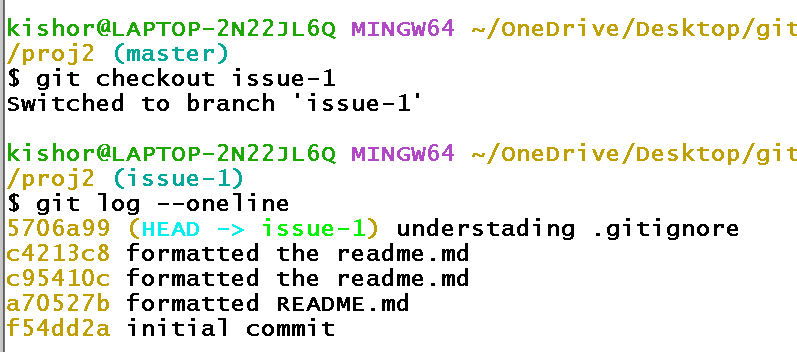
Now you can use git log and observe in the master branch the latest updates



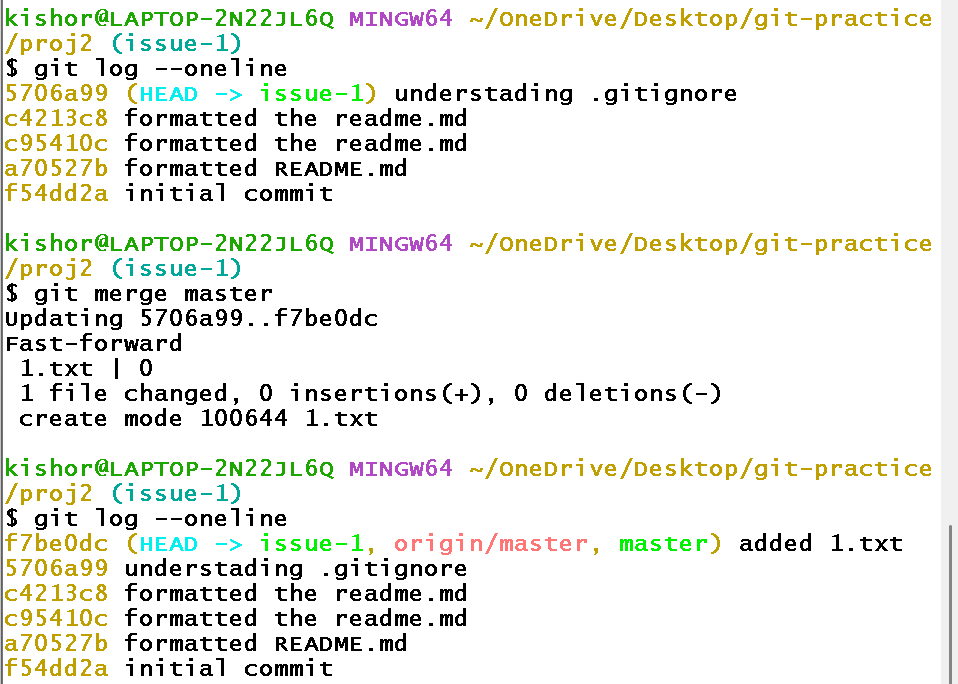
You can also use ls to see 1.txt file in the master branch



If you checkout to the issue-1 you will still have the old version, you must merge local master with issue-1 to get the new version



Notice you are still behind, you can use merge command



When does the conflict occur locally

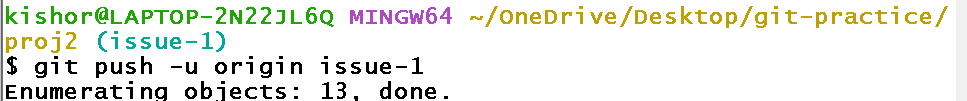
user1(master)-> update something in 1.txt -> push to the remote

user2(issue-1)-> update something in 1.txt & commit -> don’t push to the remote -> pull the remote -> try to merge the local master with issue1 -> you get conflict with a new branch (issue-1 | MERGING)

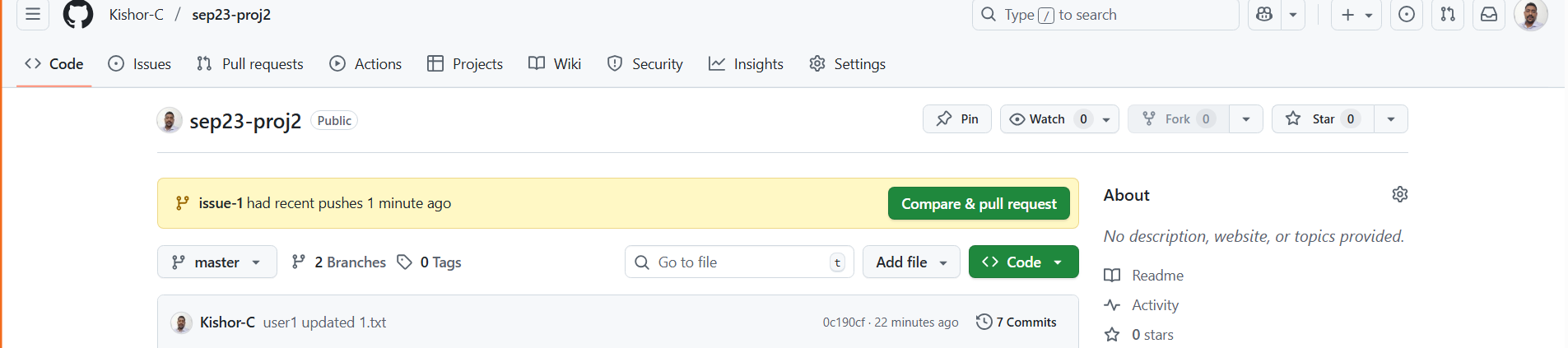
When you get the conflict fix the conflict and enter commit message



Push the issue1 to the remote so that everyone gets the latest update



Now in the remote you must merge the issue-1 to the origin/master



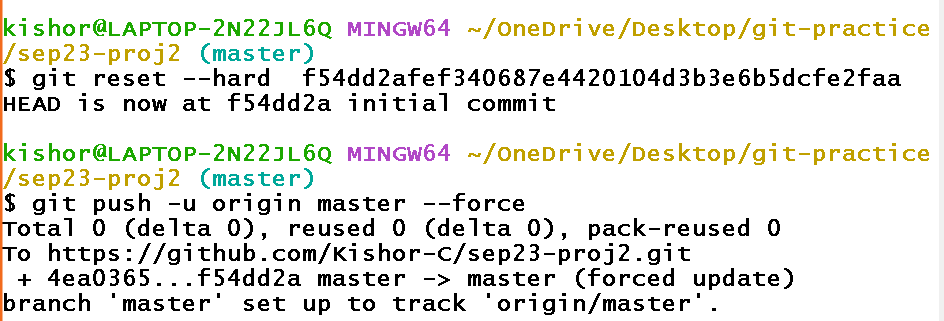
Note: In some case you may not see Compare & pull request, in that can you can manually click on the Pull requests tab and merge the master branch with the feature branch.

Activity:

1. user1(master) -> edit 1.txt -> add to staging area -> commit -> push to the remote
2. user2(issue2) -> edit 1.txt -> commit -> push to the remote
3. origin/master -> merge master with issue2 -> you will get a conflict -> close the merge request
4. user2(master) -> pull the origin/master -> switch to issue2 -> merge the local master -> resolve the conflict -> create a new commit -> push the issue2 to remote
5. remote/master -> merge the issue2 with the remote/master -> this time it succeeds
6. user1(master) & user2(master) will pull the origin/master

Hint: Merge command is “git merge master”

Git reset: It is used to go back to any of the previous commit id



fetch & pull

fetch only downloads the changes from the remote repository but doesn’t merge the changes

pull does fetch + merge

Summary of Git

Git vs Git bash vs Git hub:

GIT is a distributed version control system, which provides set of commands to manage the repository

GIT bash is a command line tool to enter git commands.

GIT hub is a cloud platform which maintains remote repositories

Git commands

git init, git add, git status, git commit, git push, git pull, git log, git branch, git checkout, git clone, git config, git fetch, git merge, git stash, git add remote, git stash, git reset

Day 3

MySQL: It is a relational database owned by oracle to maintain the data in the database in a tabular format.

What is a database

Database is a record to maintain the data in a structured way

What is Database Management System

It is a tool that provides user an interface to interact with the database to perform operations like create, update, delete, read

There are various of DBMS

1. Object -oriented
2. Hierarchical
3. Relational

RDBMS: It stands for Relational Database Management System that maintains the data in a table format which will have rows & columns

List of RDBMS software’s

1. Oracle Database
2. MySQL Database
3. PostgreSQL
4. Derby
5. MS SQL
6. DB2

The language RDBMS understand is SQL (Structured Query Language)

There are 5 types of SQL commands you can enter

1. DDL - Data Definition Language - create, alter, drop & truncate
2. DML - Data Manipulation Language - insert, update & delete
3. DQL/DRL - Data Query/Retrieval Language - select
4. DCL - Data Control Language - grant & revoke
5. TCL - Transaction Control Language - commit, rollback, savepoint

MySQL datatypes

* For Numbers: int, bigint, float, double
* For characters: char, varchar
* For Date & Time: date, timestamp
* For binary/character large object: blob, clob

Create command

It is a command used to create a table

Syntax:

create table table\_name(column\_name type [constraint], column\_name type,…)

In MySQL you need to create a database name using the below command

create database database\_name;

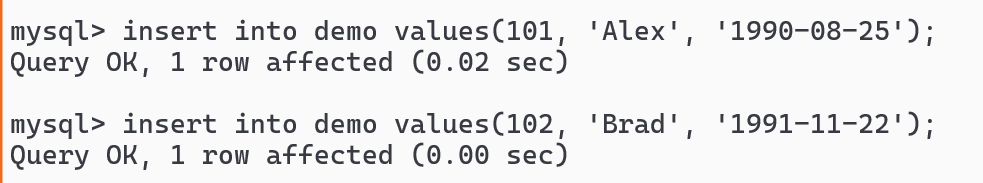


create a demo table



Insert Query

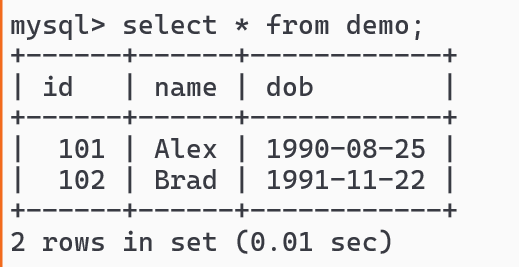
insert into table\_name(col1, col2) values(v1, v2);  
insert into table\_name values(v1, v2,…);



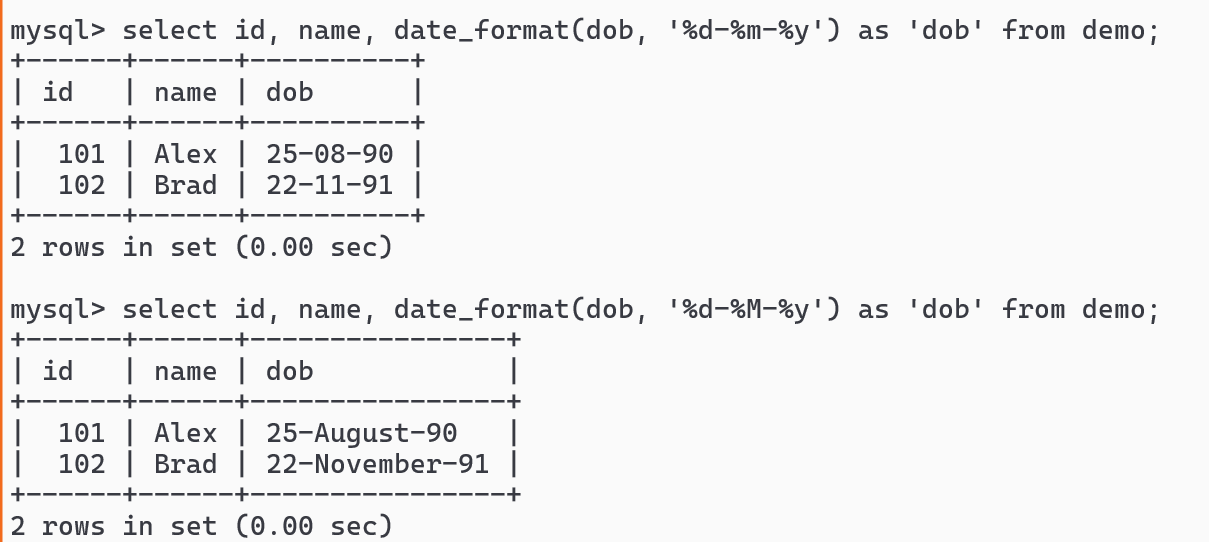
Select Query

select col1, col2 from table\_name;

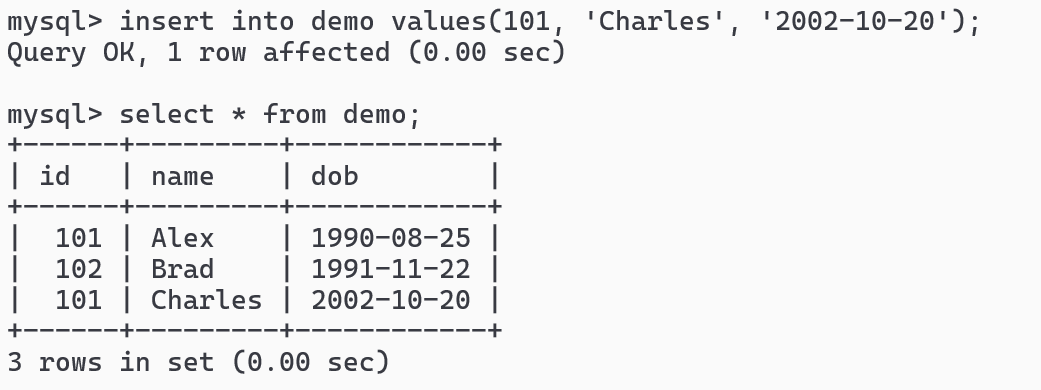
select \* from table\_name;



DATE\_FORMAT(value, ‘format’); This is a built-in function in MySQL to format the date, in the format option you must use %M for month, %d for date & %y for year



What happens if we give duplicate id in the demo table



Since there’s no constraint in demo table duplicate id’s are getting stored.

Constraints in database

Constraints are the rules you can apply on a table or a column to restrict the value, there are following constraints you can use in MySQL

1. PRIMARY KEY
2. UNIQUE
3. NOT NULL
4. CHECK
5. FOREIGN KEY

PRIMARY KEY: it is to uniquely identify the row, it doesn’t allow duplicates & null, mainly it is used for values that don’t change ex: employee\_id, customer\_id, account\_number

UNIQUE: It is to avoid duplicate values but it supports null, multiple rows can have null, each null is unique, ex: pan, aadhar, mobile\_number, email\_id

NOT NULL: It is used when a column mandatorily need value, ex: username, password, name

CHECK: It is used when a column needs to have the values that meets certain condition ex: designation, age, gender.

FOREIGN KEY: It is used when a table needs to have some dependent table i.e., customer & account table, employee & department table, customer & loan,

Adding primary key

1. You can add primary key while creating the table
2. If the table is already present you can use alter command to add the primary key, but the columns must not have any duplicates

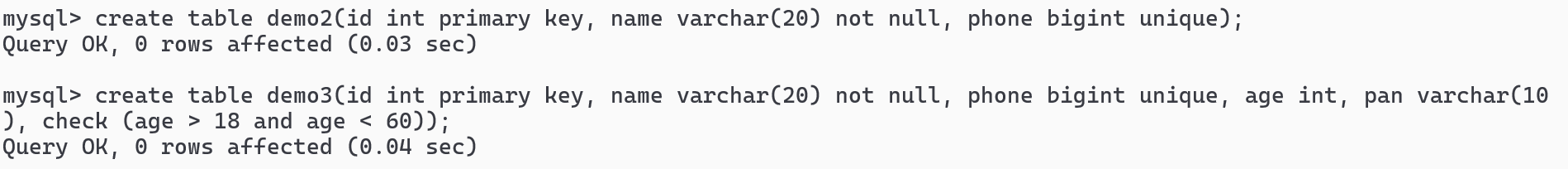
Create table with primary key

create table table\_name(col type primary key, col type, col type);

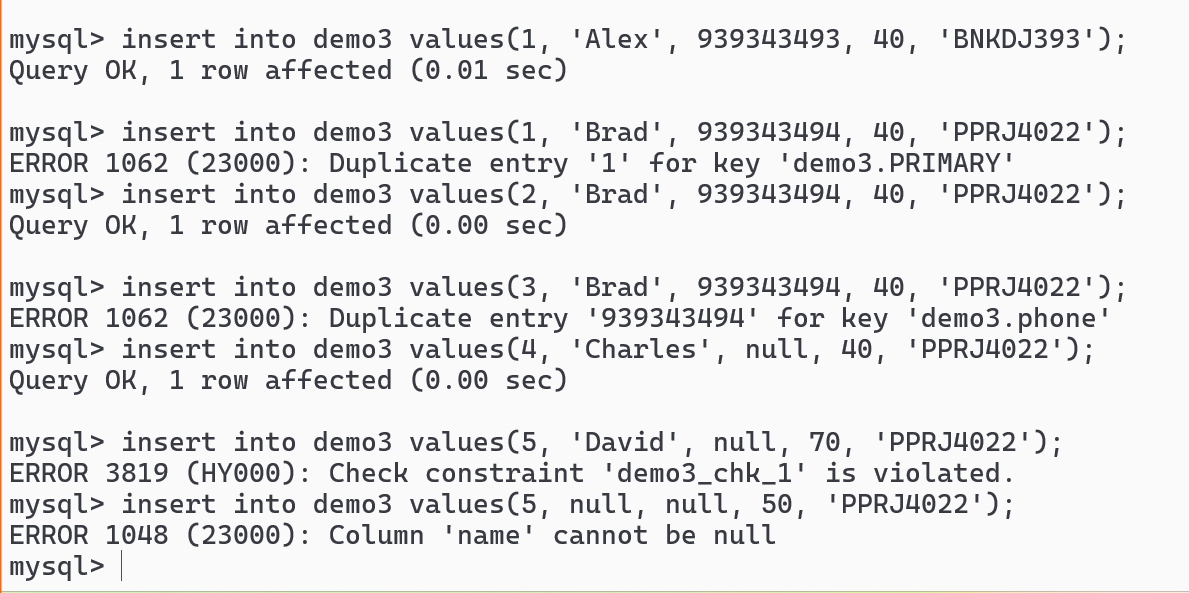
Create table with other constraints - not null, unique

create table table\_name(col type primary key, col type not null, col type unique);

Check constraint needs to be written after defining all the columns



Lets see what happens when the constraints are violated

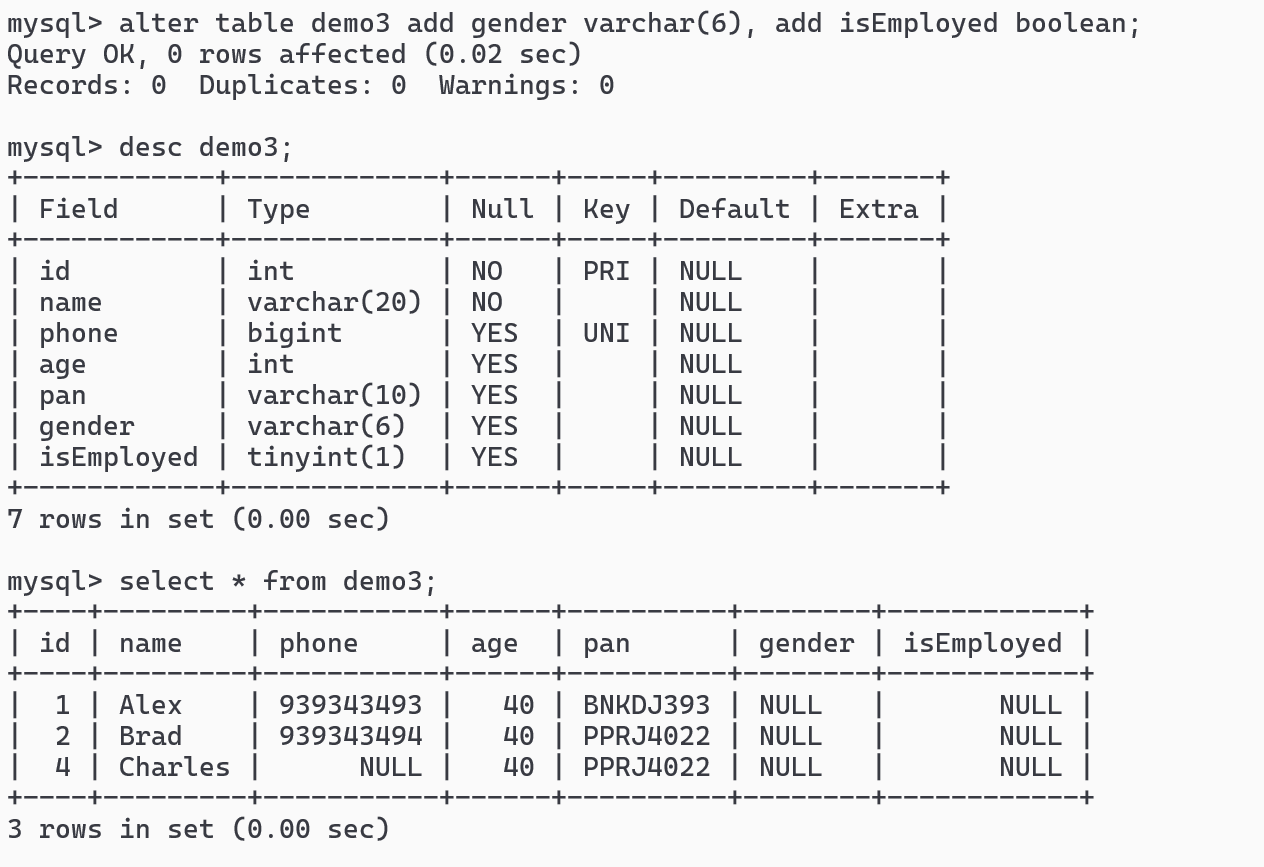


ADDING & DROPPING the columns

alter table table\_name add column\_name type, add column\_name type,…

alter table table\_name drop column column\_name, drop column column\_name

ADDING columns



DROPPRING column



ADD & DROP primary key

alter table table\_name add primary key(column);  
alter table table\_name drop primary key;



TRUNCATE

It deletes all the records from a table and you can’t undo(rollback) the changes

truncate table table\_name

DROP

It deletes the table & you can’t undo this

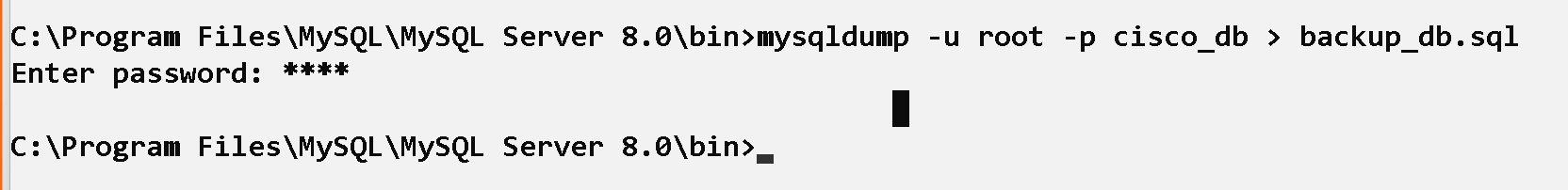
drop table table\_name;

Backup and restore

mysqldump -u username -p database\_name > backup\_db.sql

mysql -u username -p database\_name < backup\_db.sql

Backup command: Open cmd as admin in windows & for mac use sudo before the command



For mac-users: open a new terminal & enter below commands

**export PATH=”../../usr/local/mysql/bin:$PATH”**

In the same terminal: **mysqldump -u root -p cisco\_db > backup\_db.sql**

Login to the mysql

mysql -u root -p

Enter password

Switch to cisco\_db -> drop the demo table

Exit from mysql

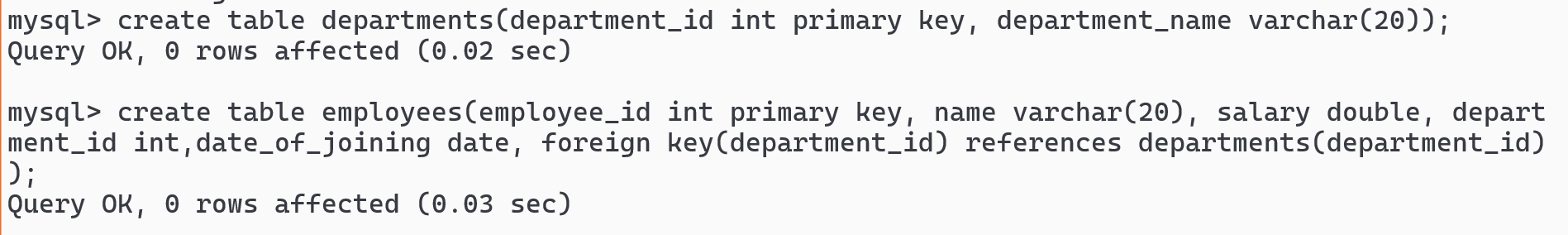
Restore the database backup file : **mysql -u root -p cisco\_db < backup\_db.sql**

Login to the mysql -> You must see the demo table in cisco\_db

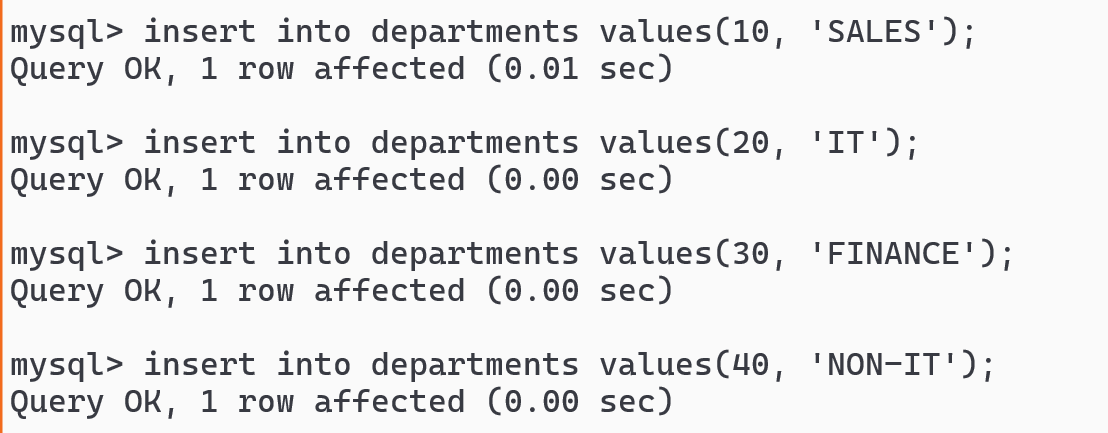
FOREIGN KEY

Whenever a table is related to another table you can use foreign key constraint, a table can have multiple foreign keys

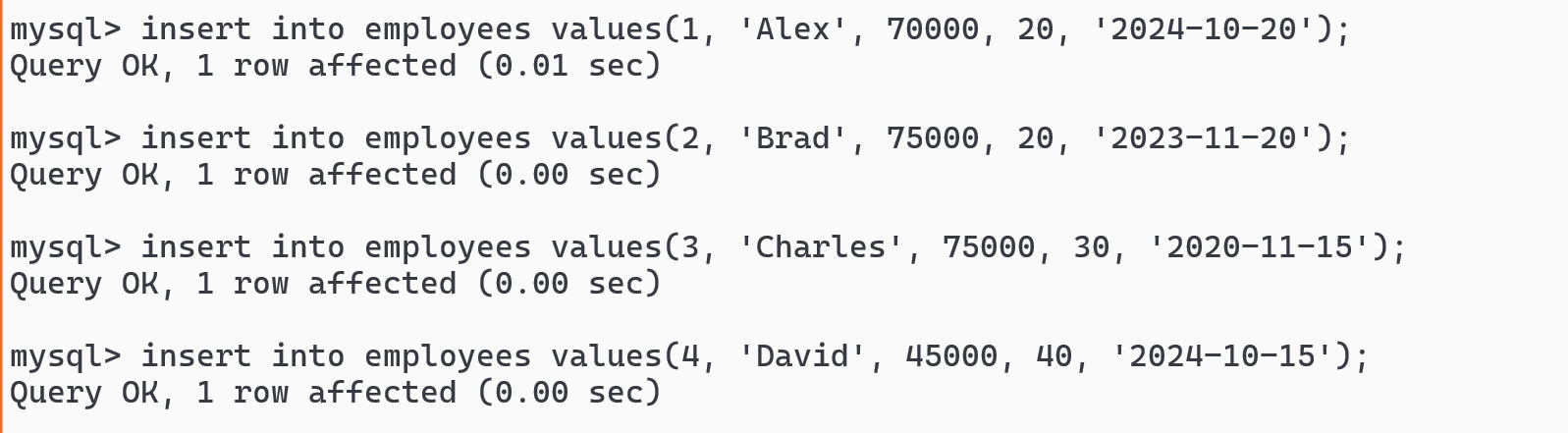
create table table\_name(col1 type, col2 type,..foreign key(column\_name) references parent\_table(primary\_key\_column)



Store some records in departments



store some records in employees

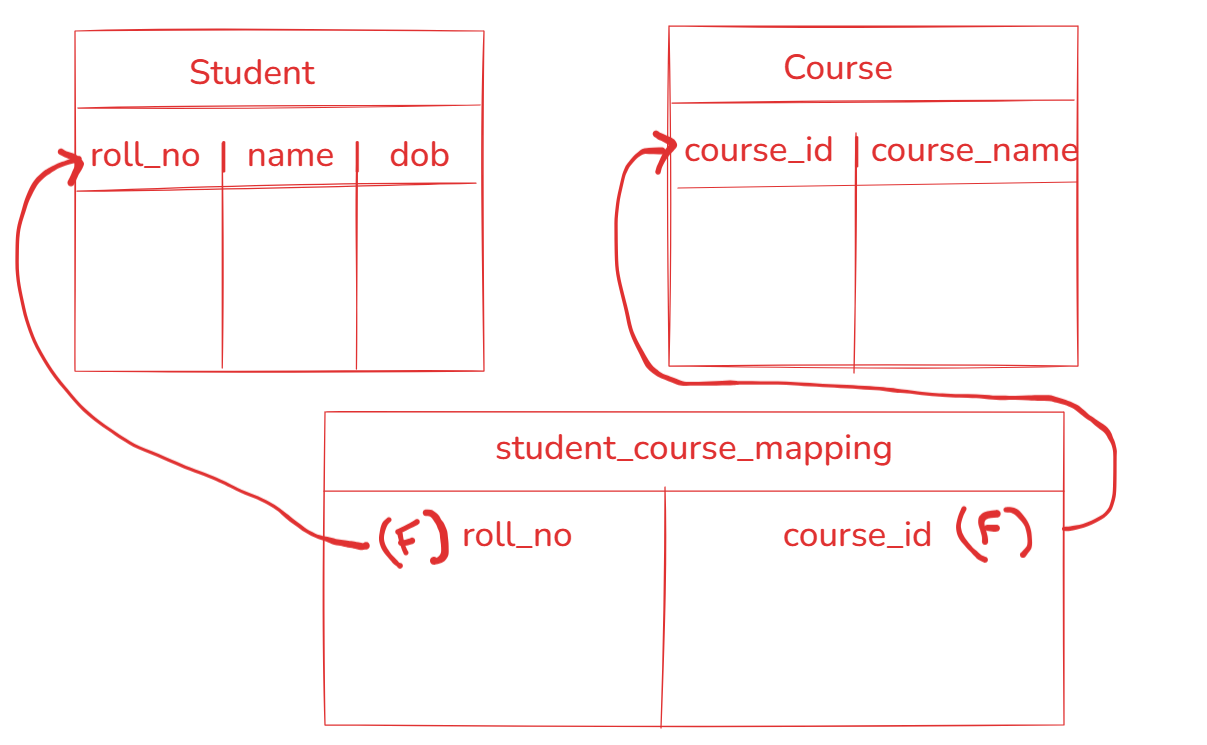


Try to store NULL in department\_id in employees table

Try to store a value that’s not part of deparments.department\_id

Activity

1. create student, course & student\_course\_mapping tables, student table will have student details, course table will have course details and the mapping table will have roll no and course id to specify which student has registered to which course, store some records in all the tables



Update & Delete commands

Update

update table\_name set col=value, col=value; # updates all the rows

update table\_name set col=value, col=value where <<condition>>; # updates only the rows matches to the condition

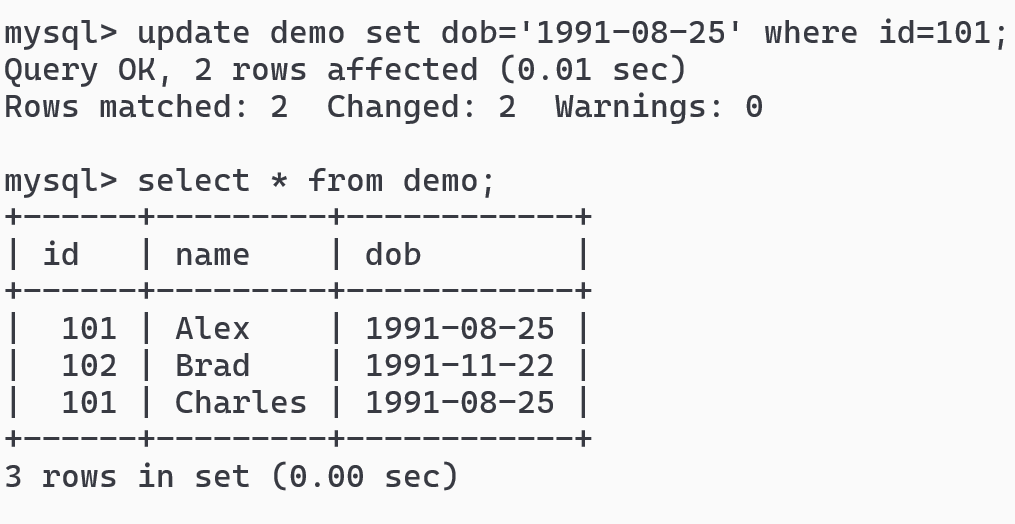
Delete

delete from table [or] delete from table where <<condition>>

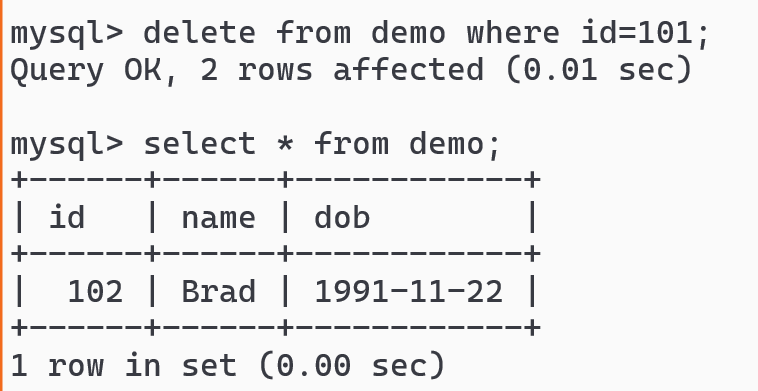
Activity on a demo table:

1. Update the dob of a row that matches to a particular id
2. Delete the row that matches to a particular id

Updating dob based on id



Deleting the row based on id

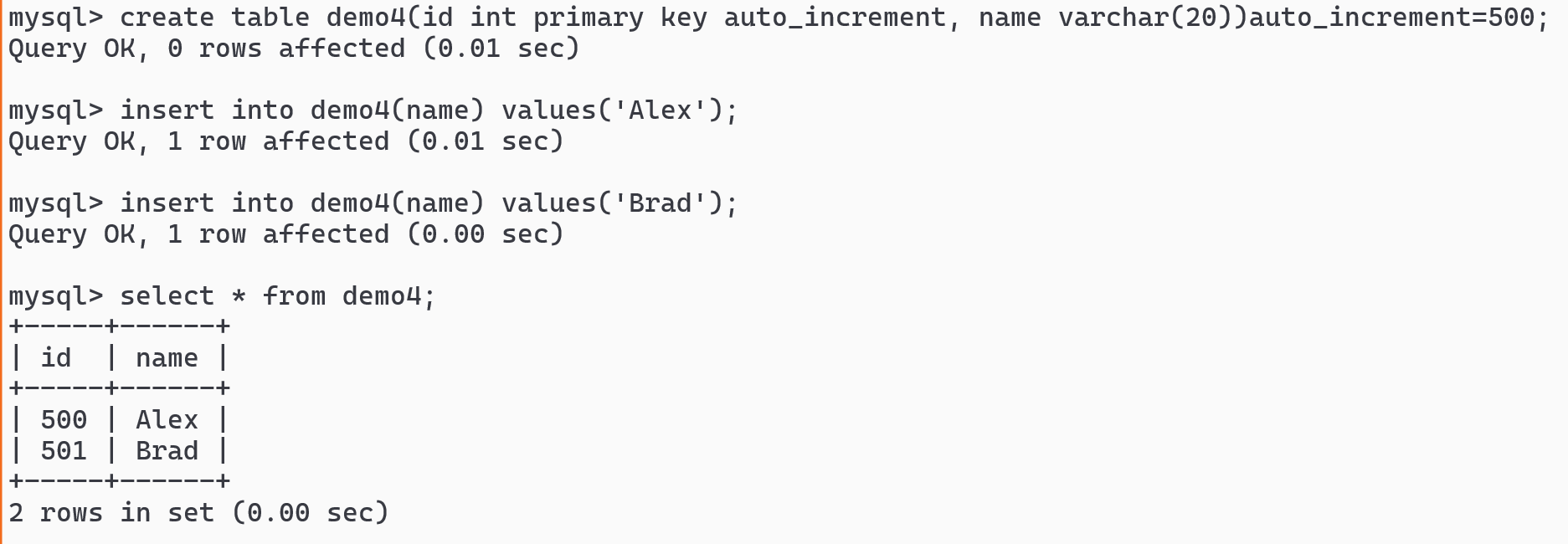


Auto\_Increment: This keyboard help primary key to automatically increment by 1

You can use this while creating a table or with alter command also.

create table table\_name(col type primary key auto\_increment, …) auto\_increment=100;

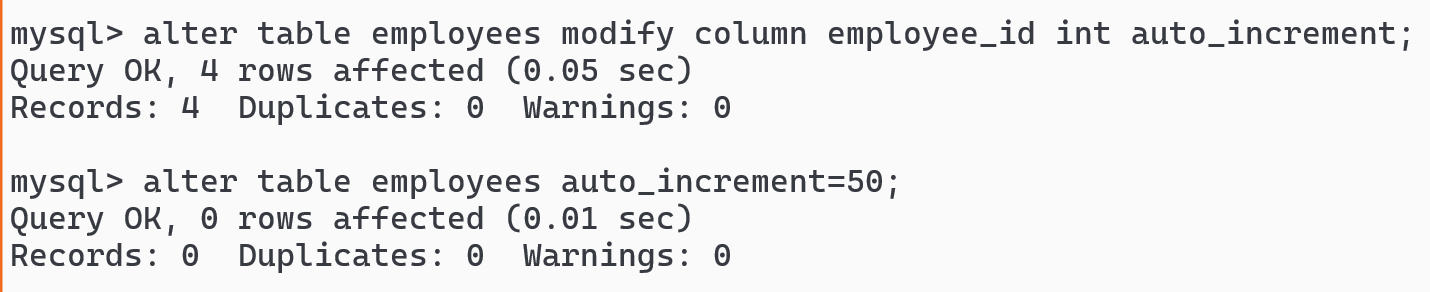
when you specify auto\_increment value then the value starts from the number you have specified else by default it starts from 1.



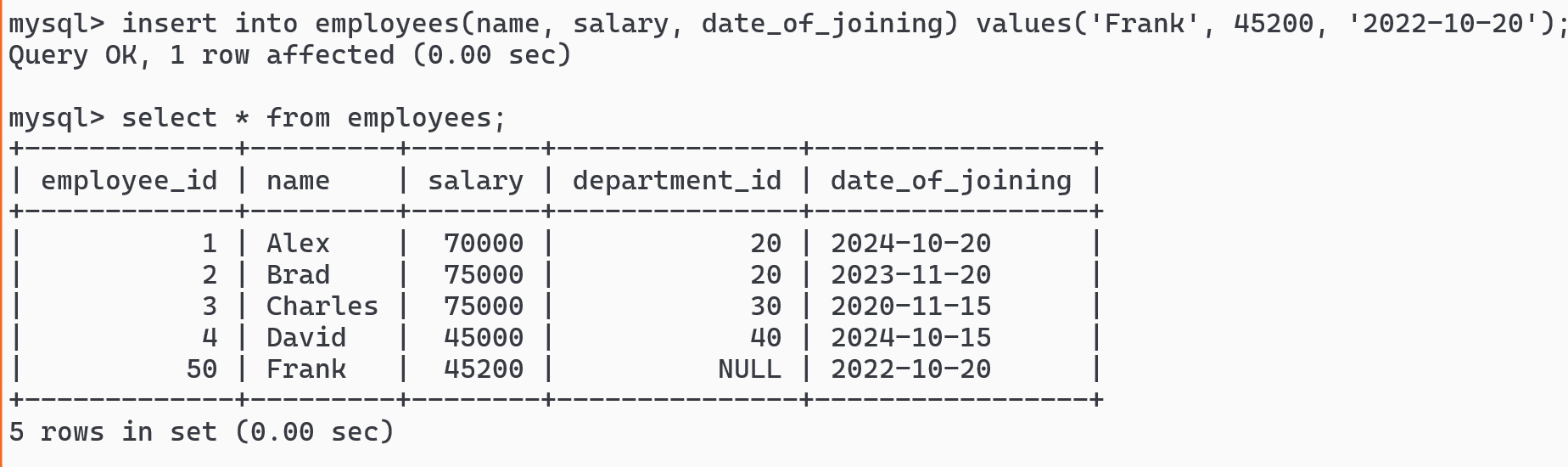
auto\_increment with alter command: when the table is already created but the column doesn’t have auto\_increment

alter table table\_name modify column column\_name type auto\_increment;

alter table table\_name auto\_increment=100;



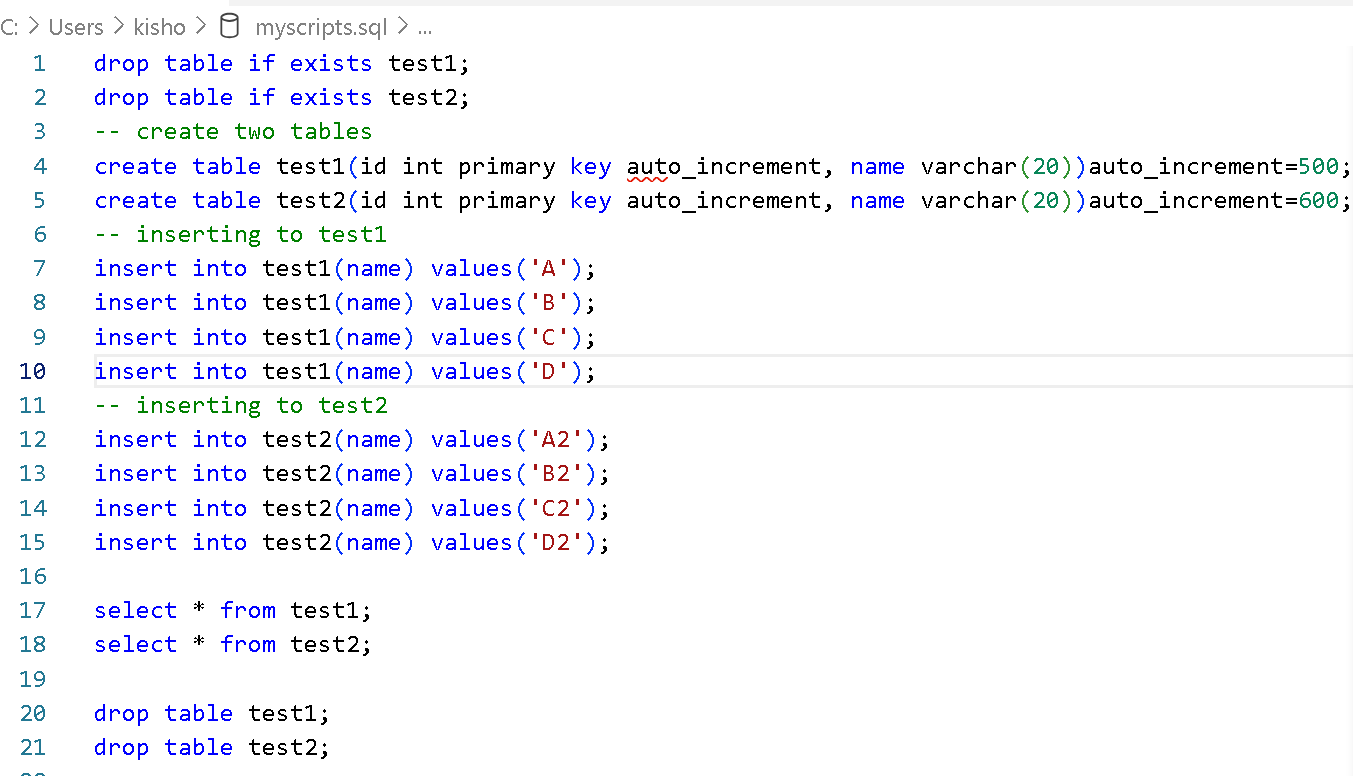
You can now the employees record with employee\_id

Running the SQL script files

You need to have an SQL file and mention the path of that SQL file using SOURCE keyword

SOURCE path-to-SQL-file

create myscripts.sql in users folder

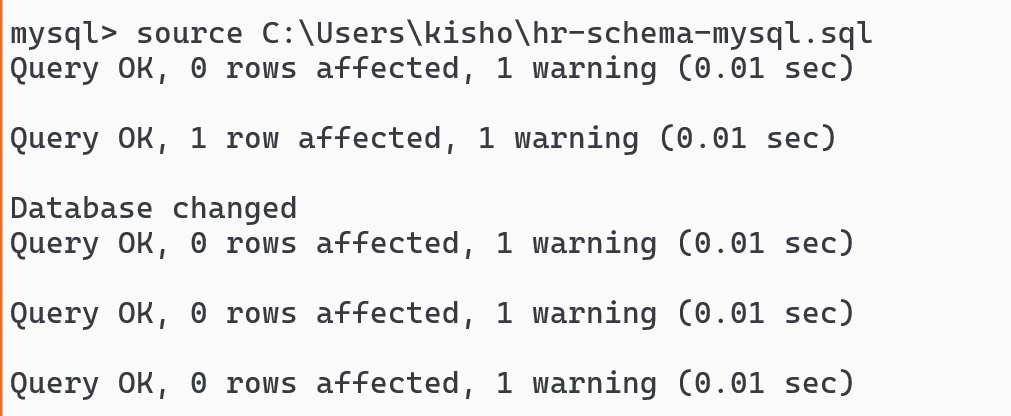




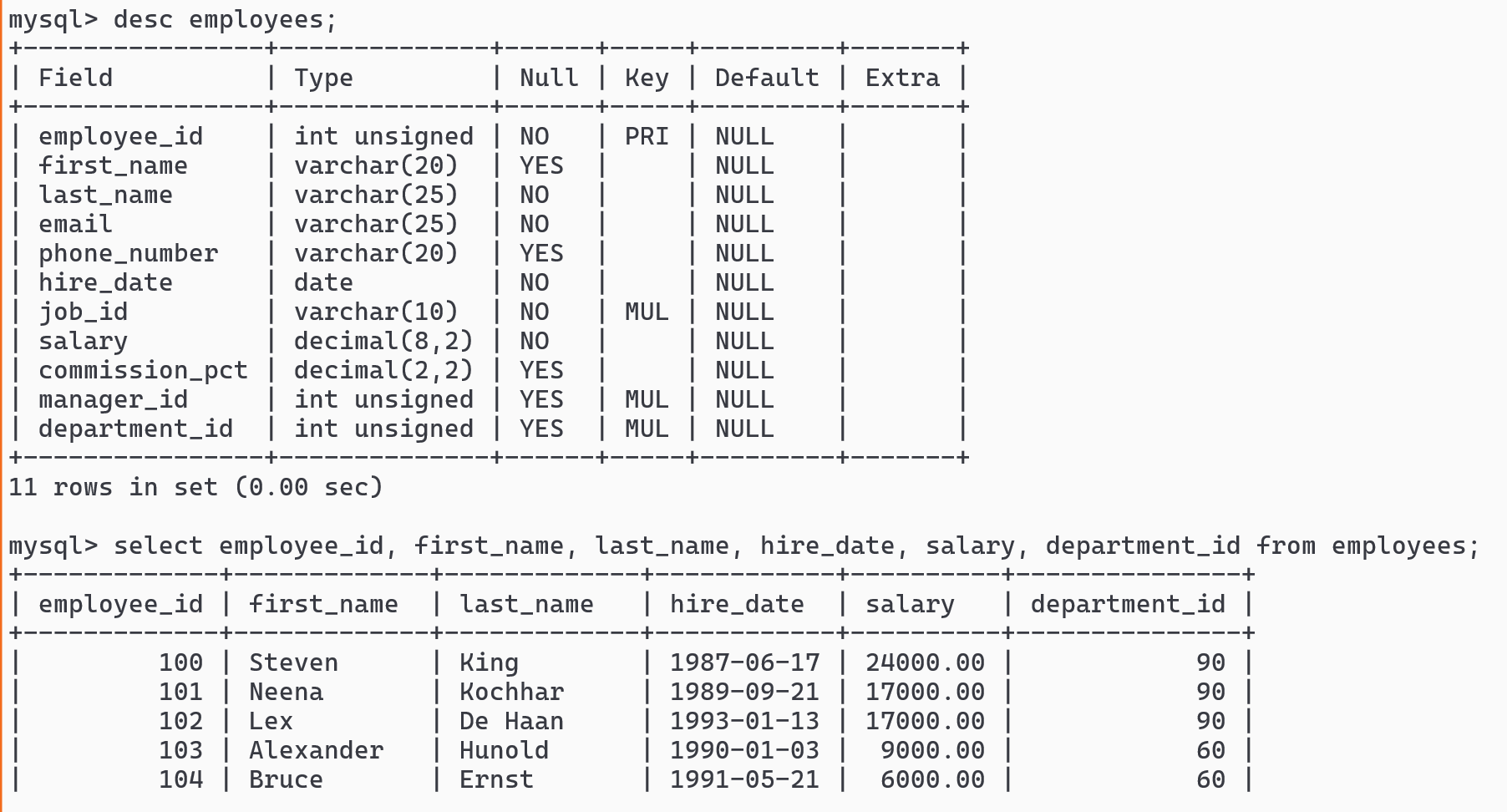
Download sample database from below URL

<https://github.com/nomemory/hr-schema-mysql/blob/master/hr-schema-mysql.sql>

using source run this script file



Describe employees and run select query on employees table



Built-in functions in MySQL

There are mainly two types of functions

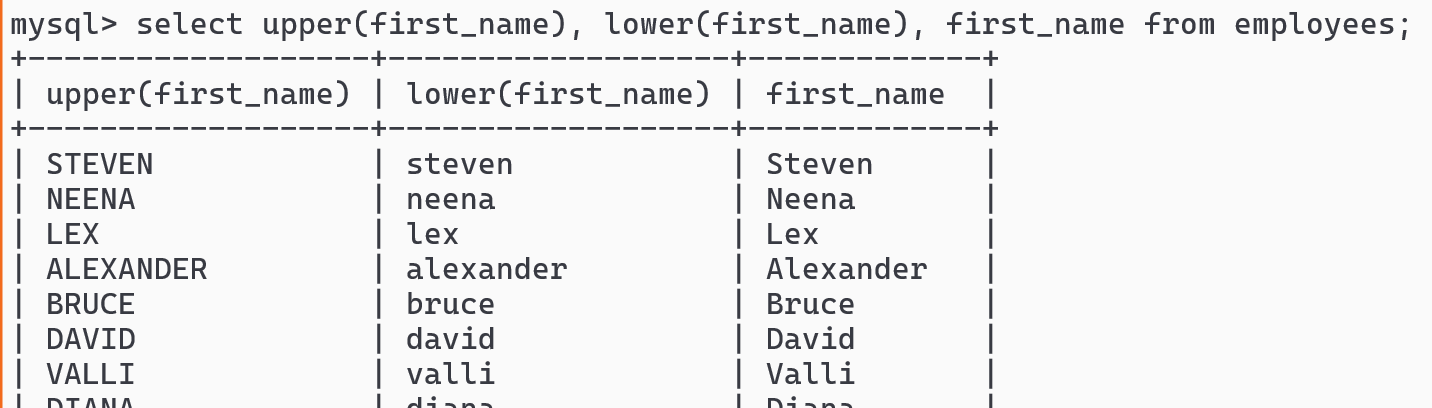
1. Single-row functions - for each row a result will be generated

upper(), lower(), concat(), length(), date\_format(), trunc(), ceil(), floor(),

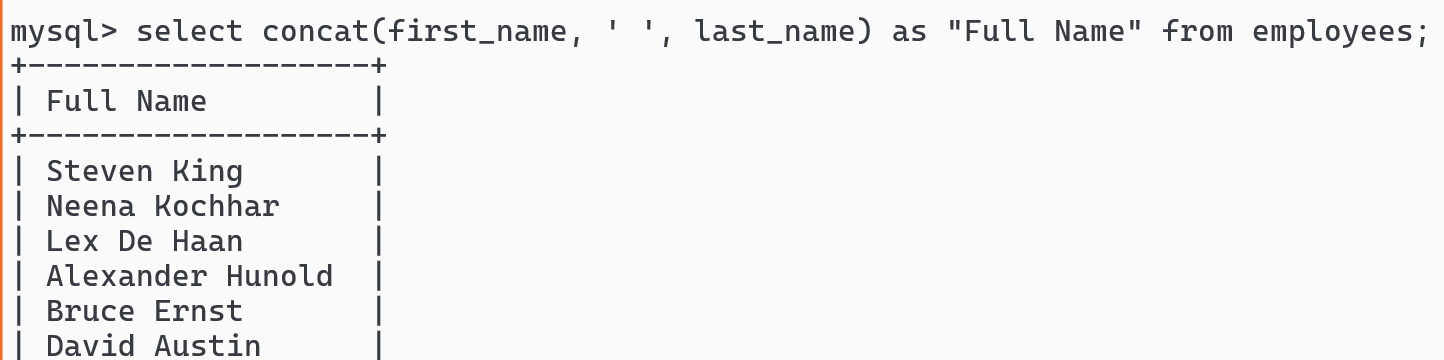
1. Aggregate functions - combines all the rows and generates a single result

sum(), max(), min(), count(), avg()

Using upper() and lower() functions

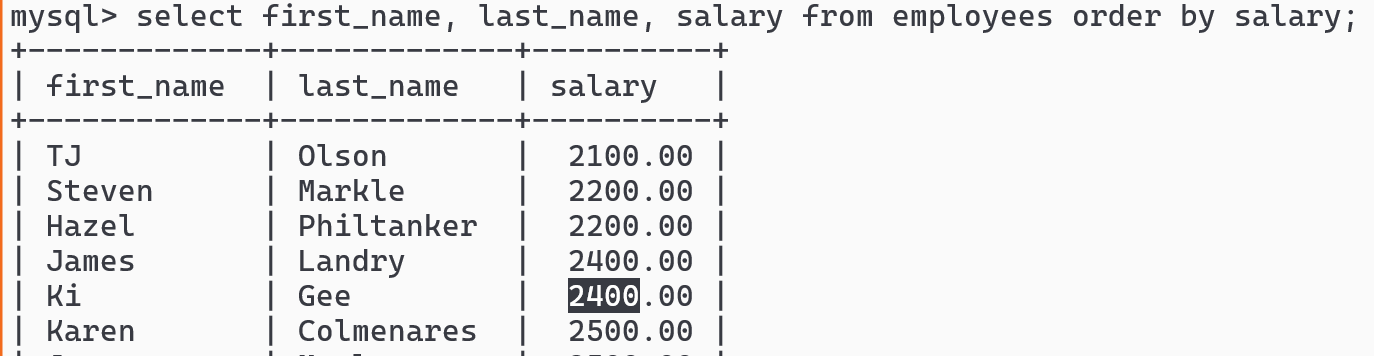


Using concat() function

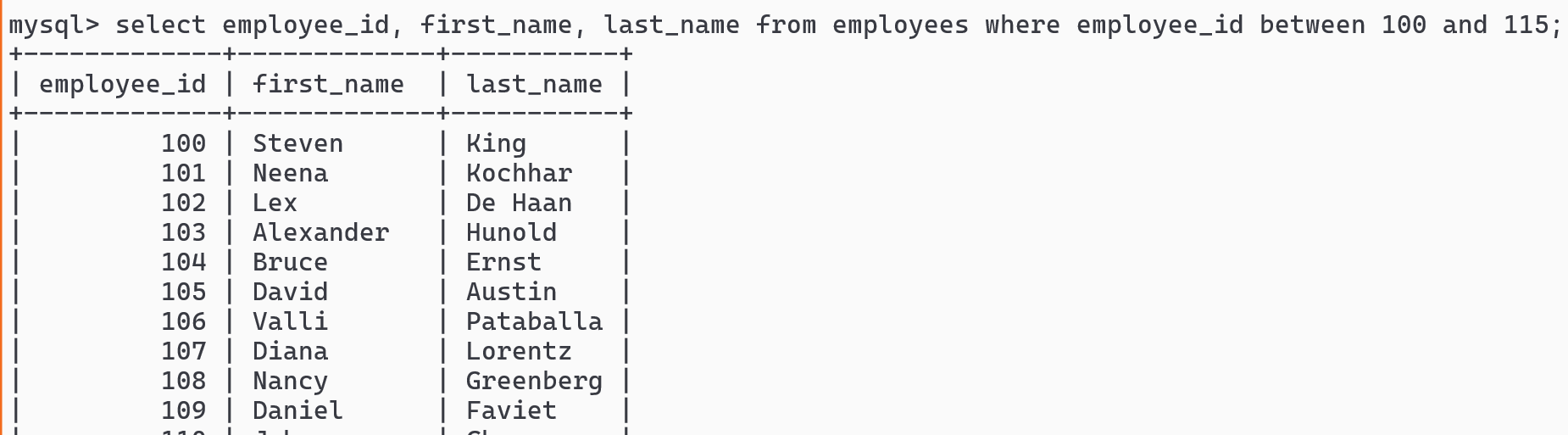


SELECT commands takes lot of clause like - WHERE, ORDER BY, BETWEEN, OR, IN, AND, HAVING, GROUP BY and etc.

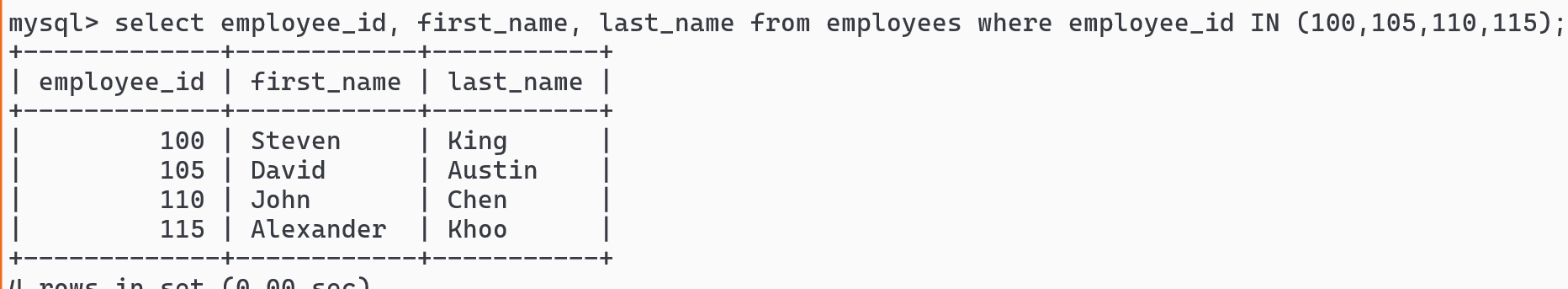
ORDER BY salary



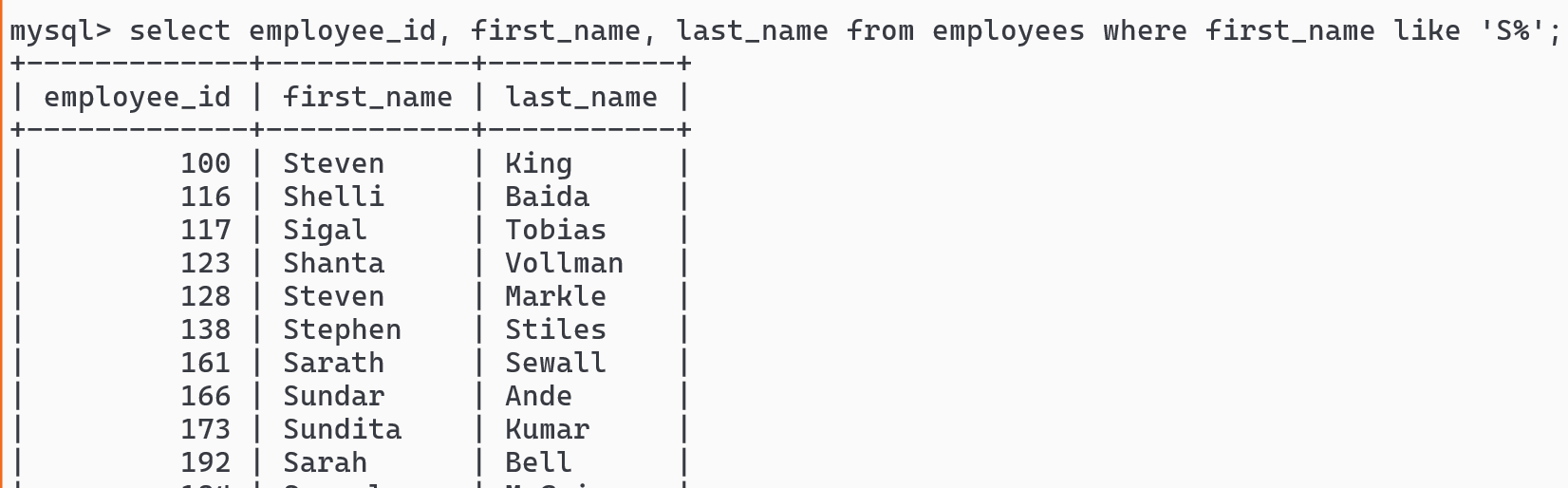
You can also use between



IN operator



LIKE operator

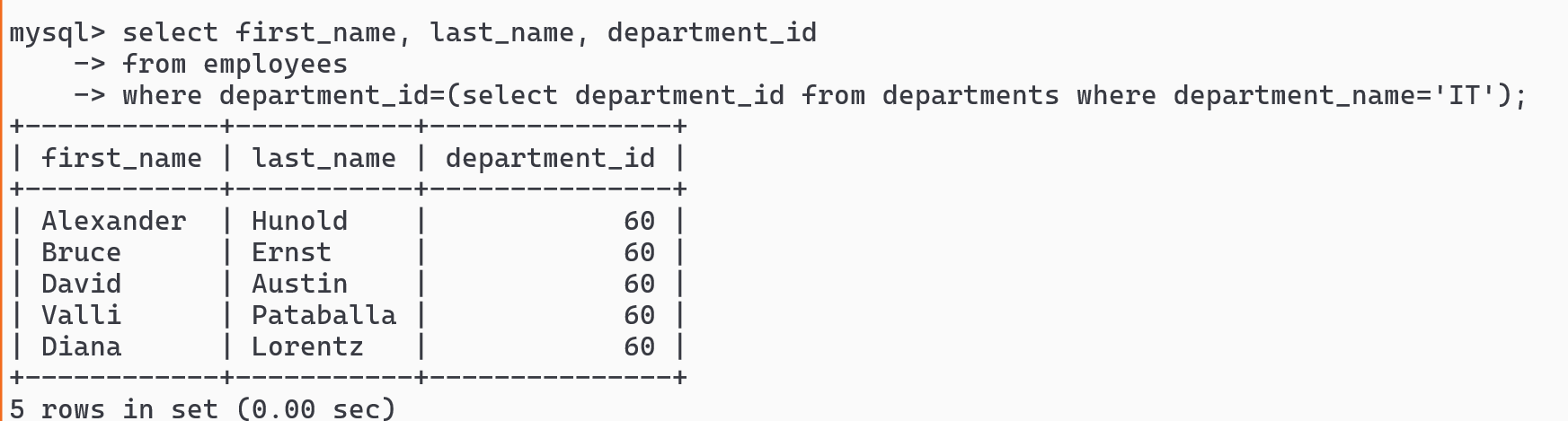


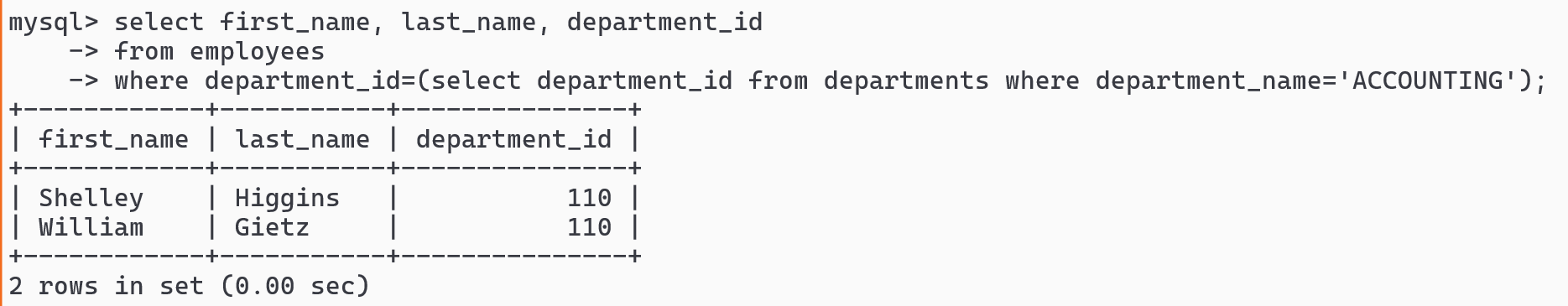
Sub-queries / Nested-queries

Sometimes a query needs a condition after executing another query and then give the result

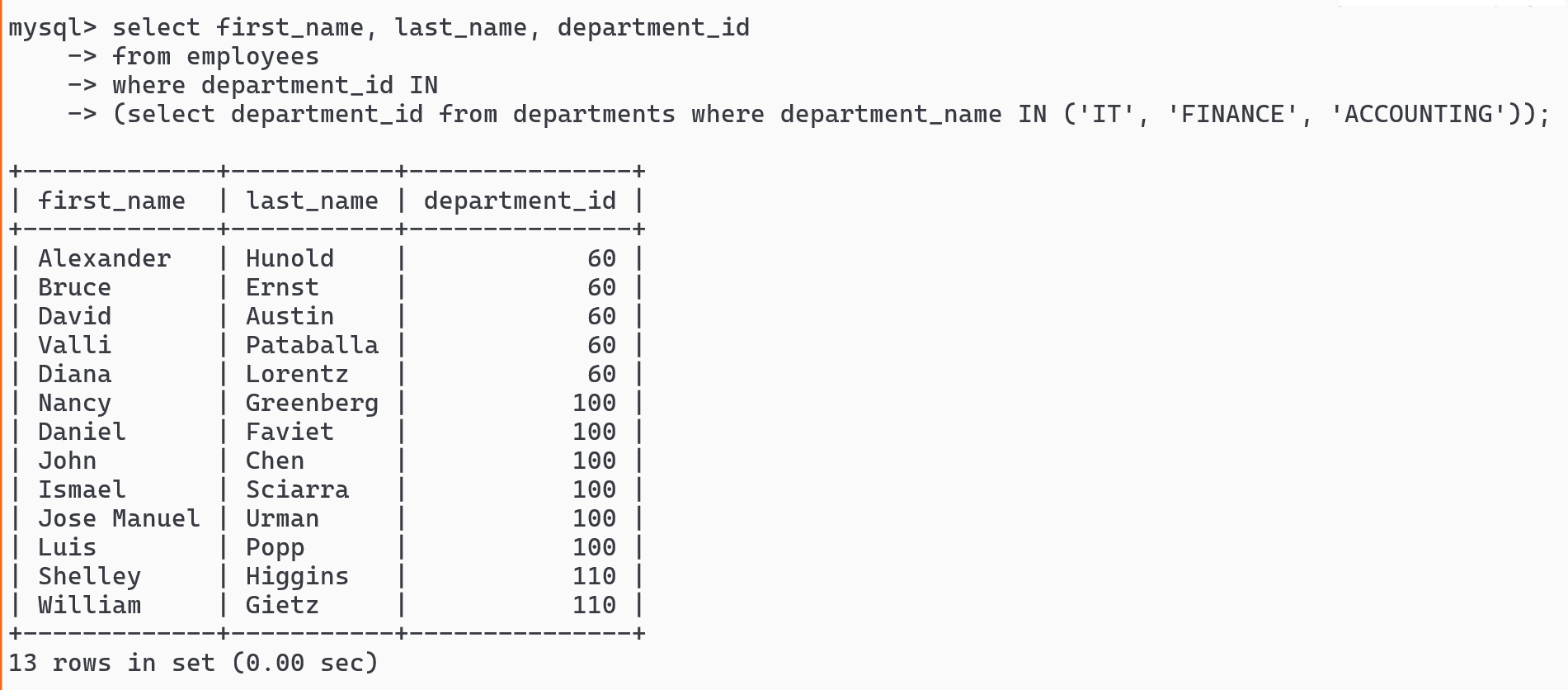
ex: You want all the employees belonging to sales, account & IT department

outer-query condition (inner-query)

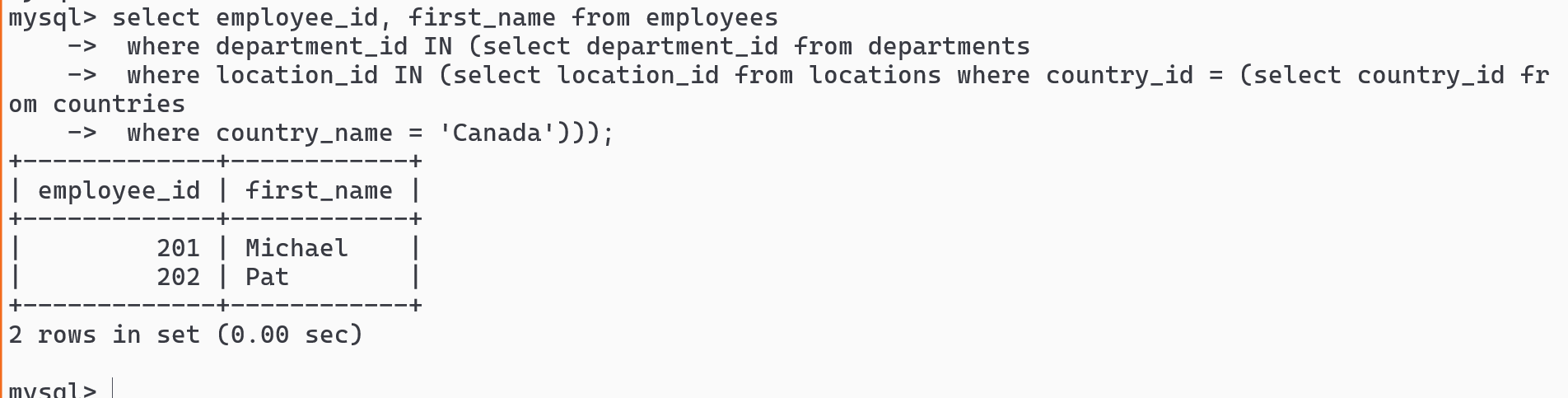
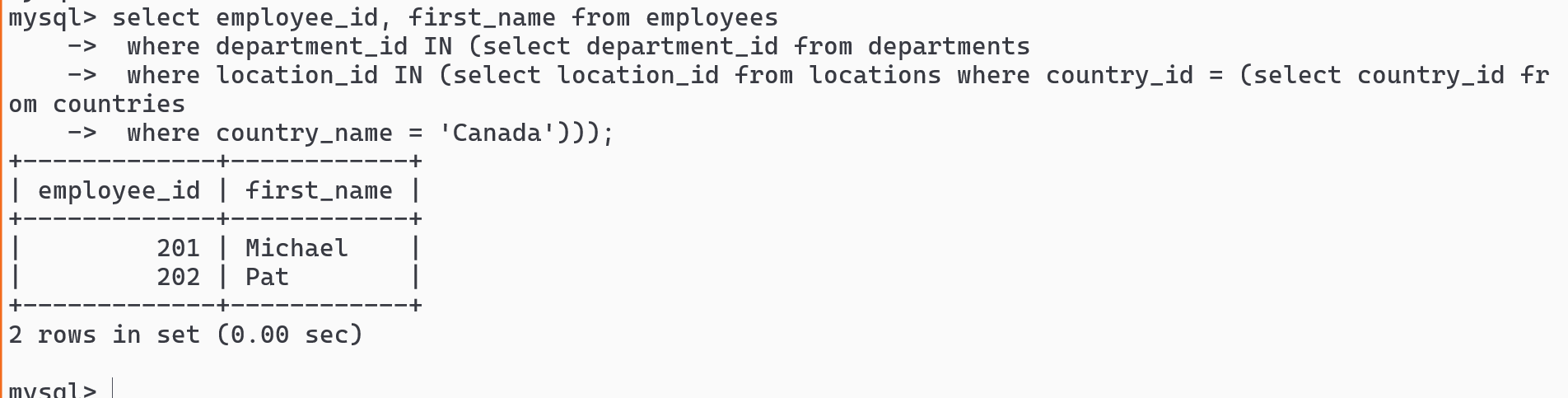
Employees belonging to ACCOUNTING



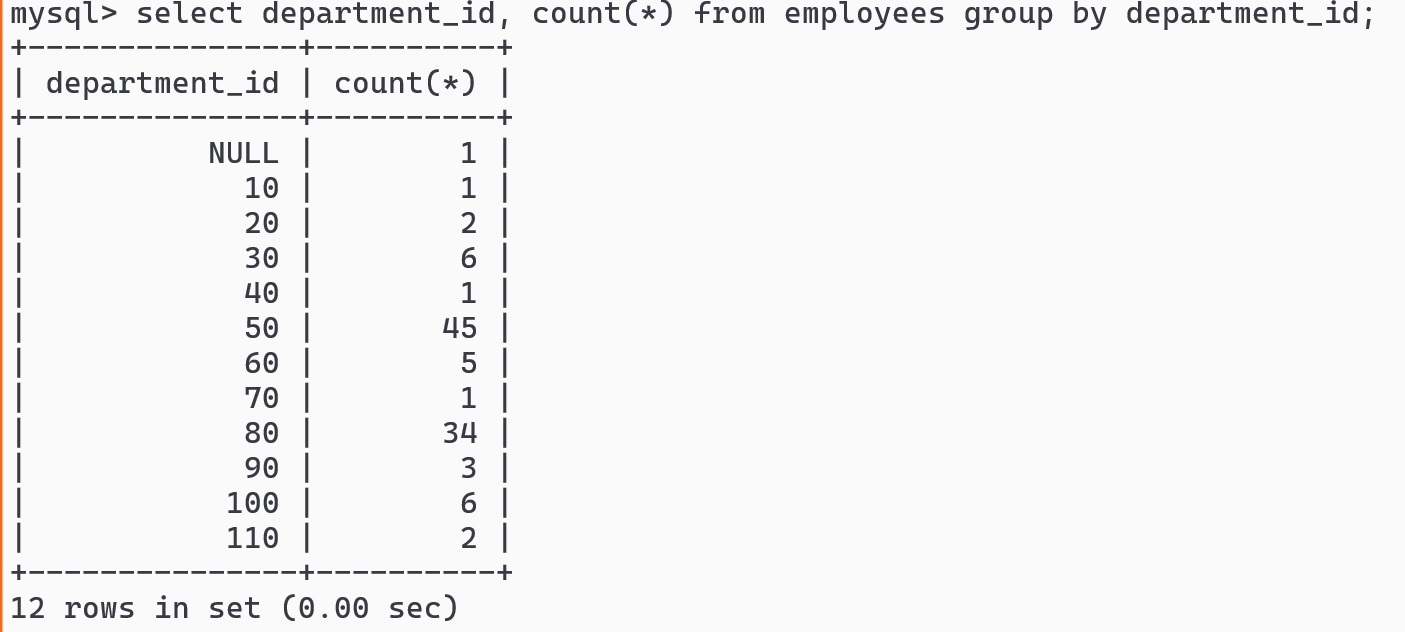
Show all the employees belonging to FINANCE, IT and ACCOUNTING



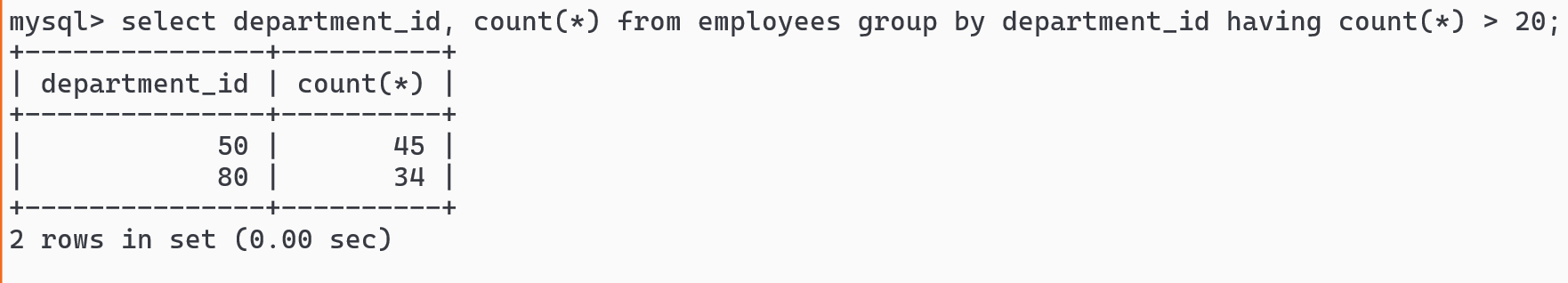
Show all the employees working in some country like Canada or United Kingdom



GROUP BY clause



Having clause



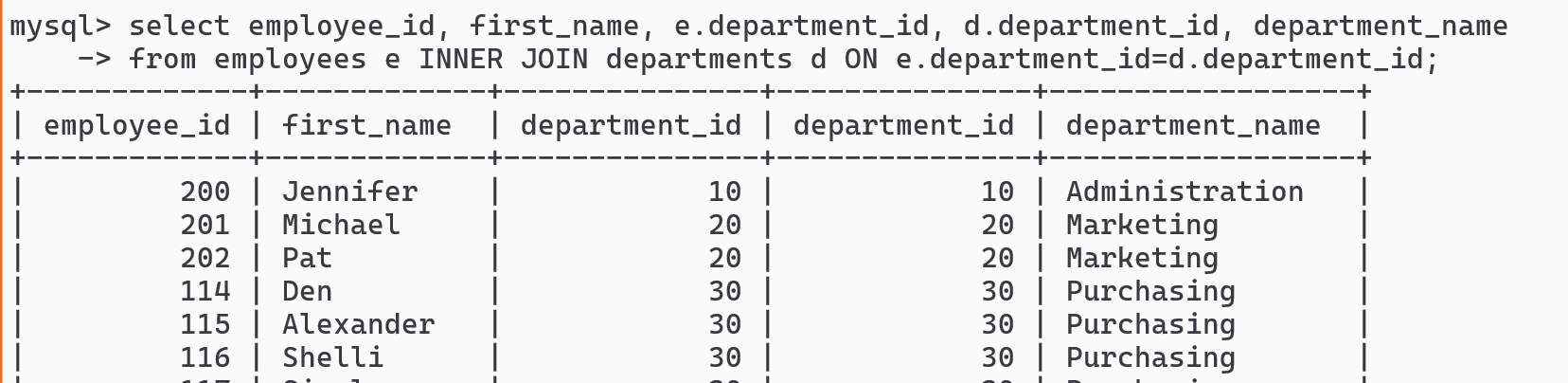
SQL Joins

It is used to join two or more tables and generate result based on some common value, types of join

1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. FULL JOIN
5. CROSS JOIN

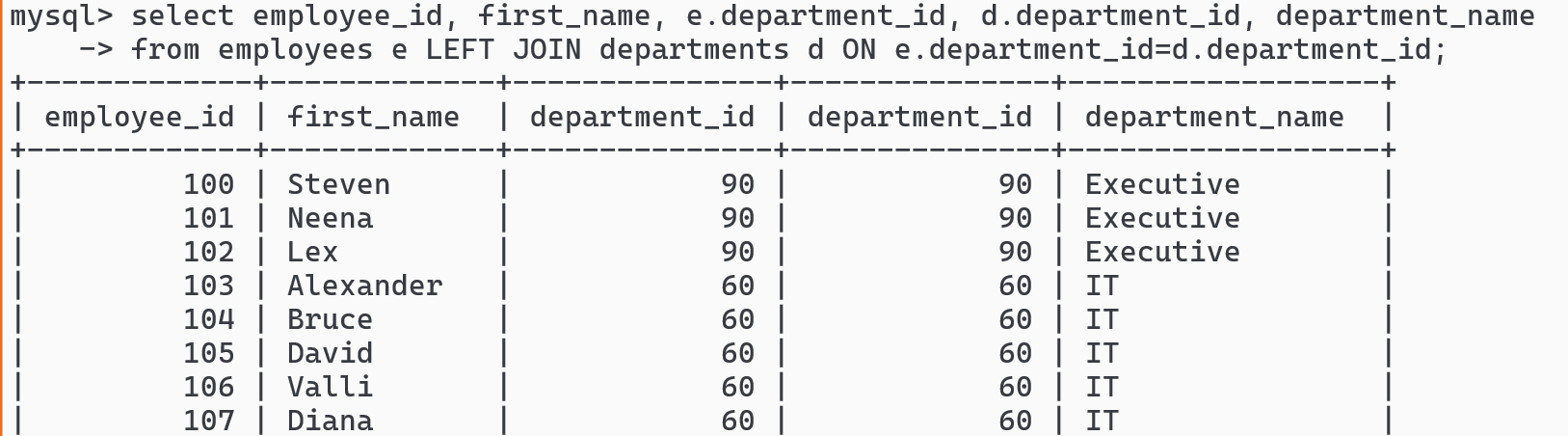
INNER JOIN

LEFT & RIGHT table gives only matching rows

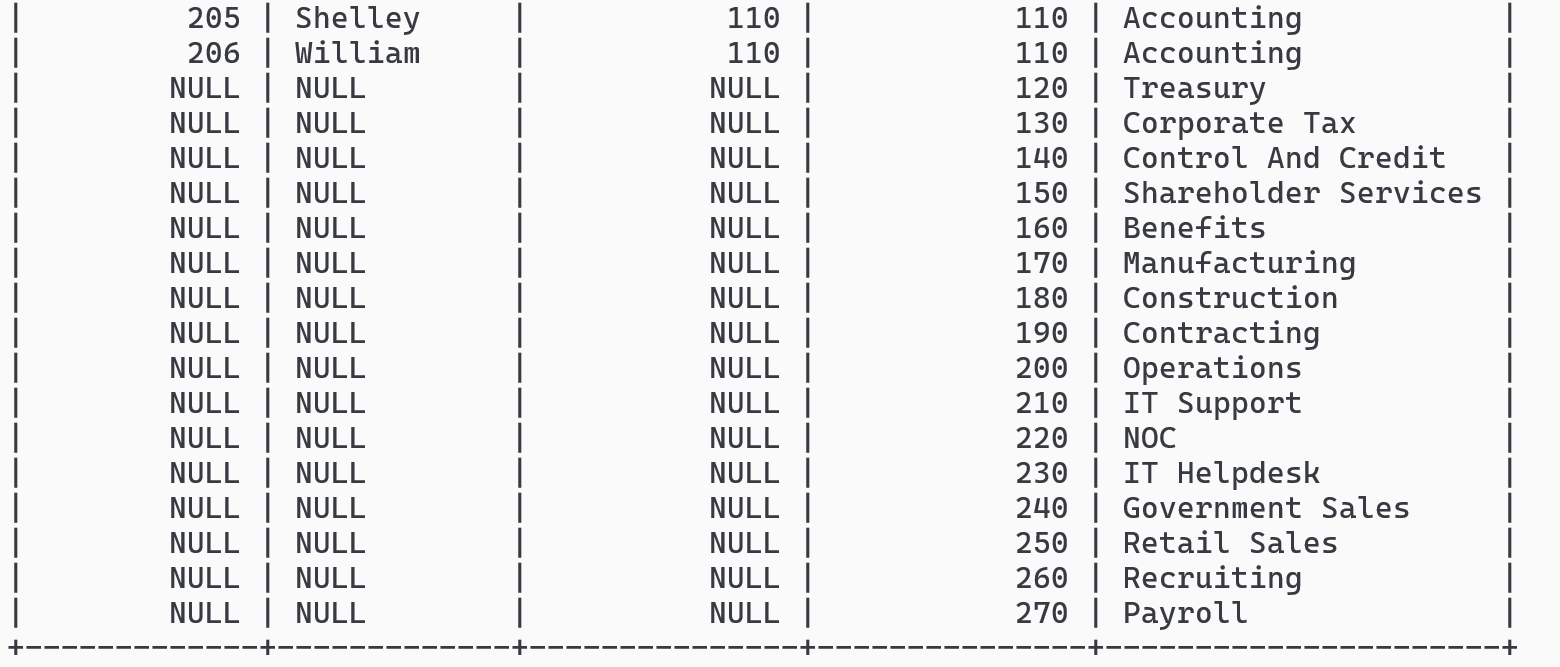
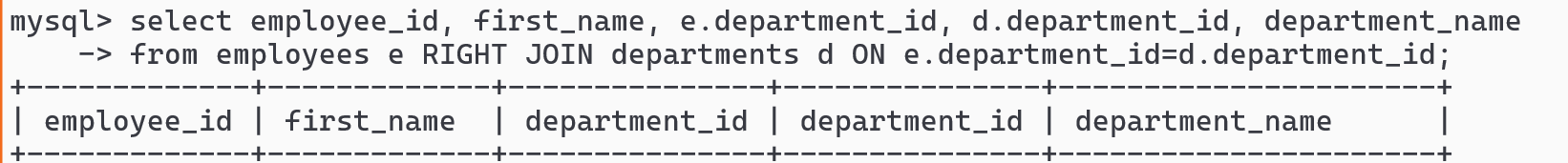


LEFT JOIN

All the left table rows will be listed & in right table only matching rows will be listed



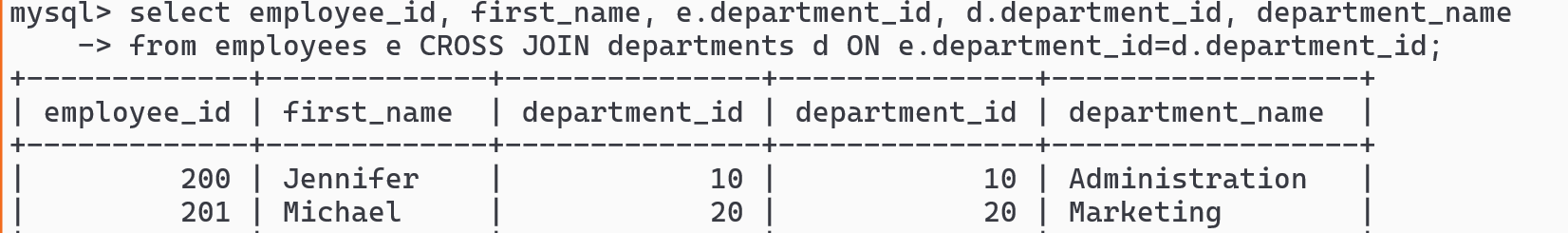
RIGHT JOIN

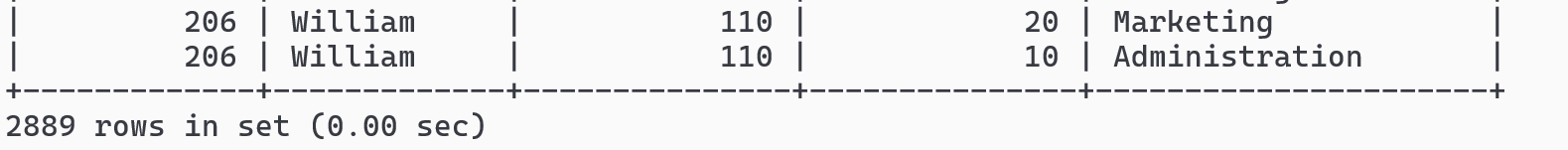


FULL JOIN: displays all the rows of left & right table

Not supported in MySQL

CROSS JOIN: Each row \* by other table rows M \* N





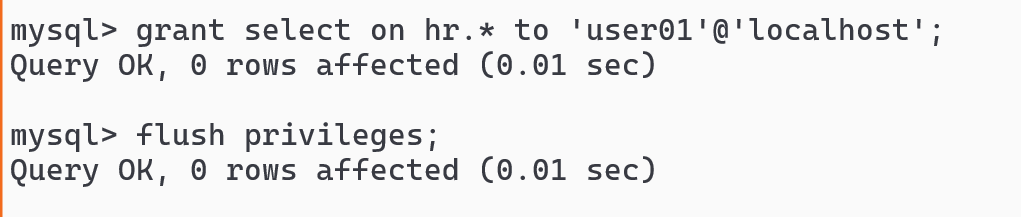
How to create users and give permissions for the users

Login as a root, create the user & grant permission to the user on a table

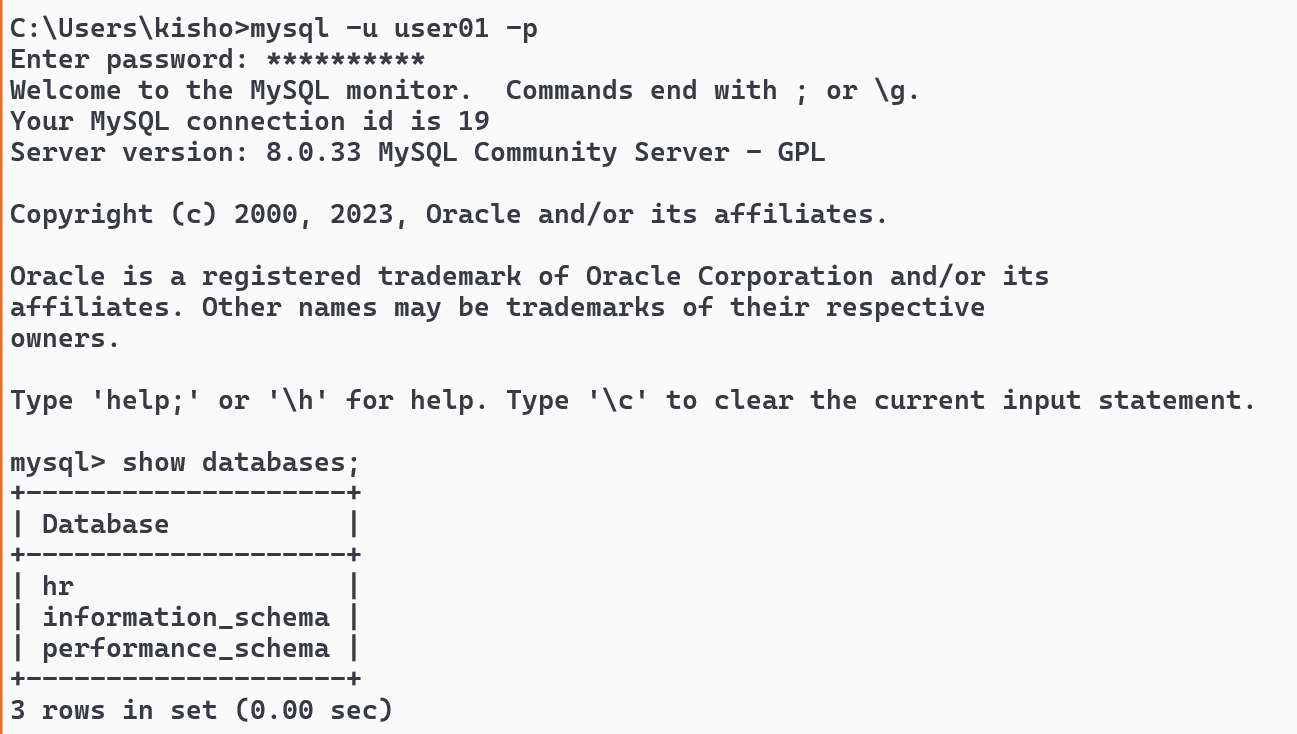
Creating the user



Grant only read permission to the user on a particular table



Login as user01

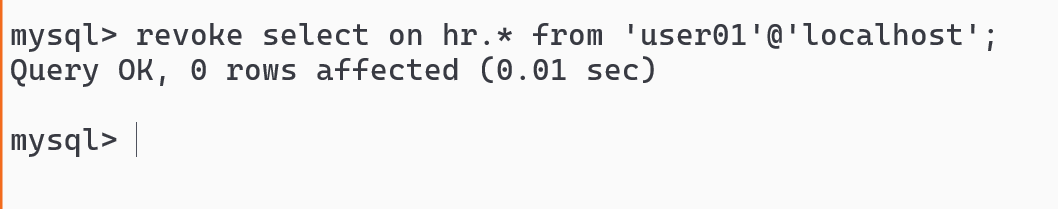


Enter select & delete commands

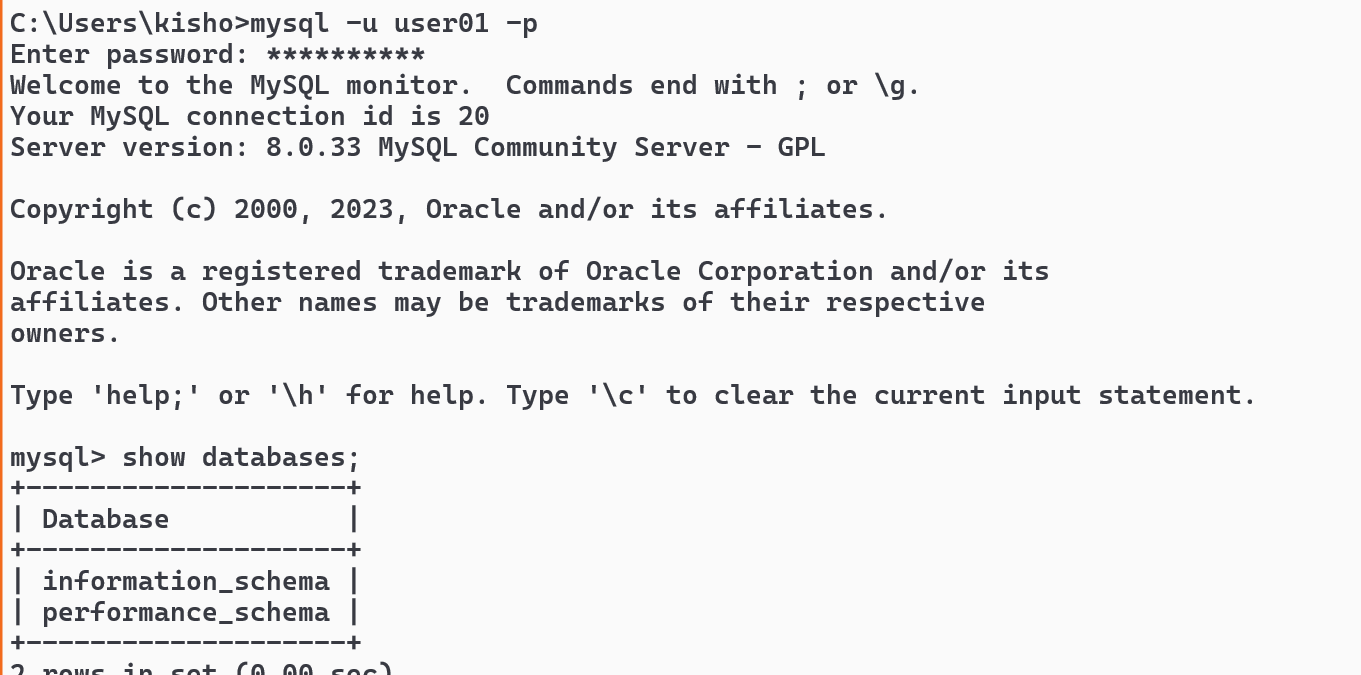


Revoke permission

From the root user - you must revoke



From the user01 try to enter select command - you must relogin



Note: User doesn’t see hr schema at all.

Summary of MySQL

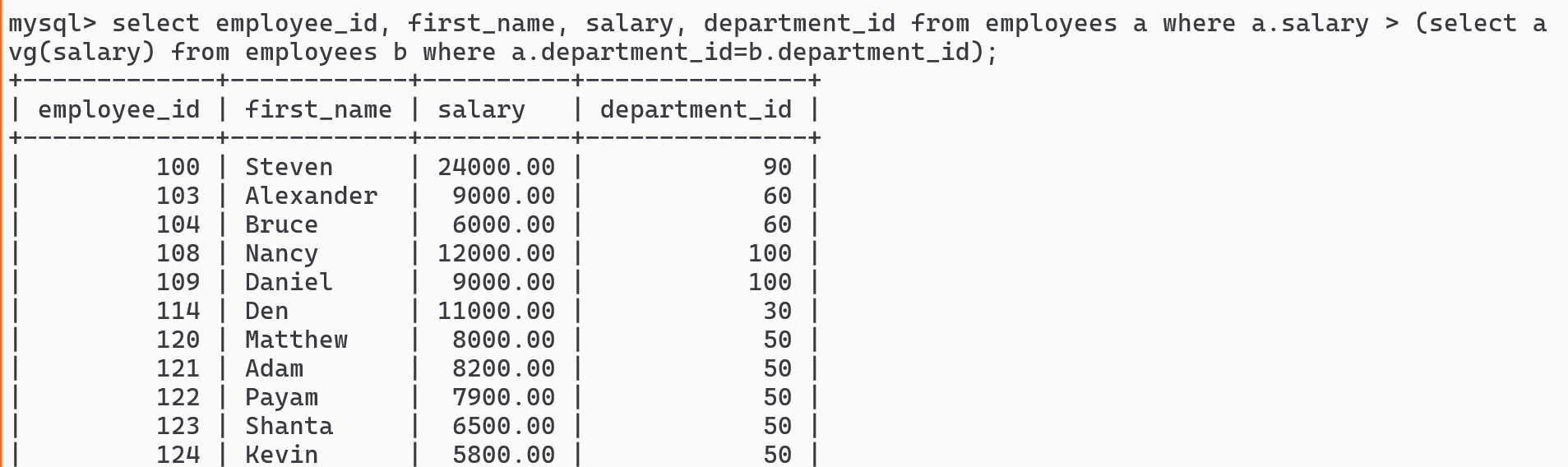
1. CREATE, ALTER, TRUNCATE, DROP, INSERT, UPDATE, DELETE, SELECT
2. GRANT & REVOKE
3. Single row functions & Aggregate functions
4. JOIN queries
5. Sub queries
6. Different clauses

Activity:

1. Find employees who are earning salary above their departments average salary
2. create students & marks table with roll\_no as a foreign key in marks table, student table will have roll\_no, name and age, whereas marks table will have marks\_id, roll\_no, physics, chemistry, maths, while displaying the result display student roll\_no, name, physics, chemistry, maths and the total marks of each student in the same row.

ex: if roll\_no 1234 has 50, 60, 70 marks then total marks will be 180

Solution for activity 1



IN vs ANY:

IN is to compare the value equal to the list exactly

like : 8 in (20, 8, 30)

ANY is used when you want to apply different comparisons like <, >

like: 8 > any (30, 40, 50) false

8 > any (5, 10, 30)

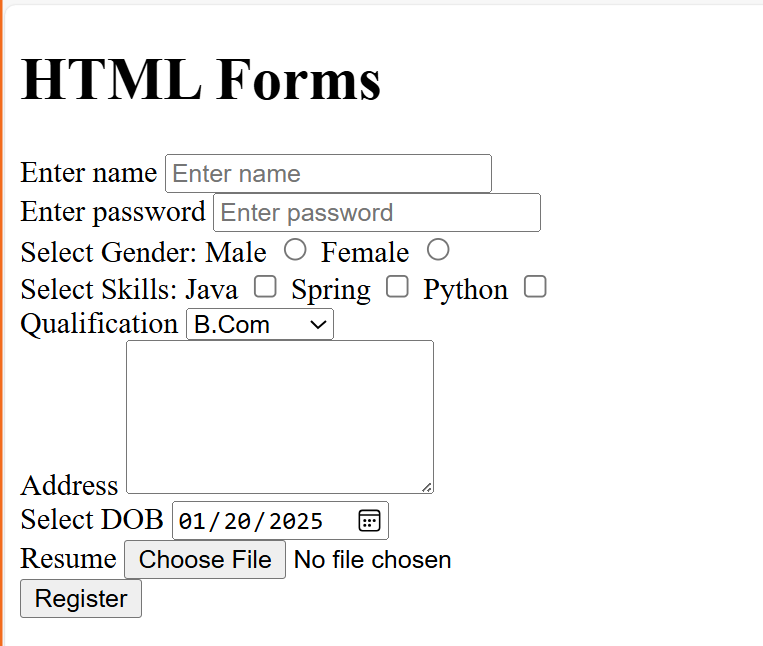
HTML

Hyper Text Markup Language: It is used to create web pages with HTML elements

A registration form that has controls like text ,password, checkbox, radio, drop down, date, text-area



Output



CSS

Cascading Style Sheet is used to apply styles to HTML document, it comes in 3 forms

1. inline: within the element
2. internal: within the html document
3. external: a reusable css file that applies styles to multiple html documents

CSS selectors

1. tag selector
2. id selector - an html element must have an unique id
3. class selector - multiple html elements can have same class

Inheritance in CSS

An element gets its style from its parent element, if the element has its own css then parent styles are overridden.

<div class = “danger”>  
 <h2>Some error message</h2>  
</div>

Bootstrap.js

It is a third party website that provides styles which you can use in your application using the built in styles.

To use bootstrap you must add its CDN link in your CSS or HTML file.

Javascript

It adds interactive effects to your web page by dynamically access HTML & CSS

Variables: They store values, in Javascript you can use let, const & var keywords, but you should avoid using var keyword because it doesn’t have a scope.

Functions: These are block of code that has names which you can reuse.

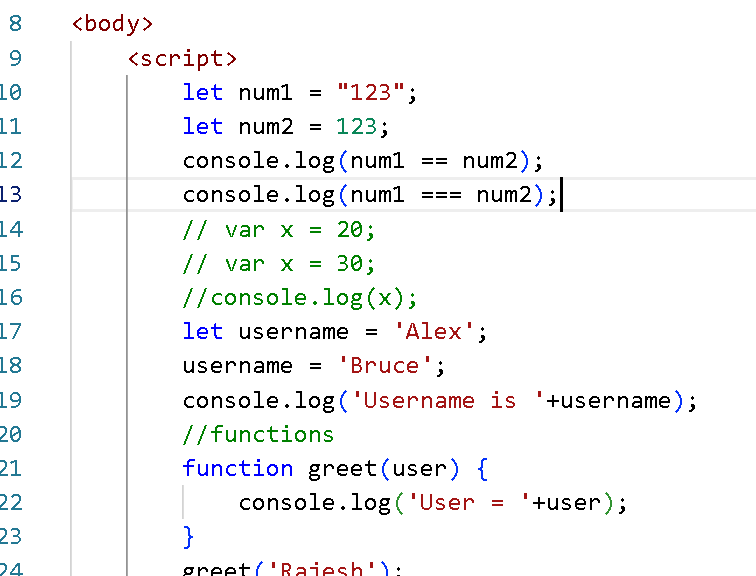
Functions & Variables



Operators

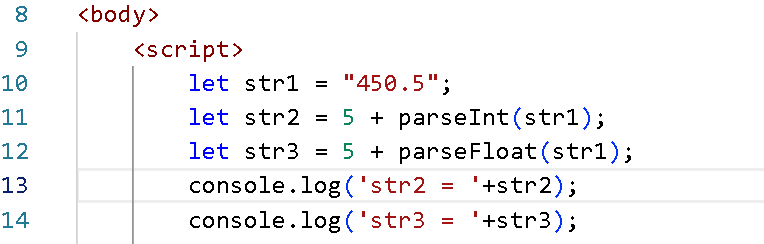
<, >, <=, >=, !=, =, ==, ===, &&, ||

== vs === : in == only value is compared, in === value & type both are compared.



Built in functions in javascript

parseInt() & parseFloat(): They are mainly used to convert numbers which are in string format.

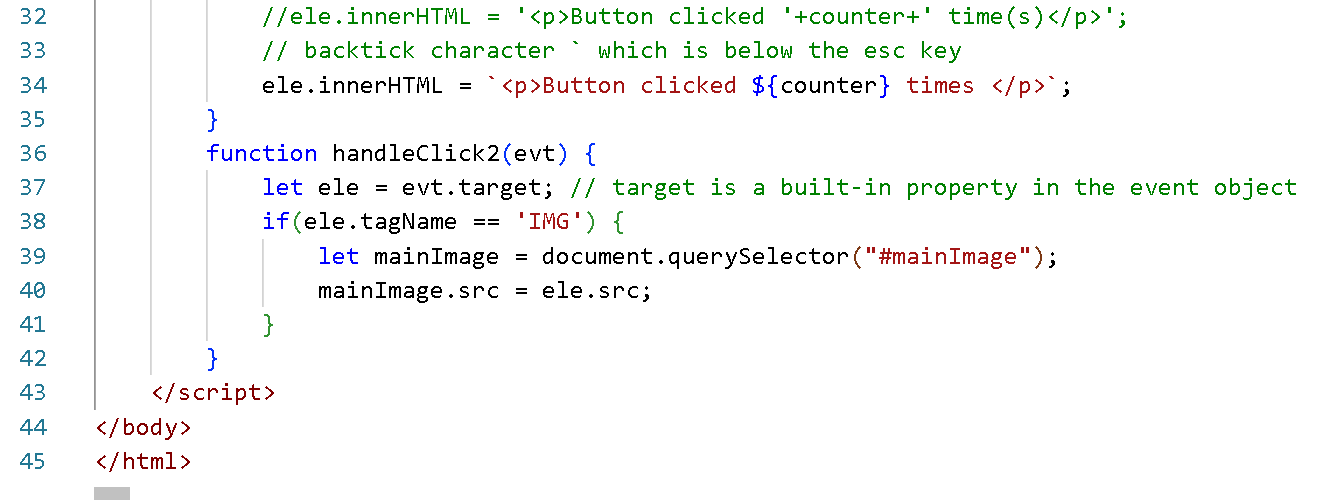


How to manipulate HTML elements using Javascript

Javascript provides a built-in object called document that provides various methods like

* getElementById(“id”)
* getElementsByTagName(“tagName”)
* getElementsByClassName(“className”)
* querySelector(#id | .class | tag)
* querySelectorAll(.class | tag)

element-demo.html



Fundamentals of javascript

1. variables
2. operators
3. conditional statements - if, if-else, if - else if .. else, switch
4. looping constructs - while, do-while, for
5. functions
6. objects

Modern syntax in javascript

It helps you to write the code in a simpler way, below are the features included in javascript

1. let, const, class, constructor, extends
2. Template strings : back tick
3. Arrow functions : simplifies writing single line anonymous functions
4. rest & spread operators
5. Promises
6. async/await

Arrow functions:

* Simplifies writing anonymous functions and event handlers
* You can use one line without { }
* If the body has more than one line then you can use { }
* return keyword is automatically considered if its one line, if you use { } you must use return keyword explicitly, this is necessary only if anonymous function had to return a value, in forEach() the anonymous function doesn’t return anything

ex:

(a, b) => a + b; // this is returning a + b value

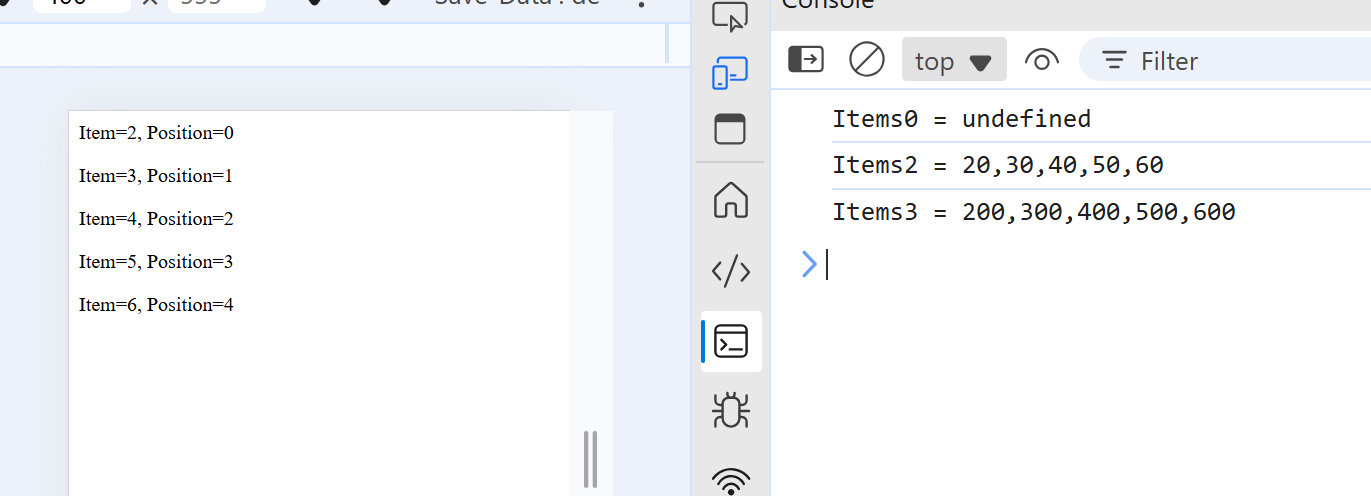
(a, b) => console.log(a + b); // this only prints a + b but returns nothing

(a, b) => { return (a + b); } // returns a + b, its necessary to use return because of { }

arrow function demo



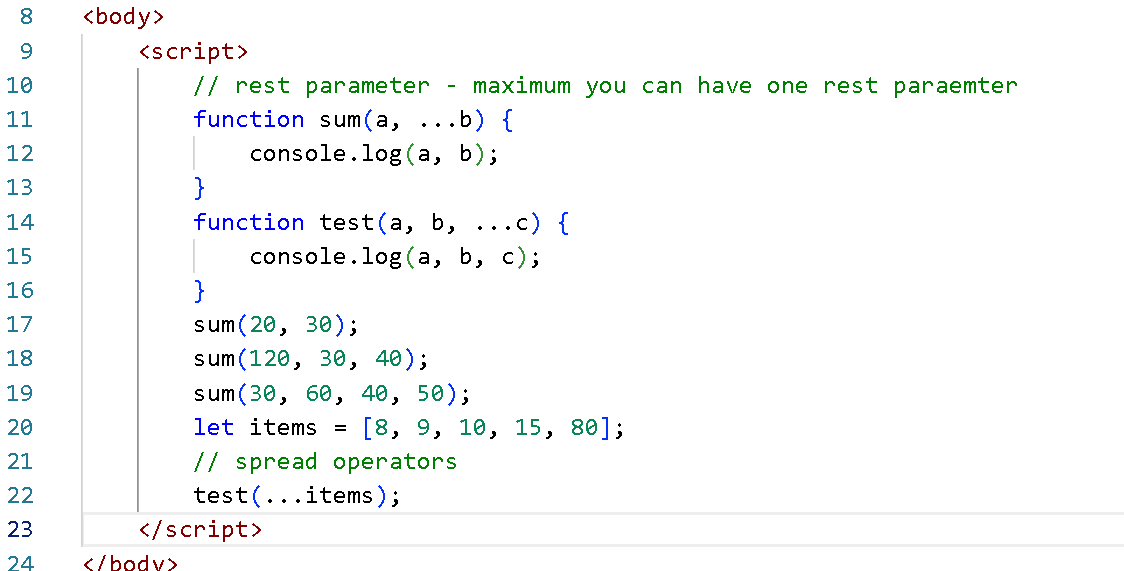
Output:



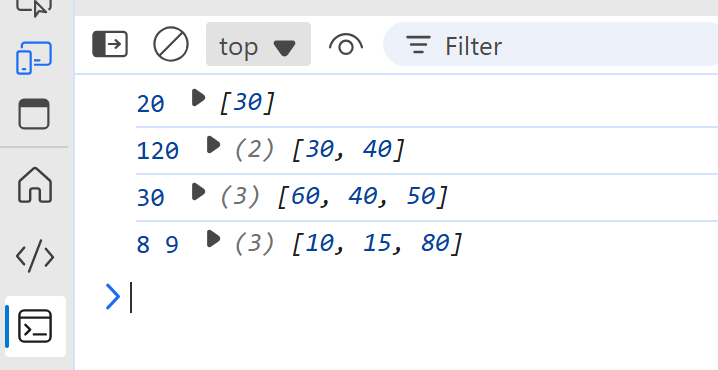
Rest & Spread

Rest: If you want a function to accept 0 or more arguments you can use this

Spread: If you want an array to distribute the values to multiple arguments then you can use this



Output:



Promises:

These are objects which an perform asynchronous operations, they can result 2 states while performing an async operations

1. resolved - promise is successful
2. rejected - promise is failed

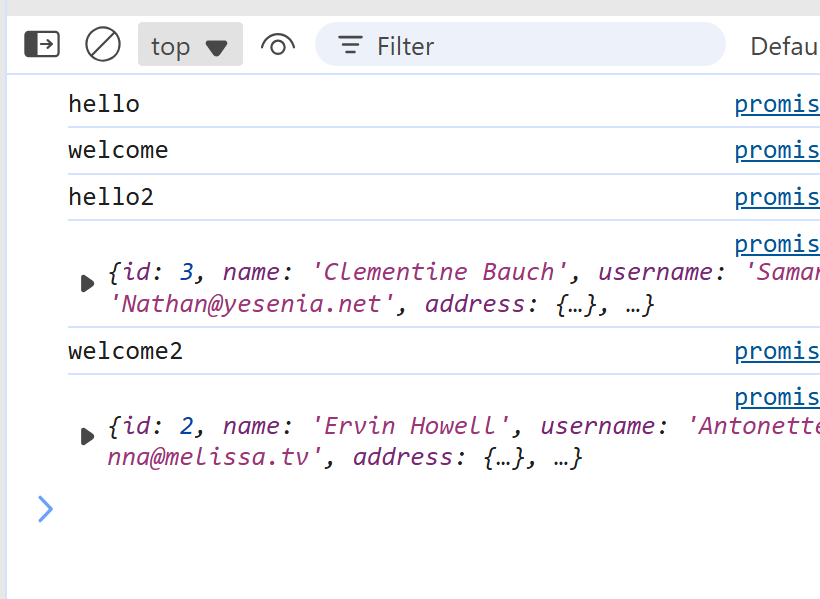
Promises handle these 2 states using .then() and .catch() functions which should chained, these functions take a callback function that is executed when the promise is settled (resolved or rejected)

example: fetch() is a built-in method in Javascript that can return a Promise by sending a request to the backend-server

fetch(server-url).then(callbackFn).catch(callbackFn);



Output:



Typescript vs Javascript

Typescript is more safe compare to javascript, it is a superset of javascript which adds types to the data and a function.

typescript code can be compiled to javascript, you can write all the codes in typescript and use it without touching the javascript code.

Javascript:

let username = “Ajay”;  
function sum(x, y) { … }

You can call sum by passing a number, string, boolean or objects

Typescript

let username : string = “Ajay”; // you can’t assign any other type other than string

function sum(x : number, y : number) { } // now you can only pass numbers to the sum function

Angular Framework

It is used to develop single page application using HTML and Typescript, it is mainly used to develop client side application i.e., front-end application.

Install Angular-CLI

It is a tool provided by Google to create, develop and build angular application

npm install -g @angular/cli

npm: It is a node package manager which you get from node.js

Command to create angular project

ng new app-name --no-standalone

Command to run the angular project

ng serve

Component class data can be accessed in its HTML template using {{ data }}

username = “Kishor”; // in TS

<div> {{ username }} </div> // in HTML

What happens when you hit ng serve

main.ts -> bootstrap(AppModule) -> App

Component: It is a reusable template or UI which you can independently create & use in other components.

Angular framework provides many commands to create many features in the application like components, services, pipes and etc.

ng generate component component-name [or] ng g c component-name

Activity

Create a new component and declare 2 properties like name and phone and display the name & phone in its template and use that component in the root component.

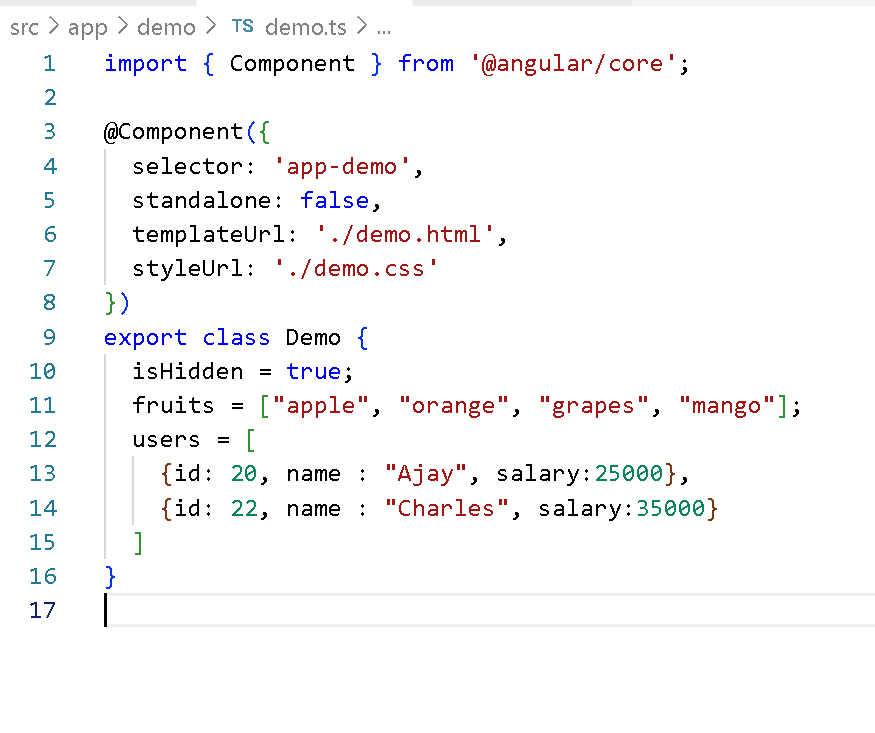
Angular directives

These are special features in angular that manipulates HTML DOM

1. Component directive - @Component that helps to structure the HTML element using the custom tag
2. Structural directive - \*ngFor, \*ngIf, NgSwitch, which modifies HTML DOM elements at runtime.

\*ngIf & \*ngFor

demo.ts



demo.html



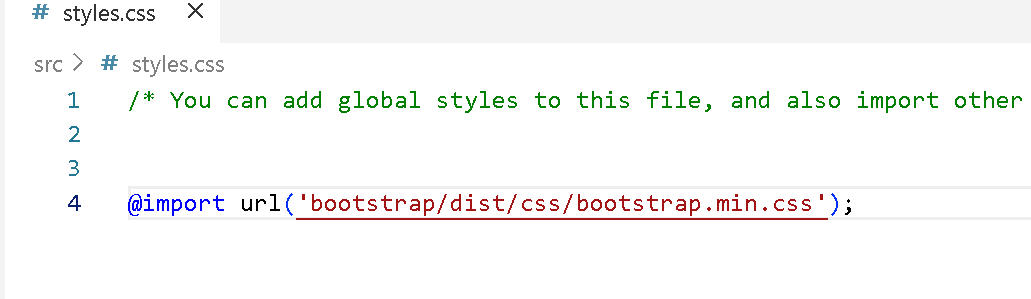
Output:



Adding bootstrap to the project

1. use npm i bootstrap
2. Add the path of the bootstrap.min.css in the styles.css

styles.css



Data Binding

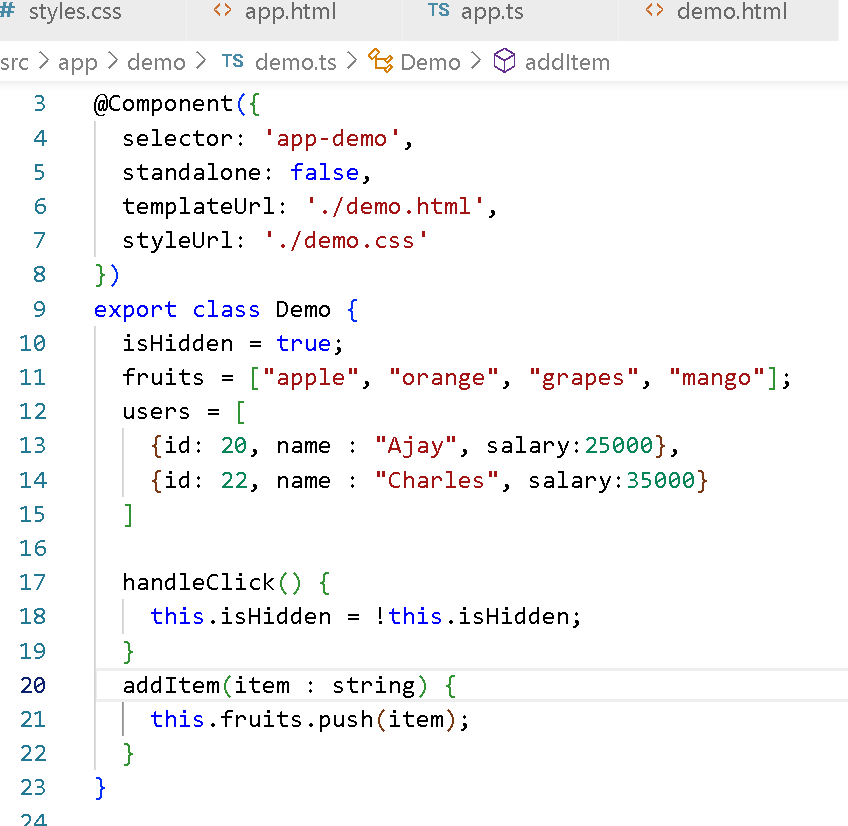
This helps to share the data between components & the HTML template.

1. Interpolation : {{ }} is used to share the data from component class to HTML template, it is mainly used to display the content
2. Property Binding : [ ] is used to share the data from component class to HTML element, this is mainly used to add / remove DOM properties
3. Event Binding : ( ) is used to share the data from HTML to component class, it is mainly used to share data from user to the application, like login, registration details
4. Two way data Binding: [( )] is used to share the data in both the ways, i.e, HTML to component and vice versa, mainly used to sync the UI and the application data, like validating forms

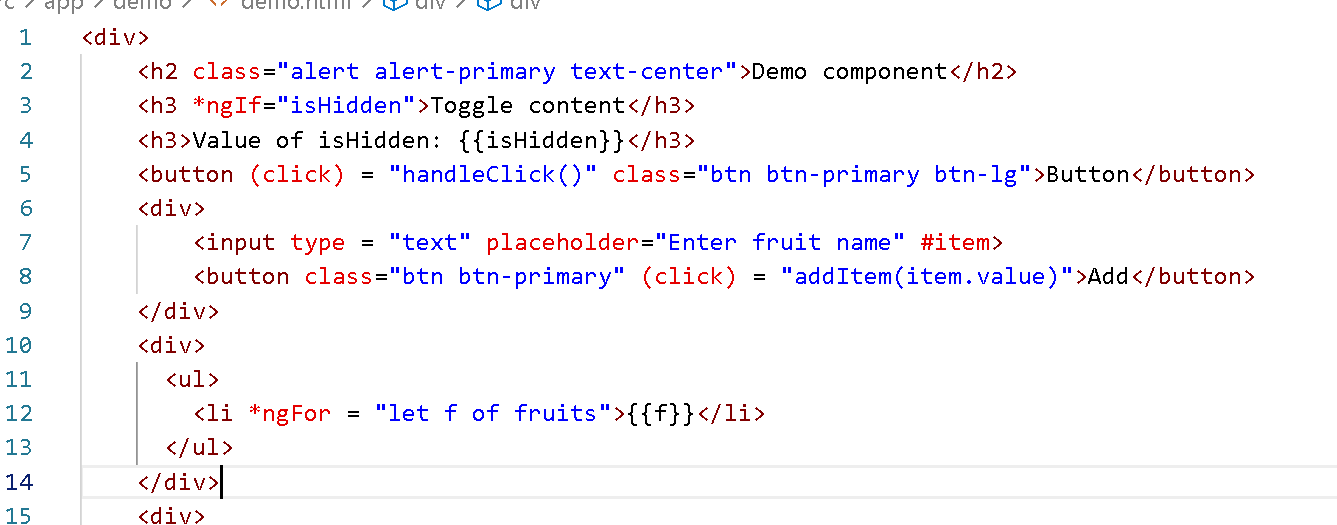
Event binding: In angular you can handle events with the name like click, ngSubmit, input, change and so on.

<button (click) = “handleClick()”>Button</button>

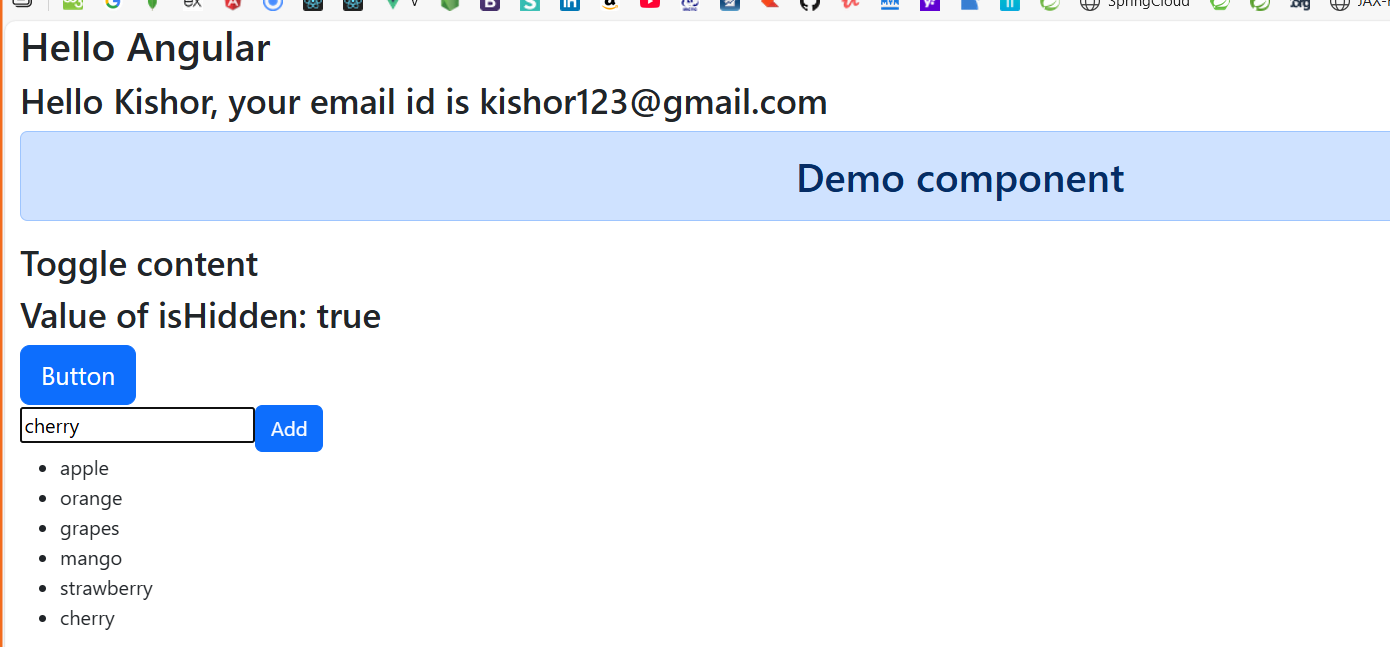
ts



html



Output:



Revise

* creating a new project
* installing bootstrap & importing
* creating a new component and using it in the root component
* running the application

Property Binding: This allows data to be shared from component class to an HTML template, it helps to modify HTML DOM properties like hidden, disabled

In HTML

<button [disabled] = “isDisabled”>Button</button>

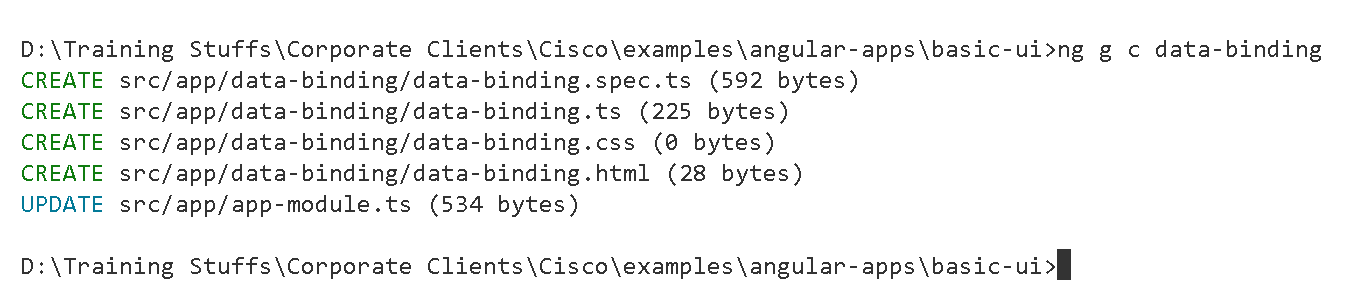
<div [hidden] = “isHidden”>Some content</div>

In TS

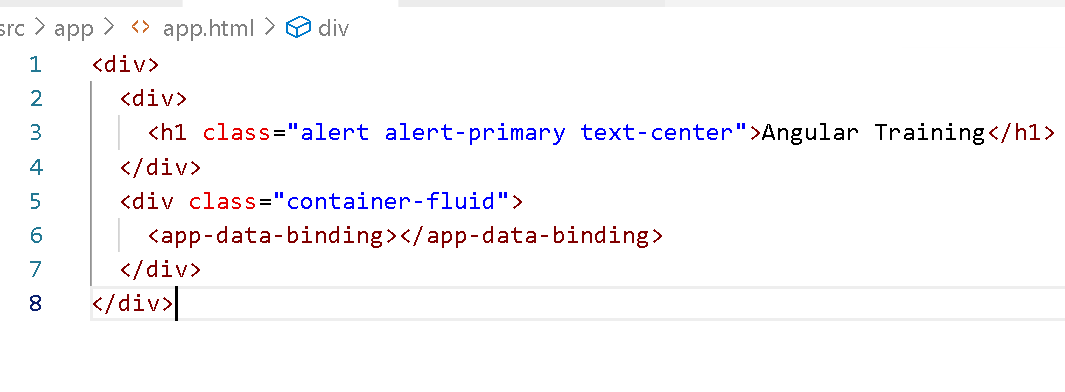
isDisabled = true or false;

isHidden = true or false

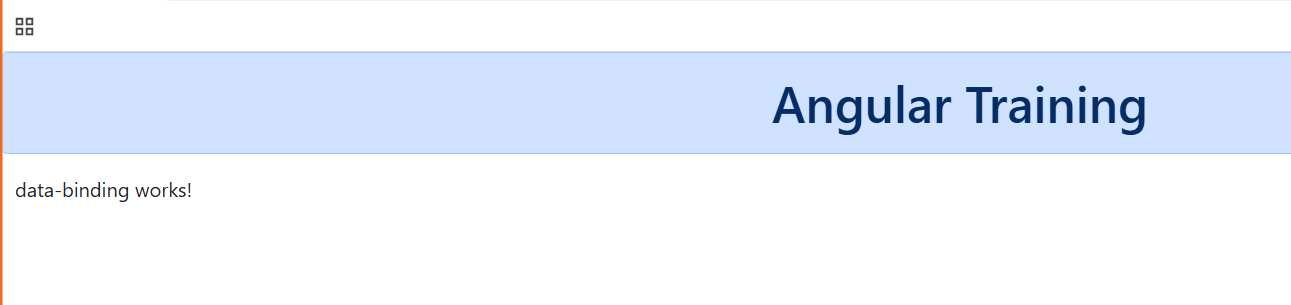
Generate a new component data-binding



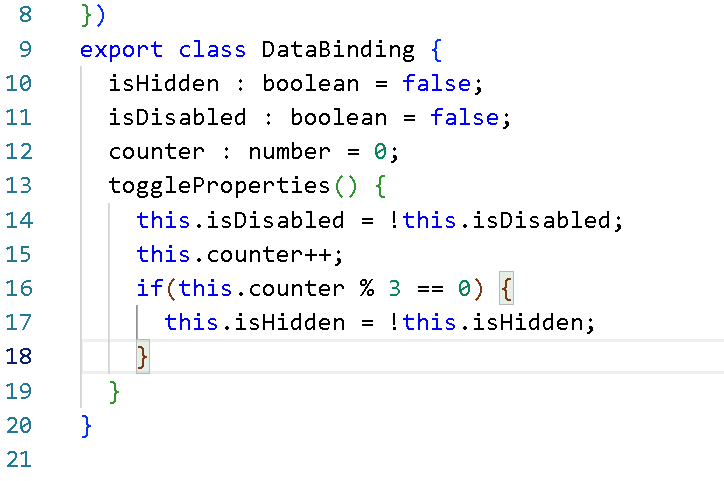
Add this component to the root component before developing anything in this



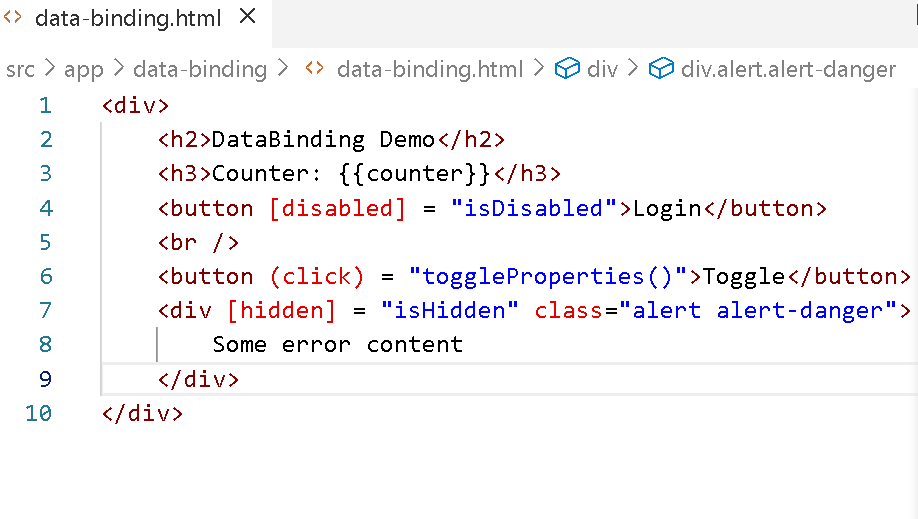
Output:



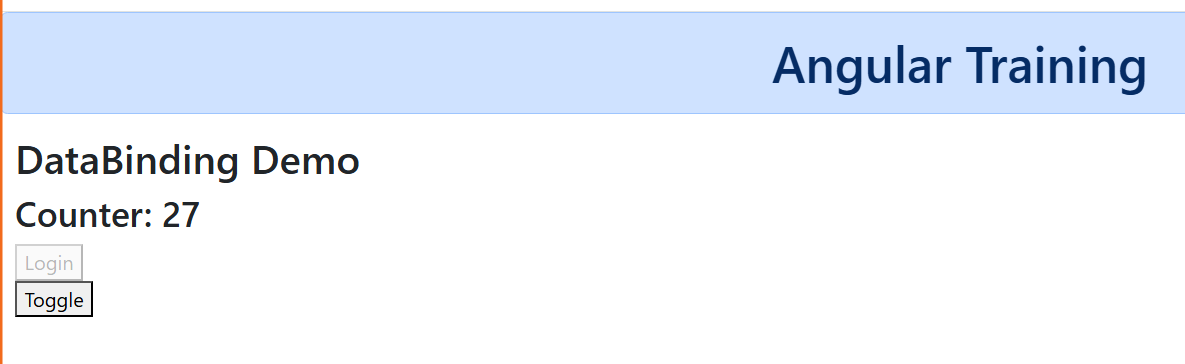
Modify data-binding.ts file



Use the above properties in data-binding.html



Output:



Two way databinding

This enables to share data in both the ways from component to template and vice versa, you must use [(ngModel)] attribute in the HTML element, the [(ngModel)] is by default not available hence you must import the FormsModule in the app.module.ts

app-module.ts

@NgModule({  
 imports : [FormsModule]  
})

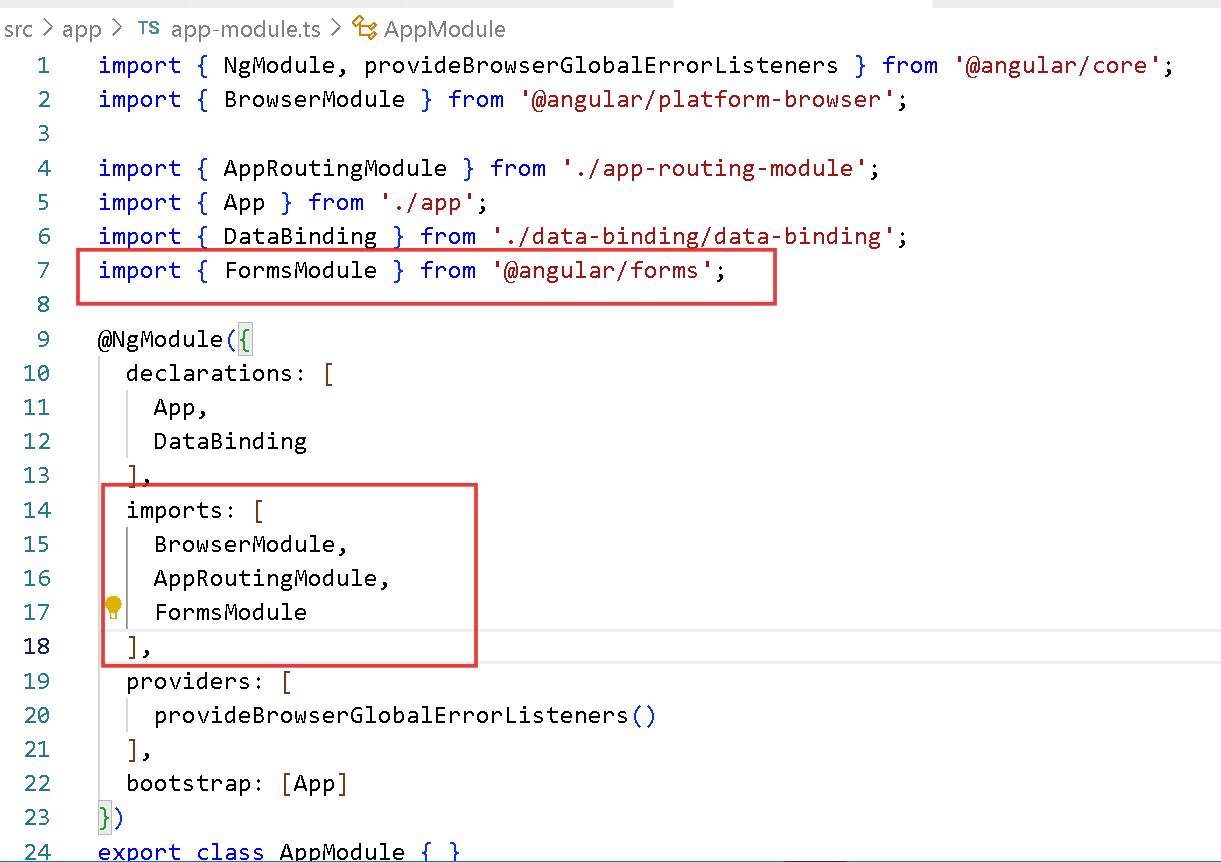
data-binding.ts

username : string = “Guest”;

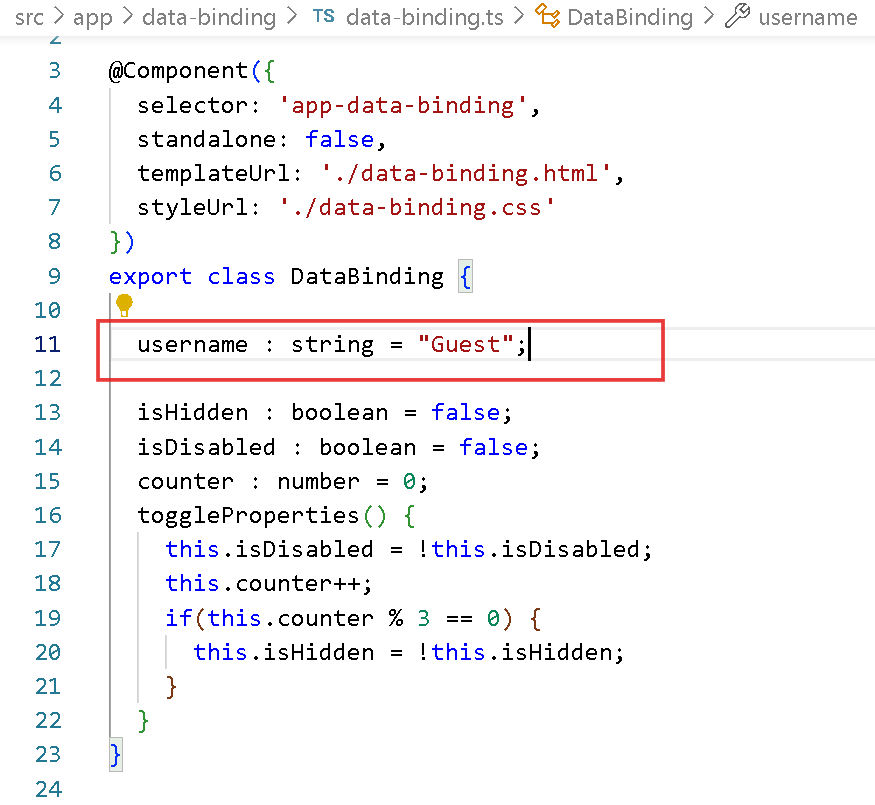
data-bindint.html

<input type = “text” [(ngModel)] = “username”>

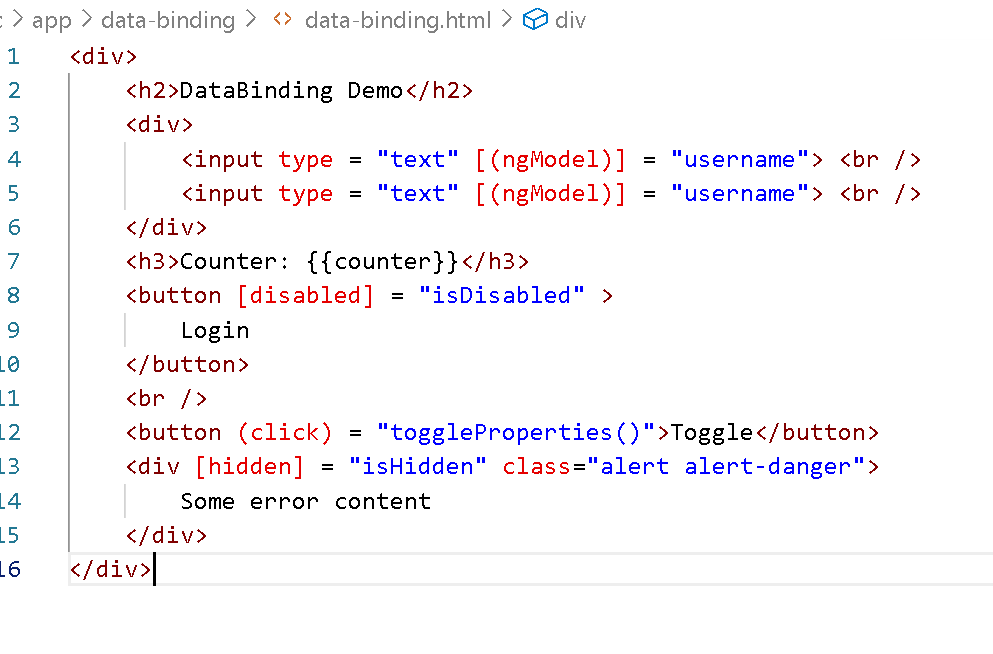
app-module.ts



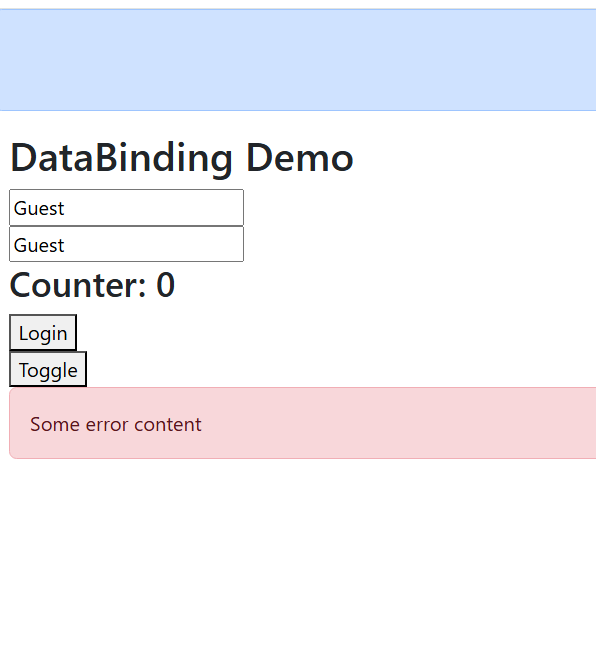
data-binding.ts



data-binding.html



Output:



Style & Class Binding

Style binding helps to add inline styles to an element at runtime

<div [style] = “myStyles”> in HTML

myStyles = {“color”:”red”; “border”: “2px solid black”} in TS

Class binding helps to add or remove css classes for an element at runtime

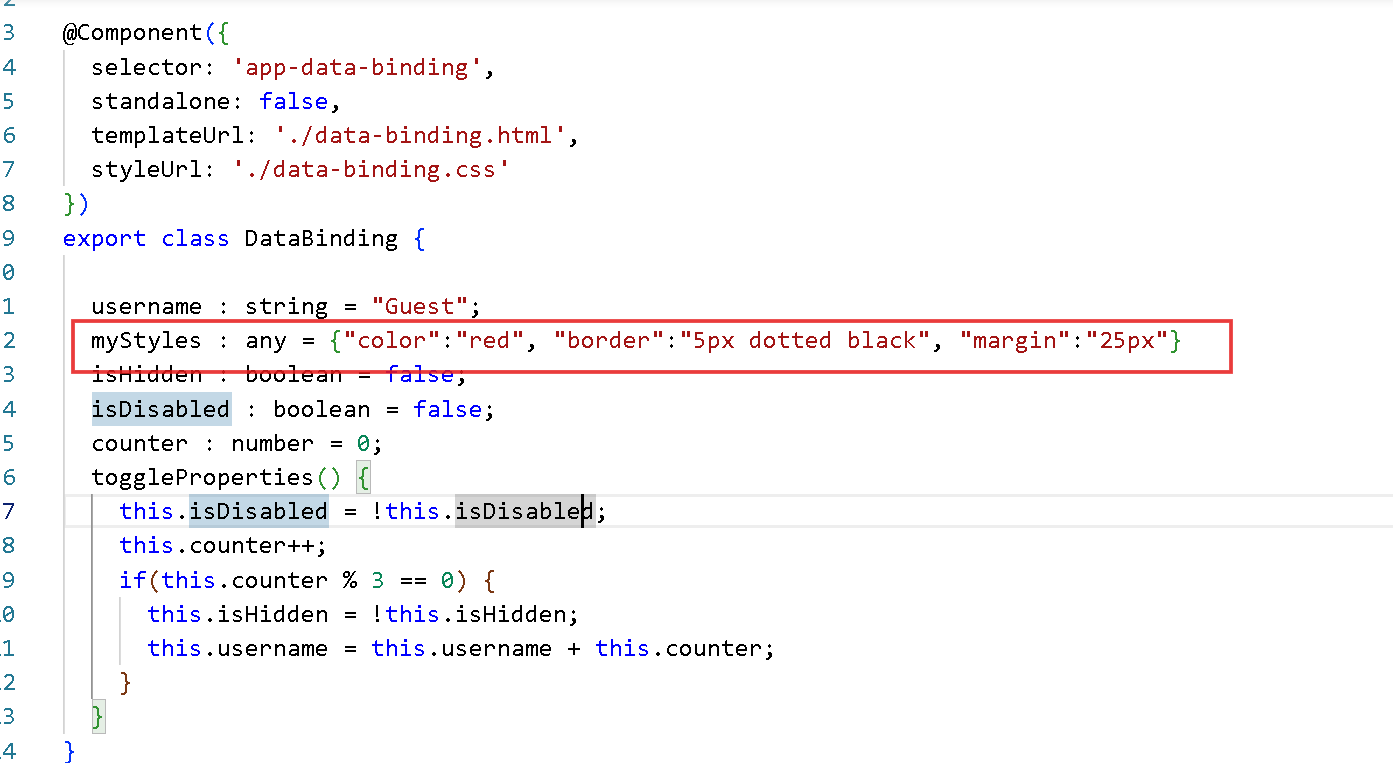
**.blueColor** { color : blue } in CSS

<div [class.blueColor] = “addColor”>Some content</div> in HTML

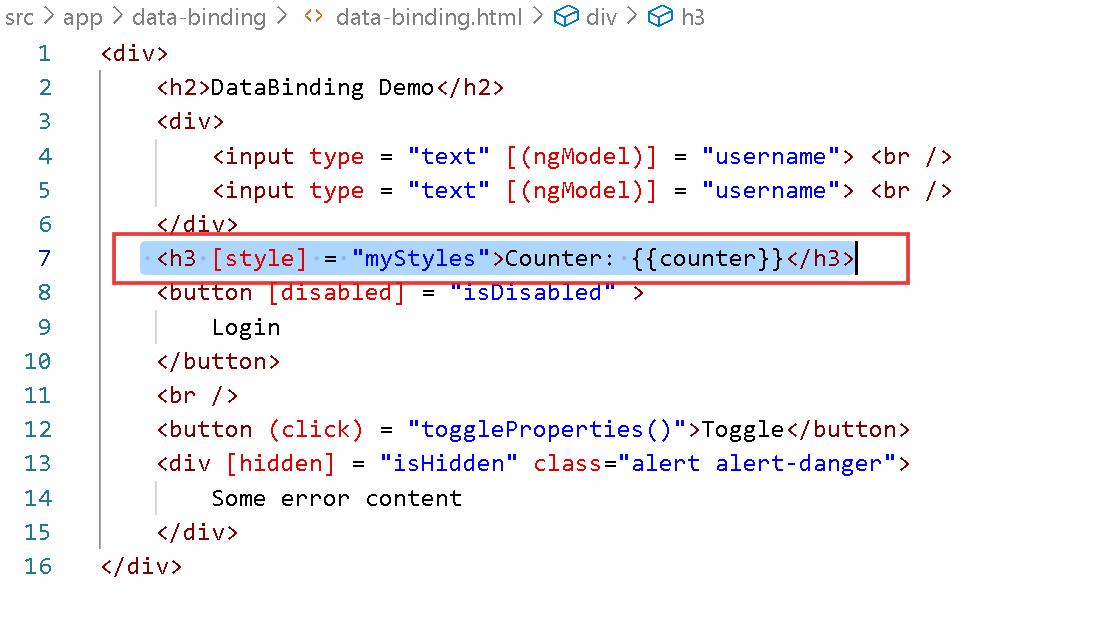
addColor = true or false in TS -> change the value from true to false and vice versa

Style binding

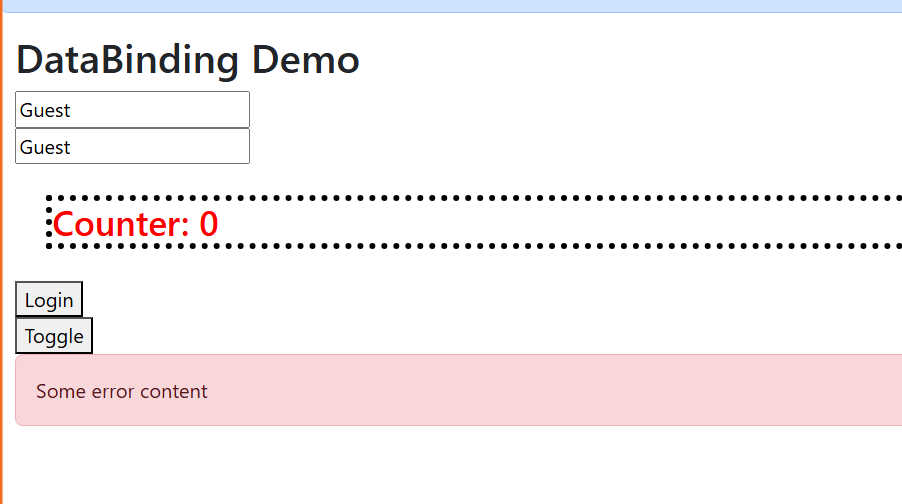
TS



HTML

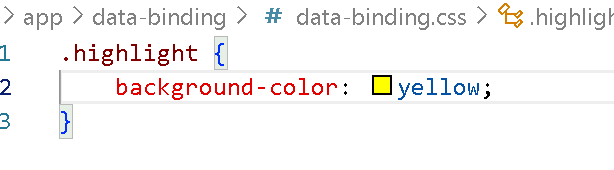


Output:

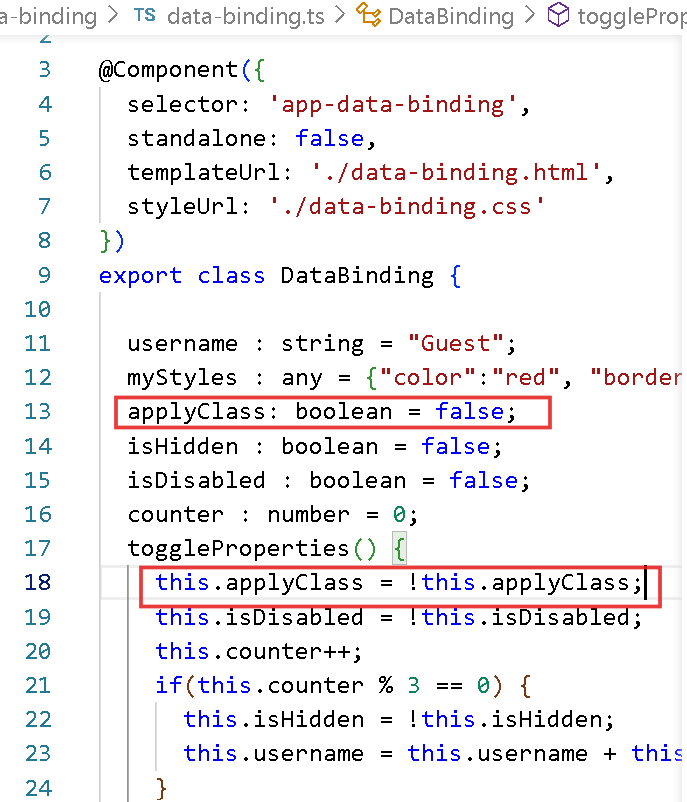


Class binding

data-binding.css



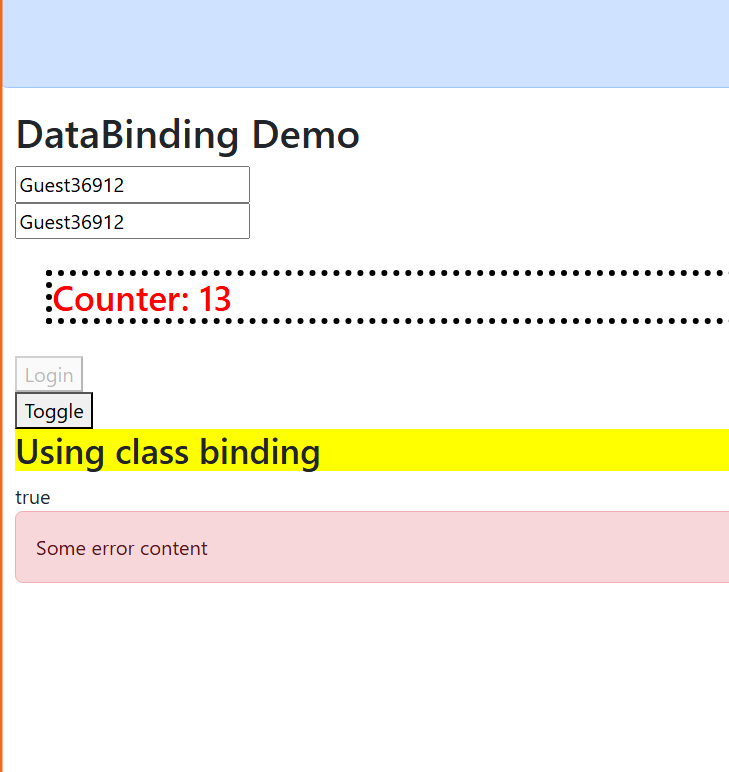
data-binding.ts



data-binding.html



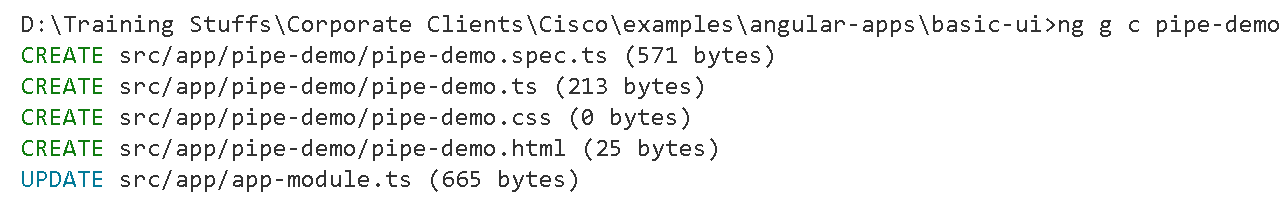
Output:



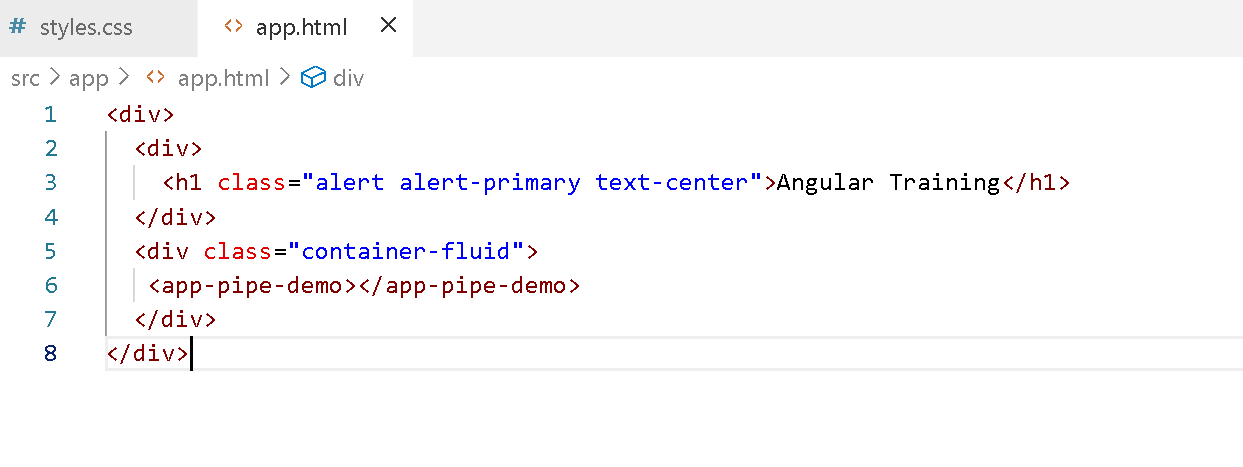
Pipes

These are special features that helps to transform the output, you get many built-in pipes in angular like upper, lower, currency, date, json, you can also create your own pipes (custom pipes) for your project (ng g p pipe-name)

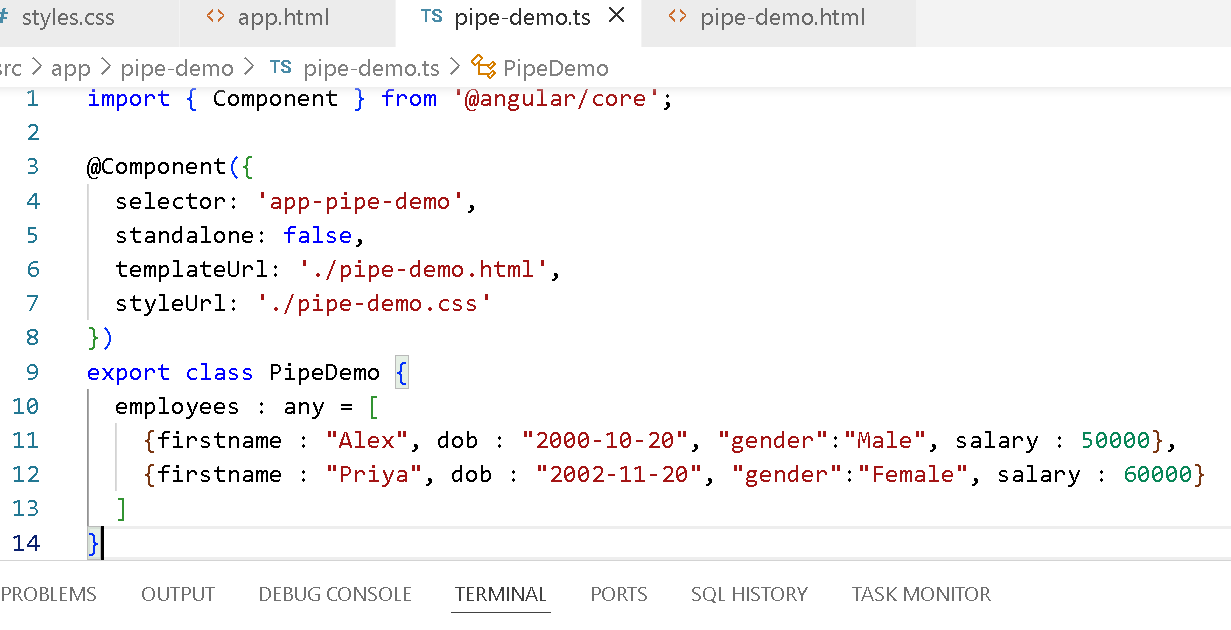
Generate a component



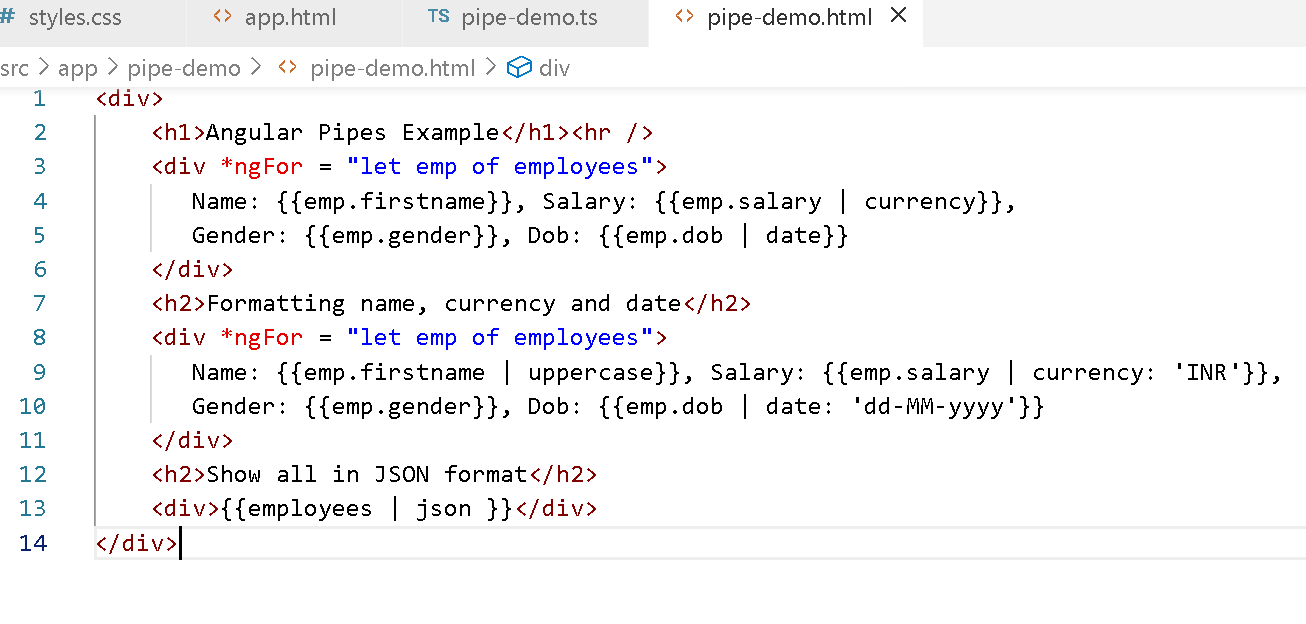
Using it in root component



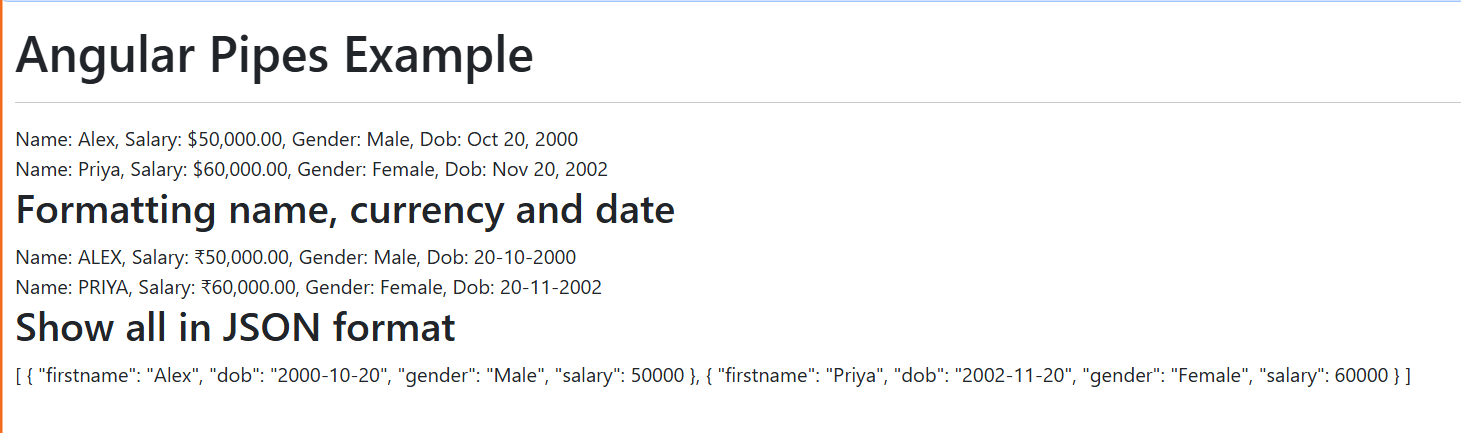
pipe-demo.ts



pipe-demo.html



Output:

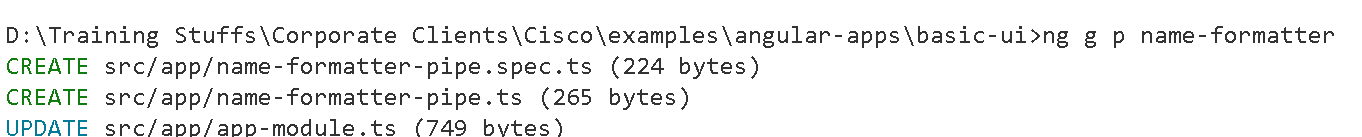


Custom pipes

You must use a class with @Pipe decorator that tells what is the name of the pipe and also you must implement a PipeTransform that provides a transform method that gets called when you use the pipe name

@Pipe({name: ‘someName’})  
class NameFormatter implements PipeTransform {   
 transform(value : any, args : any) {   
 return transformedValue;  
 } }

In angular you can use ng g p pipe-name to get the class with @Pipe and also it updates the app-module.ts file



We will transform the name to have a prefix like Mr or Ms before the name based on the gender

i.e.,

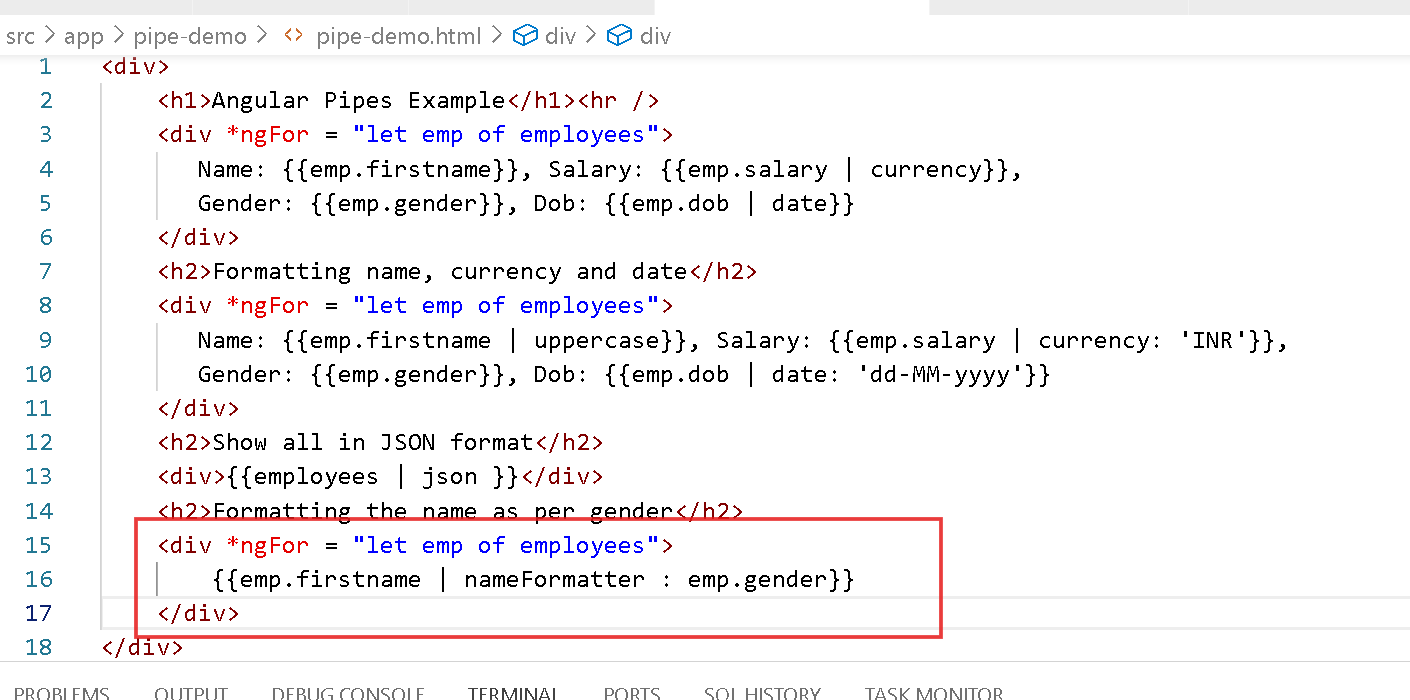
{{ emp.firstname | nameFormatter : emp.gender }}

name-formatter-pipe.ts

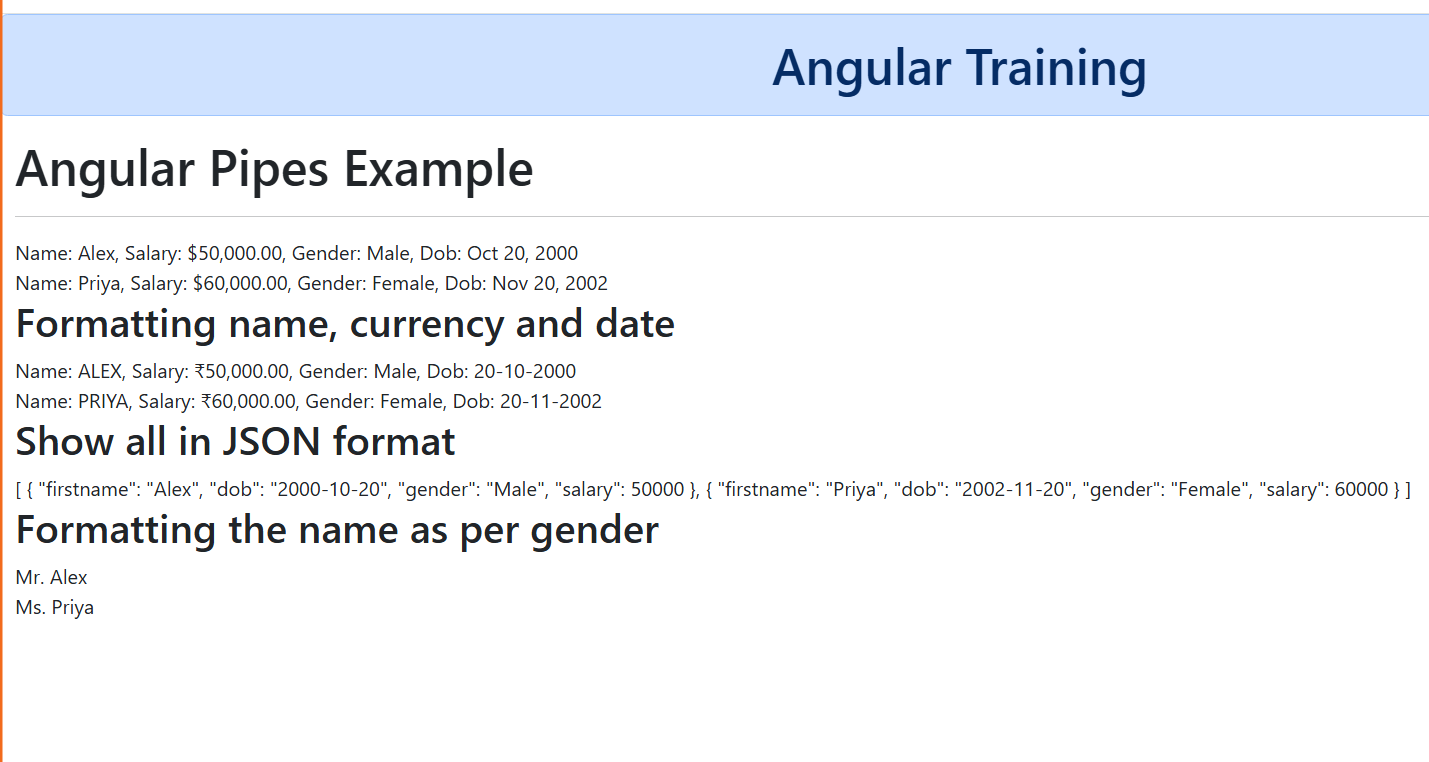


Note: args[0]?.charAt(0) means if args is undefined, then it doesn’t access charAt(0)

pipe-demo.html



Output:

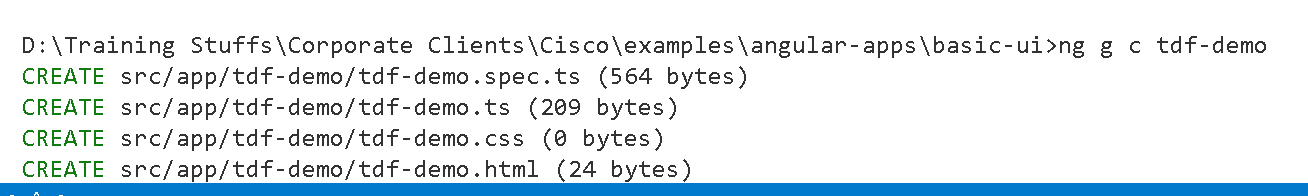


Angular Forms

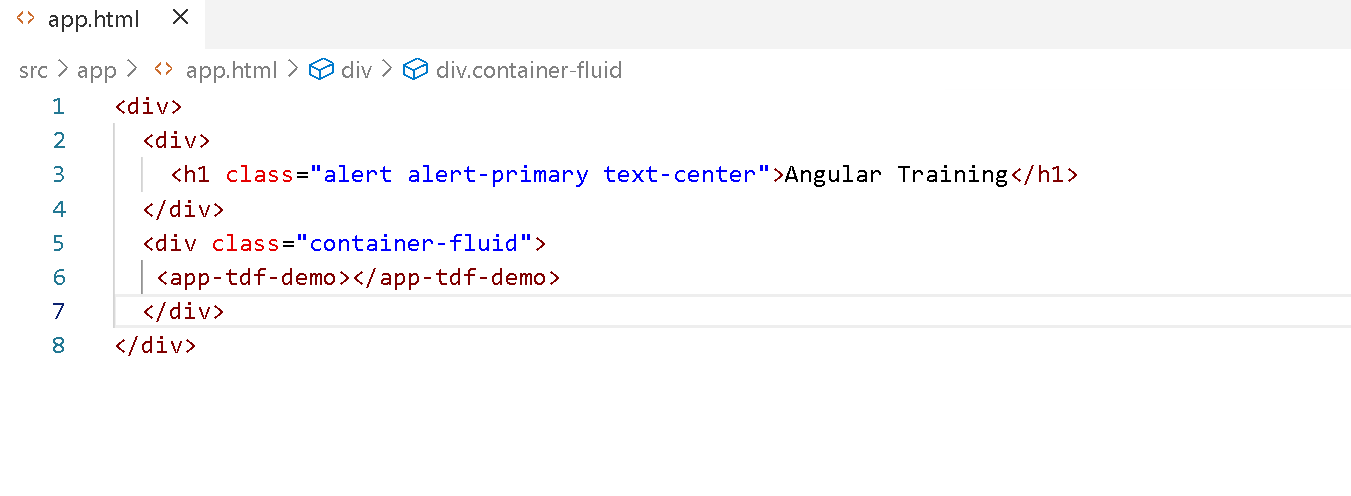
Forms are the building blocks in UI that allows user to enter the data which you can submit to the application, in angular you can create forms in 2 ways.

1. Template driven form - data can be controlled in HTML template, this is better for simple forms, it uses [(ngModel)] to control the data
2. Reactive form or Model driven form - data can be controlled in TS file, this is better for complex logics and forms, it uses form group service to control the data.

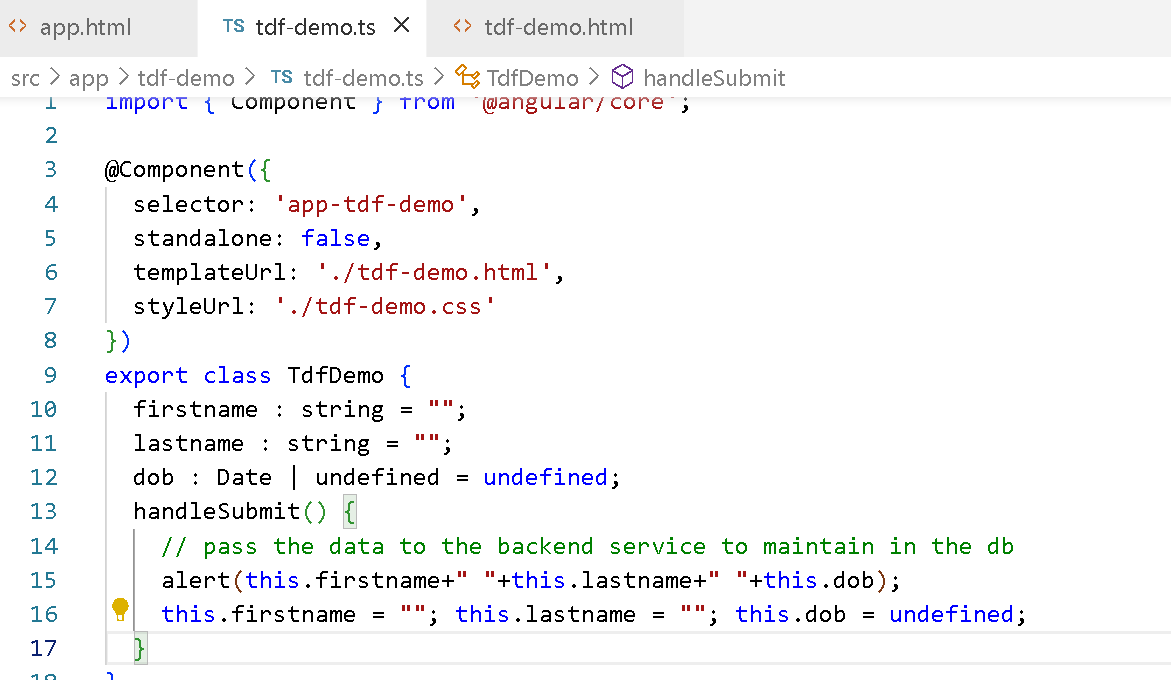
Template driven form



app.html



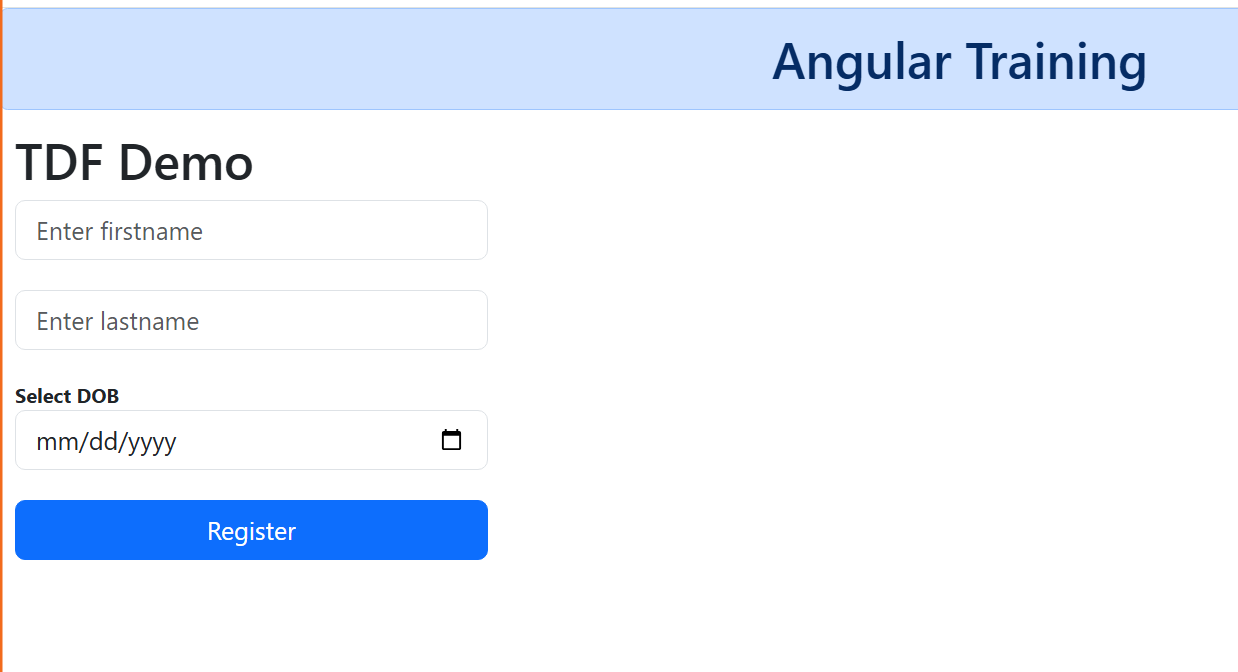
tdf-demo.ts



tdf-demo.html



Output:



Reactive forms: It helps to create forms and control in the typescript code, you will get more control over the data when form controls are created using typescript compare to HTML.

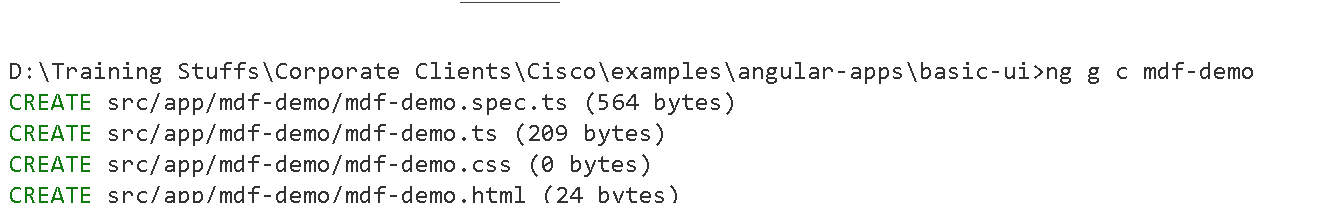
Reactive forms use ReactiveFormsModule that provides a service called FormBuilder that provides a function called group() inside which you can create form-controls.

Note: You must import ReactiveFormsModule in the app-module.ts, then you can get the FormBuilder object in the component via DependencyInjection done with inject().

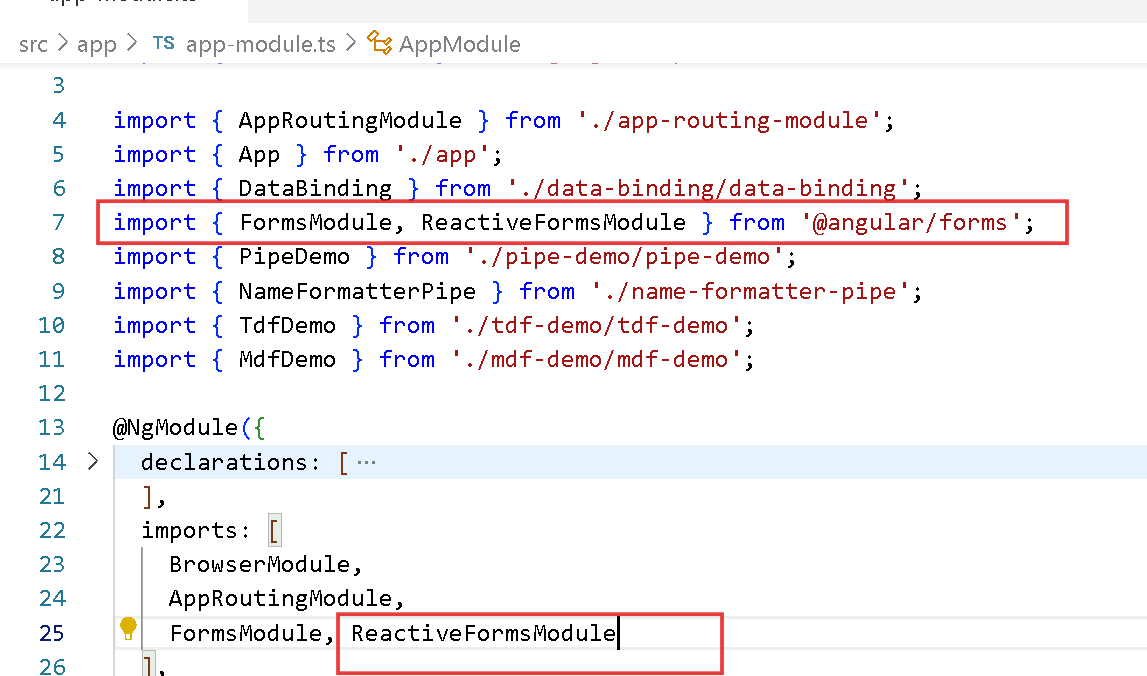
builder = inject(FormBuilder);

Angular supplies the object of FormBuilder to this class, the reason you need to use private is you can use that variable in other methods of the class

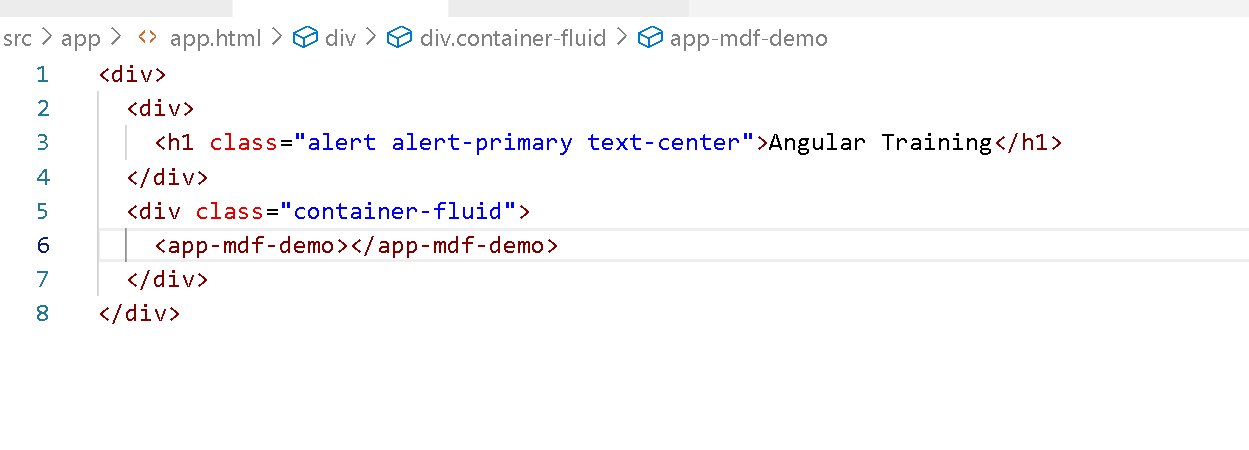
Generate the component



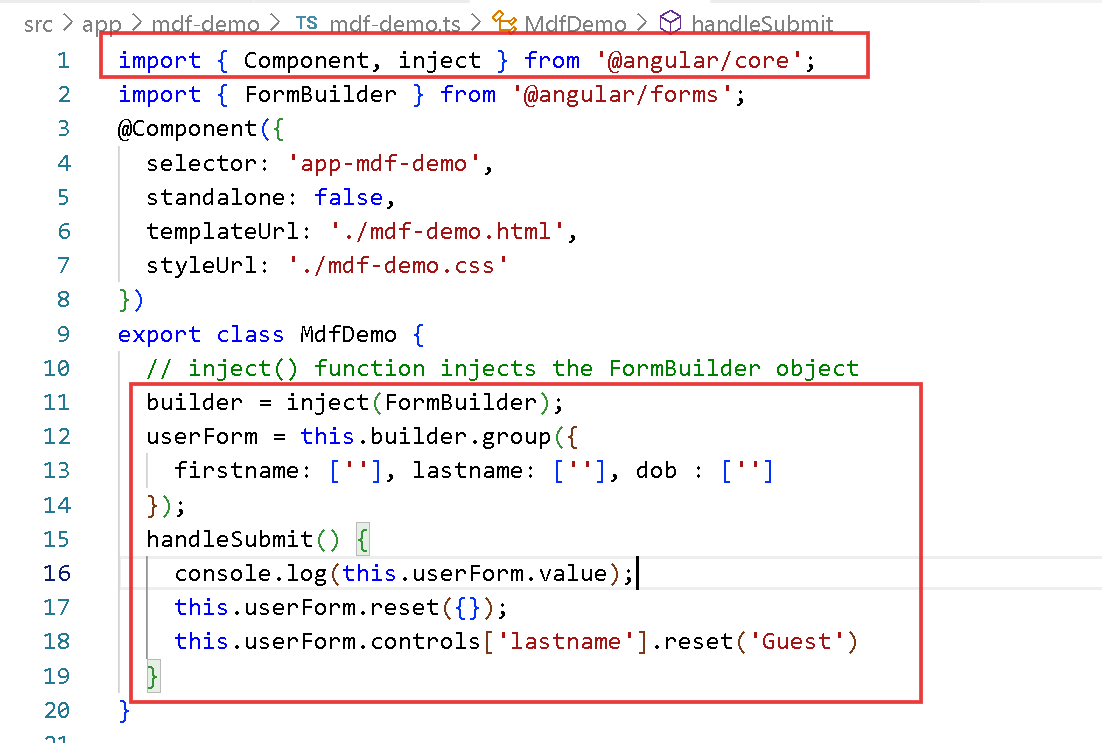
Add ReactiveFormModule in the imports of @NgModule



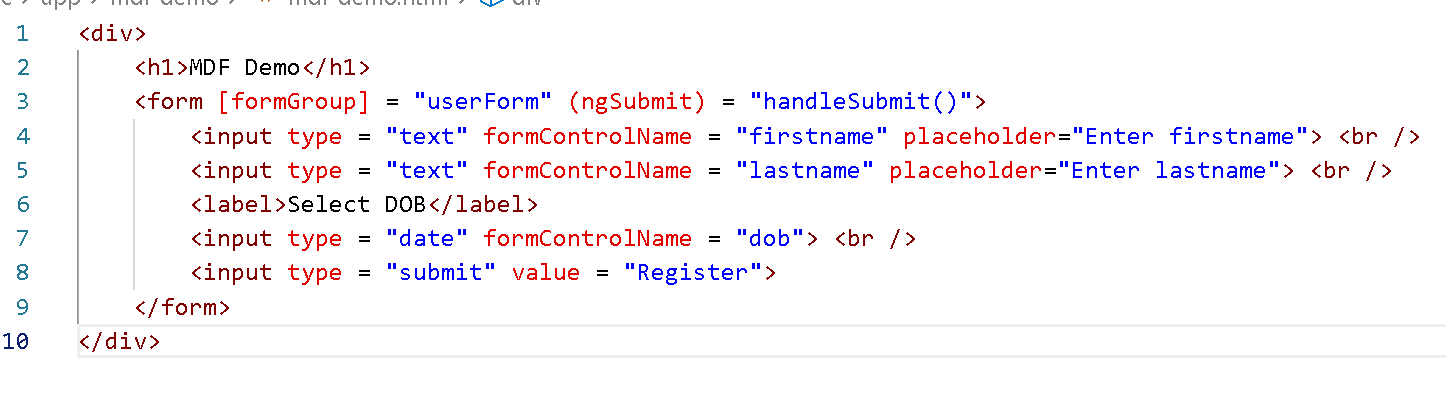
Add the mdf-demo to the root component



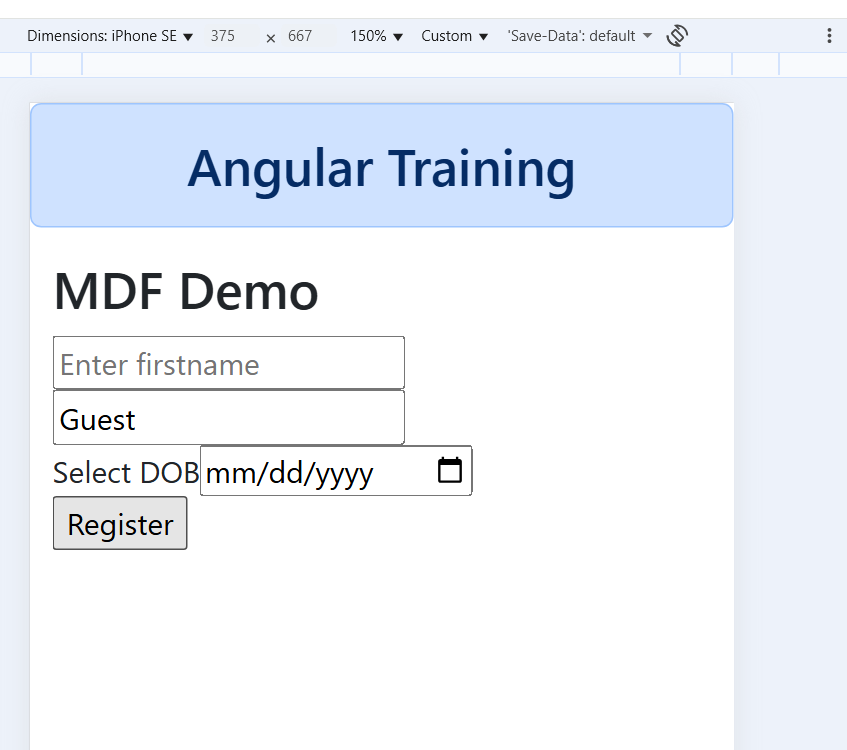
mdf-demo.ts



mdf-demo.html



Output:



Form Validation:

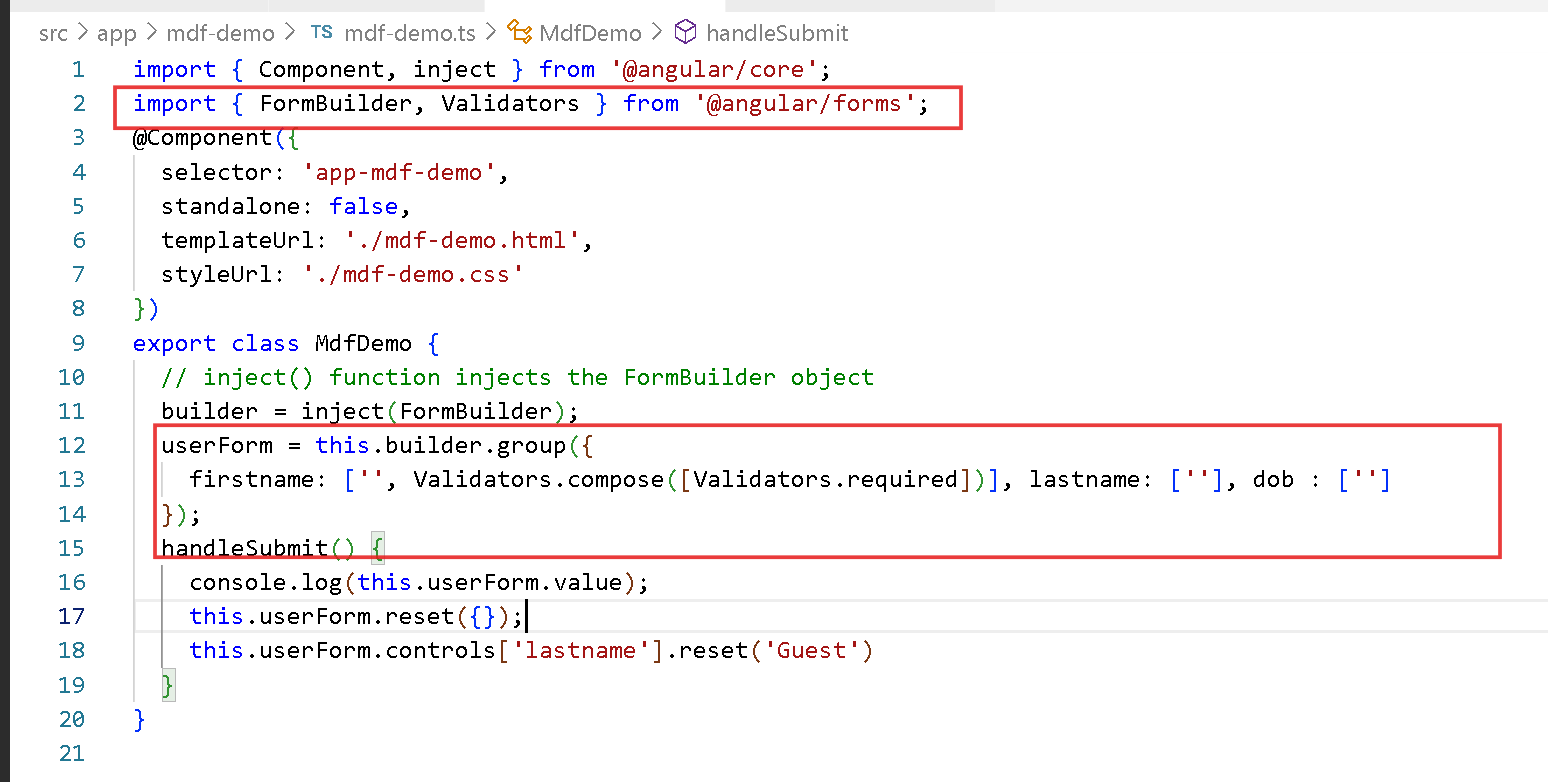
Sometimes you don’t want an empty form to be submitted or a data that doesn’t match to certain conditions to be submitted, in that case you can validate and submit the forms.

In TDF you need to write validators in HTML template like maxLength, required and so on.

In MDF you need to write validators in TS file having form group like

userForm = this.builder.group({firstname: [‘’, Validators.compose(Validators.required, …)]});

mdf-demo.ts



mdf-demo.html



Output:

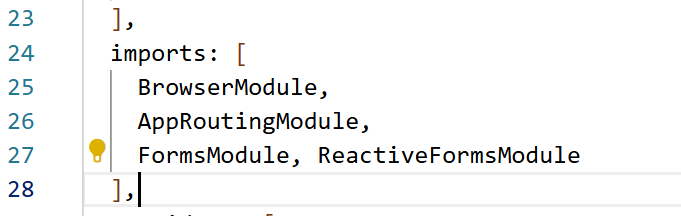


Activity:

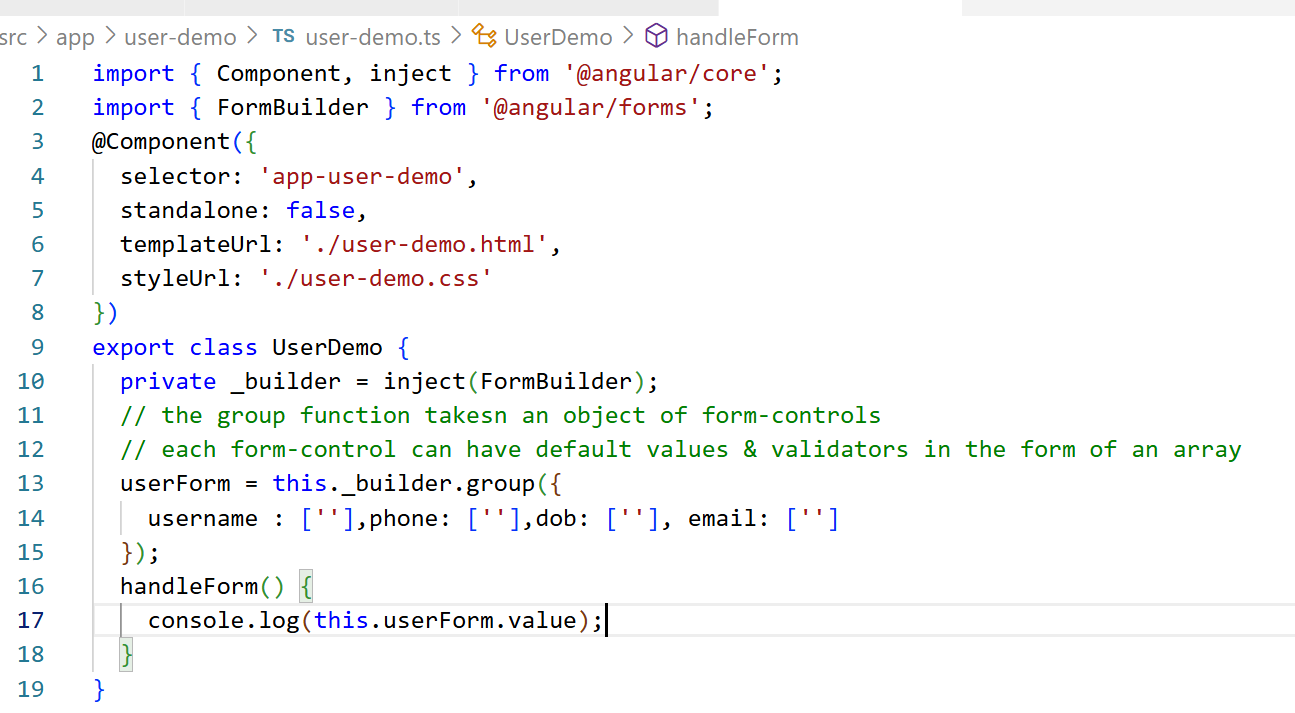
Create validators for lastname and display the error message if the last name is invalid

Understanding Model Driven Form using User Registration from

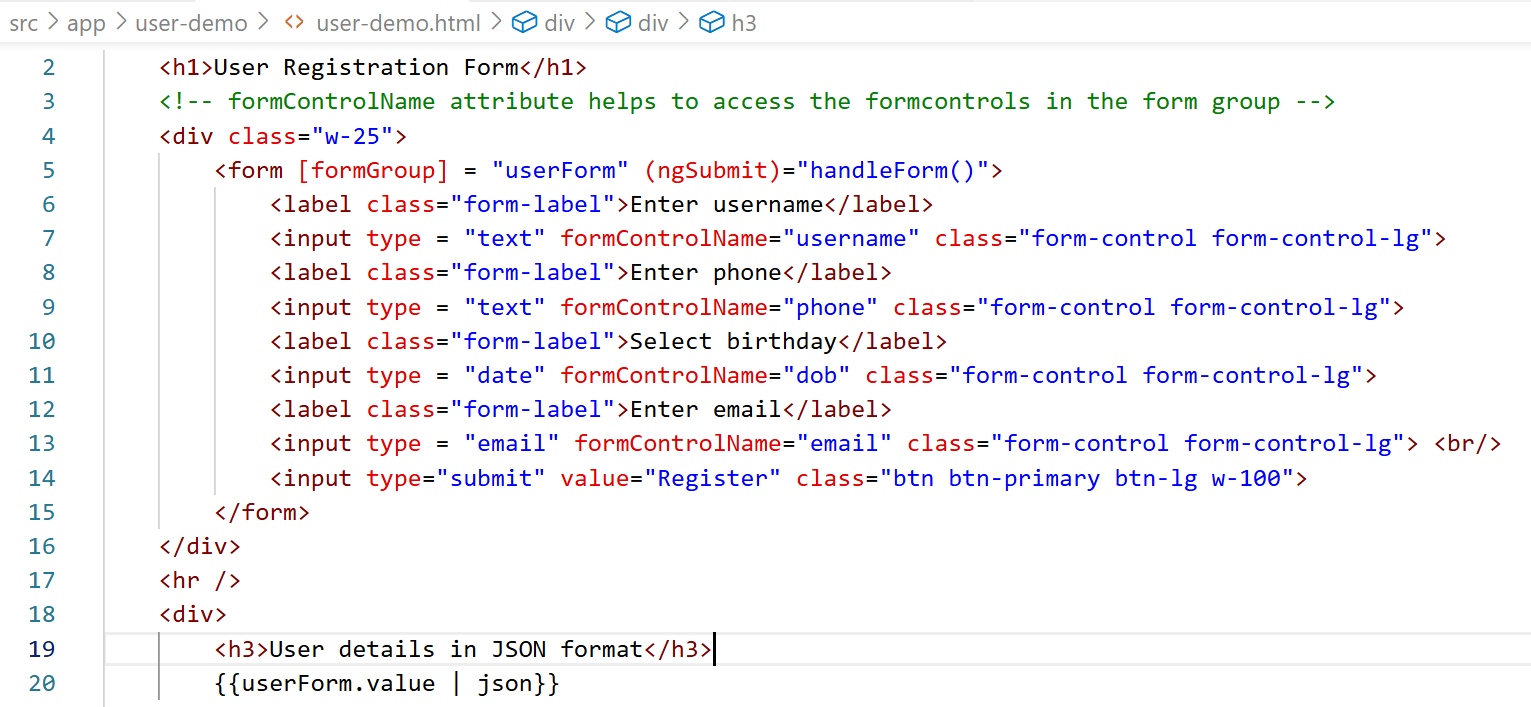
Ensure that app-module.ts file has ReactiveFormsModule imported



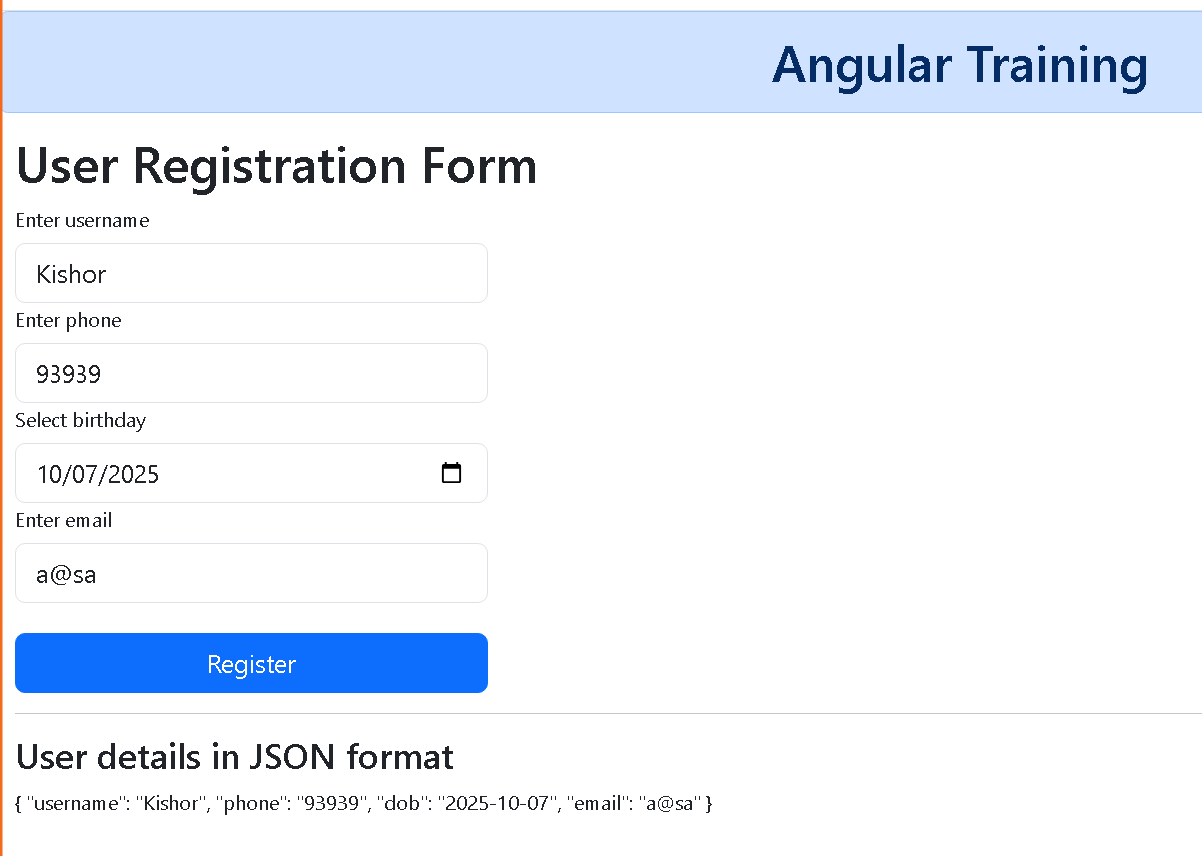
user-demo.ts



user-demo.html



Output:



Validation:

In the form-control you can directly pass array of validators or use Validators.compose() to that you can pass array of validators

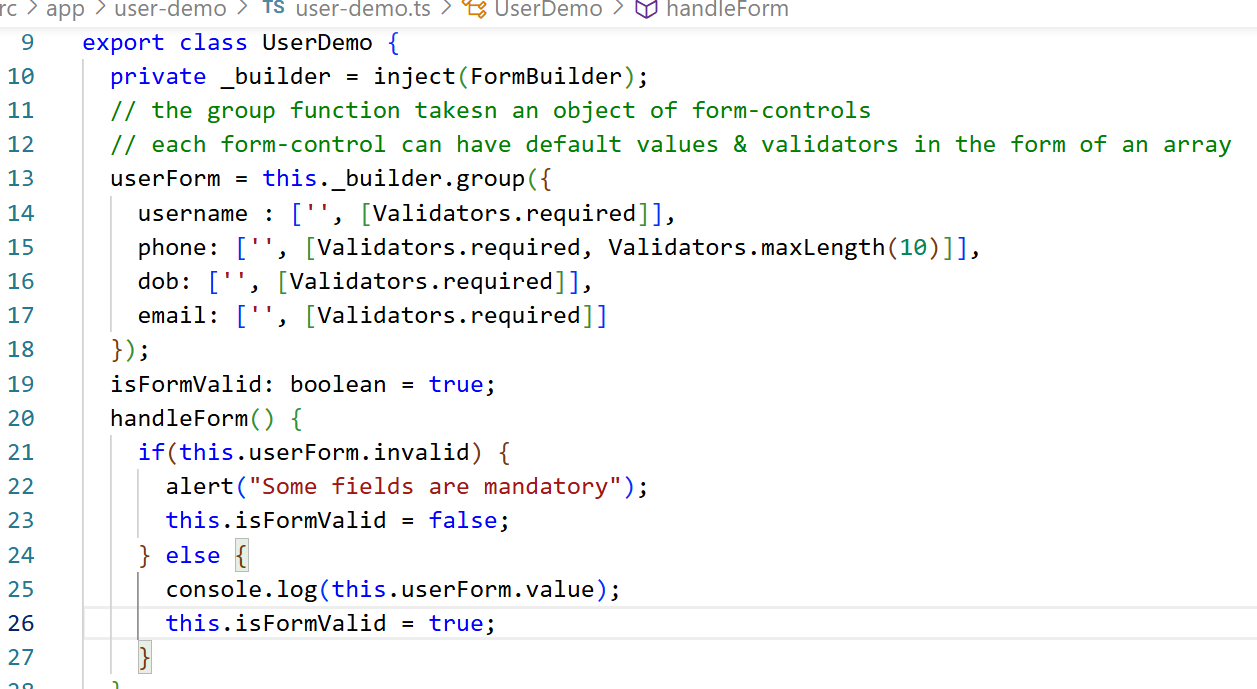
ex:

userForm = builder.group({ username : [“”, [valdtrs,valdtrs] ], phone: [“”, [validrs,validrs]] });

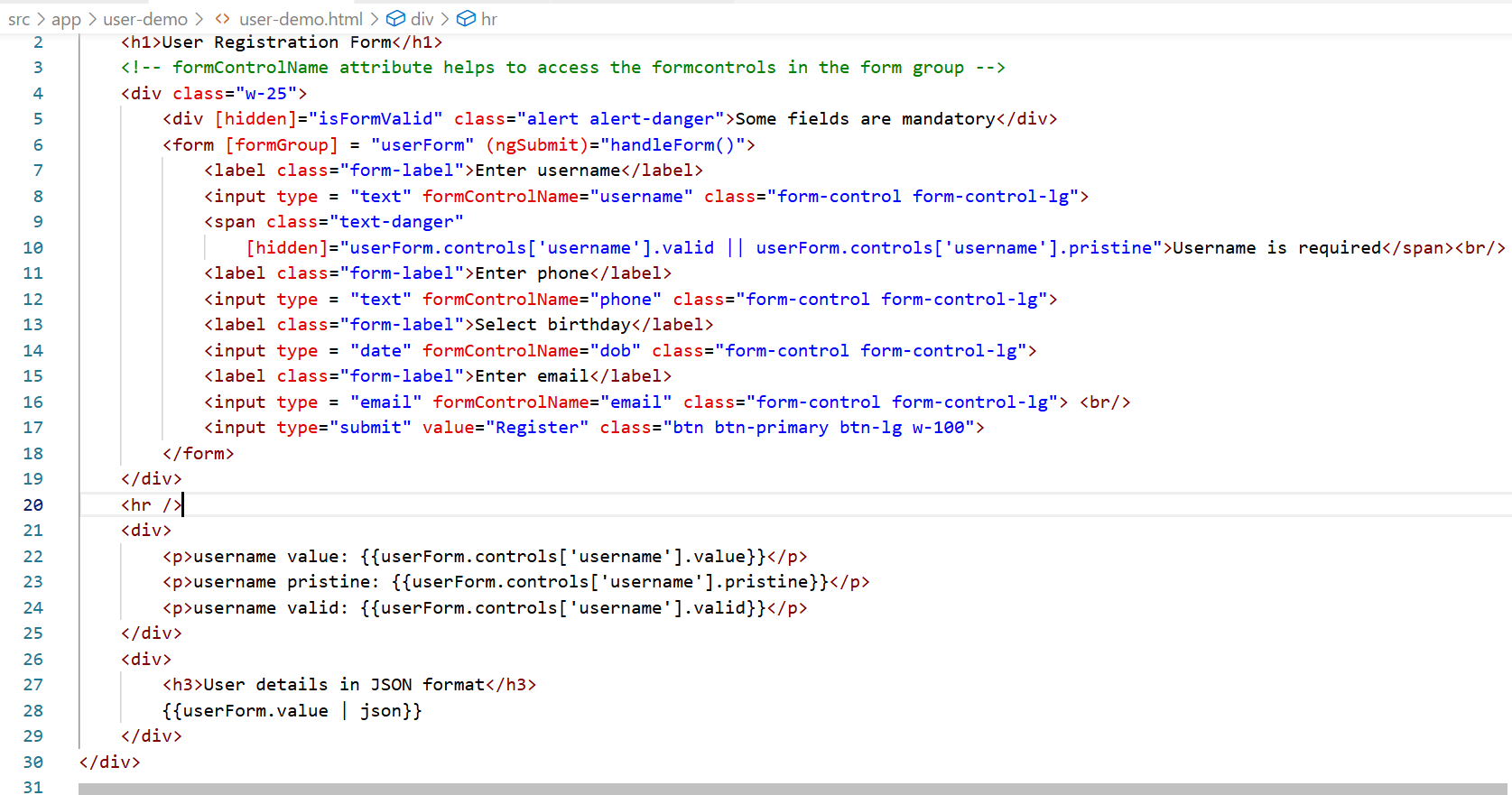
or

userForm = builder.group( { username: [“”, Validators.compose([validrs,validrs,..])]});

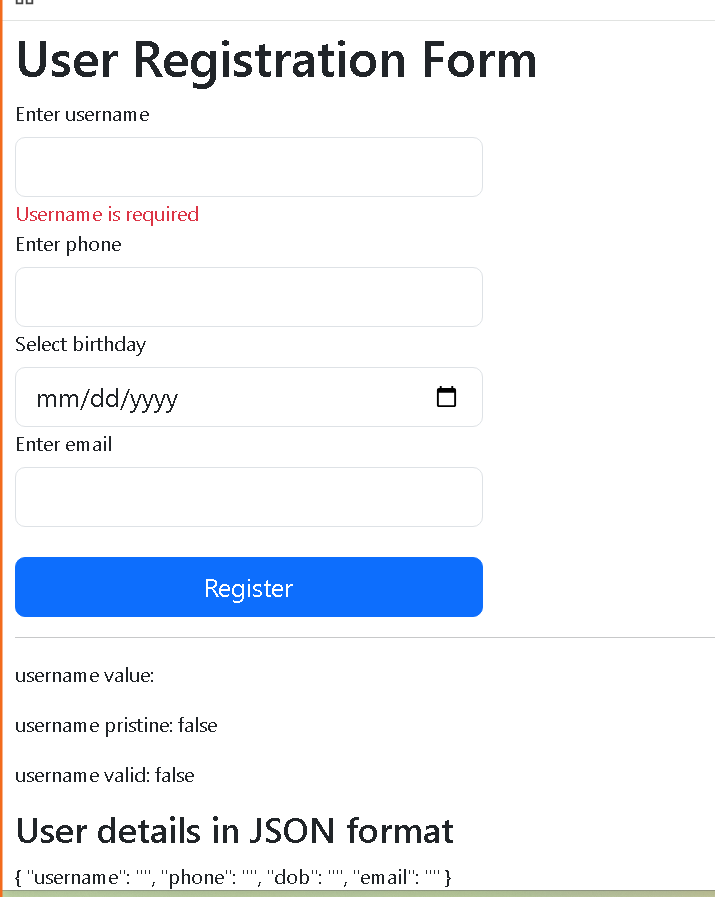
ts file



html file



Output:



Component to Component data sharing using @Input() and @Output() decorator

There will be two components minimum one will be parent component and another will be child component.

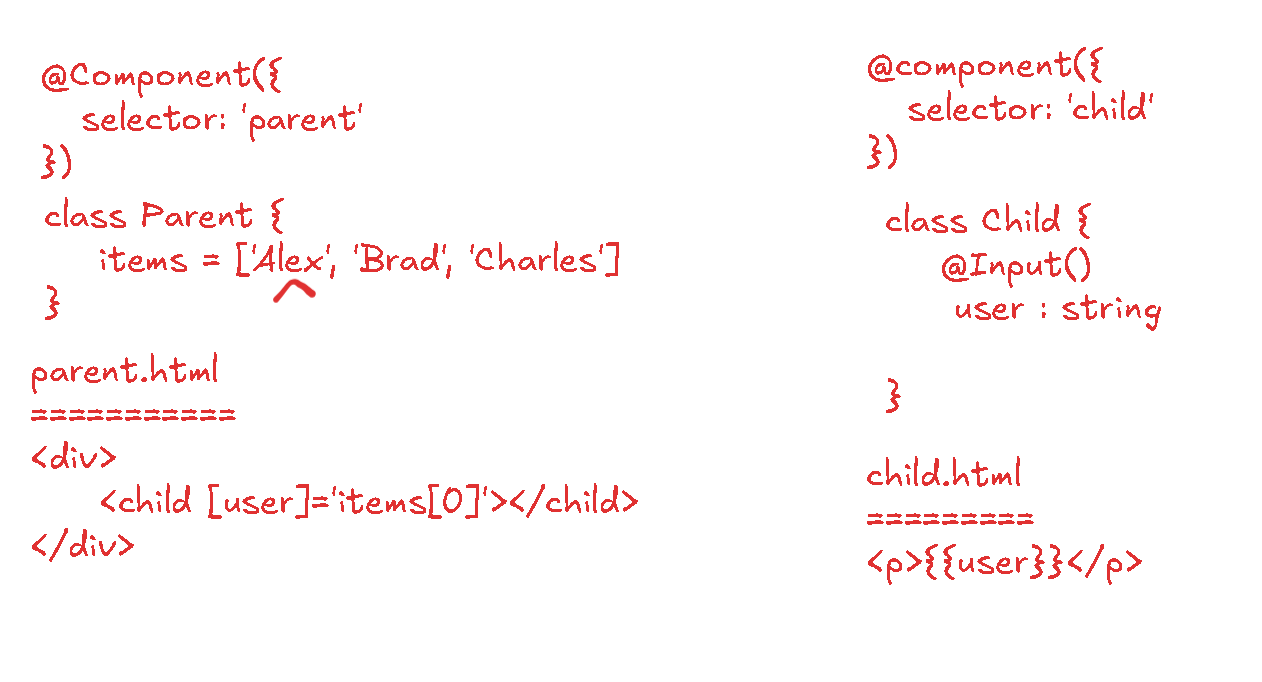
parent component: a component that nests the another component

child component: a component which is inside another component

@Input(): it is a decorator that reads the data / gets the data from the parent component, you must use this on top of the variable inside the child component

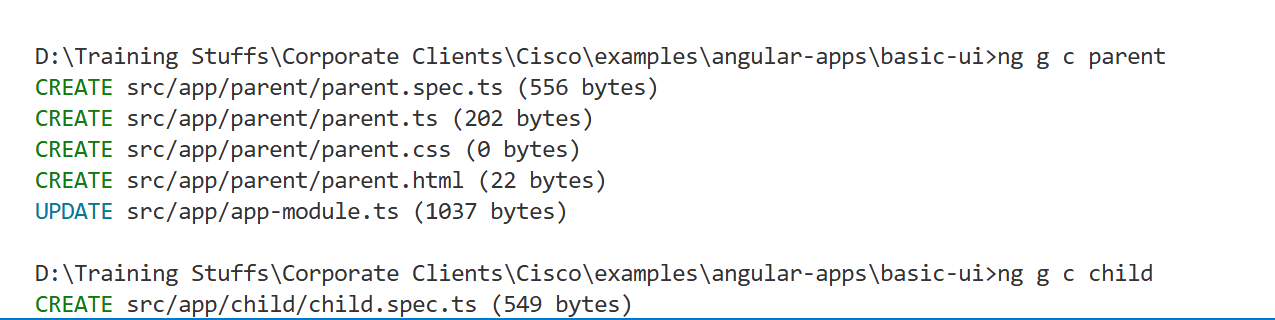
@Output(): it is a decorator that writes the data / pushes the data to the parent component, you must use this on top of the variable inside the child component

Passing the data from parent to child using @Input()



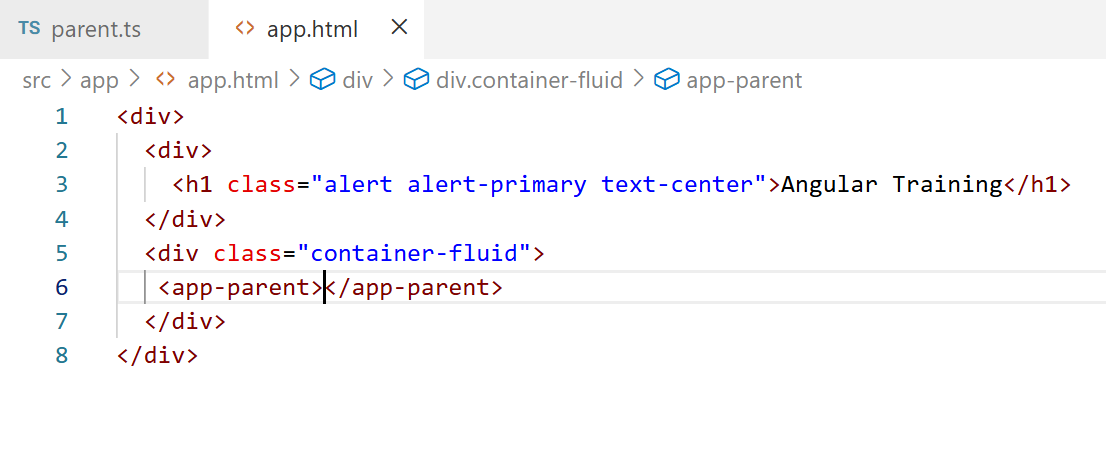
Here the parent component passes items[0] value to the child component property user, the user property must have @Input() decorator.

Generate two components parent & child

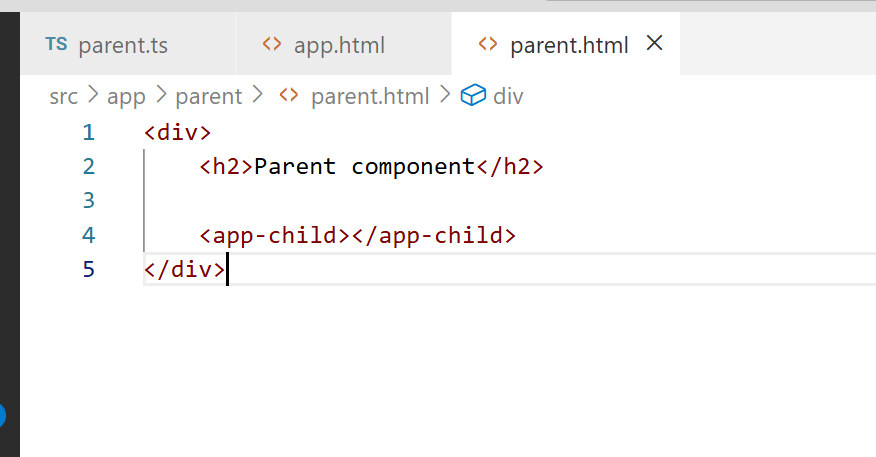


Add the parent to the root and then child to the parent

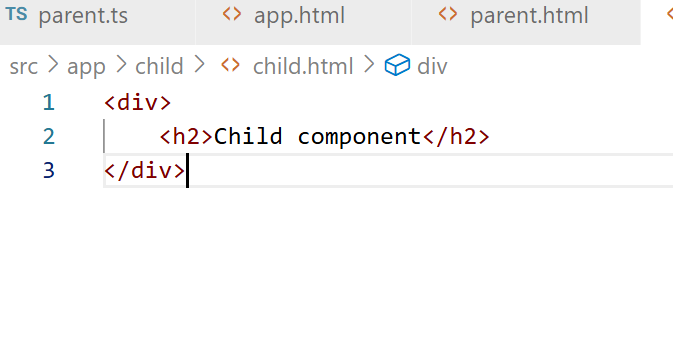
app.html



parent.html



child.html

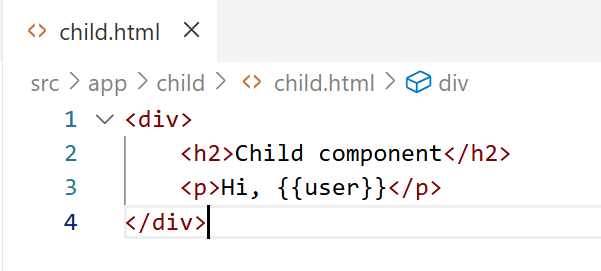


In the child component have a default value to the user variable

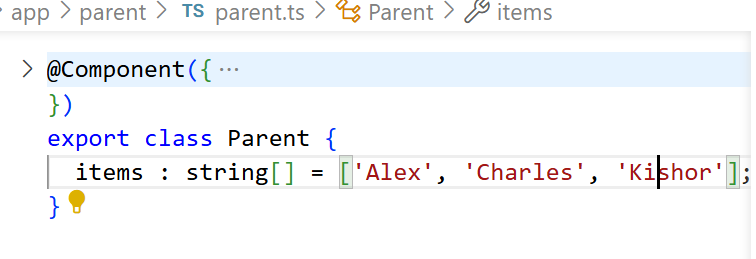
child.ts



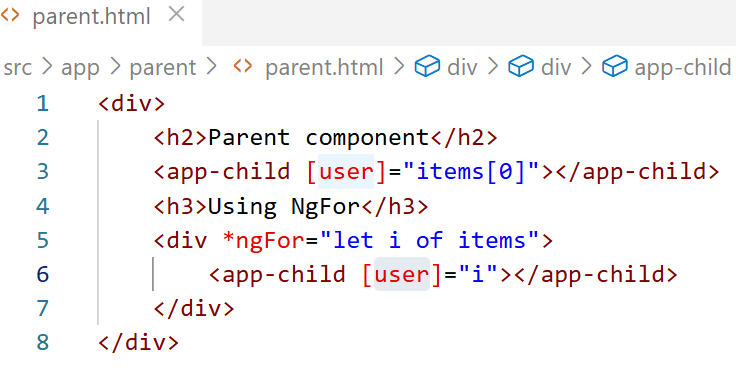
child.html



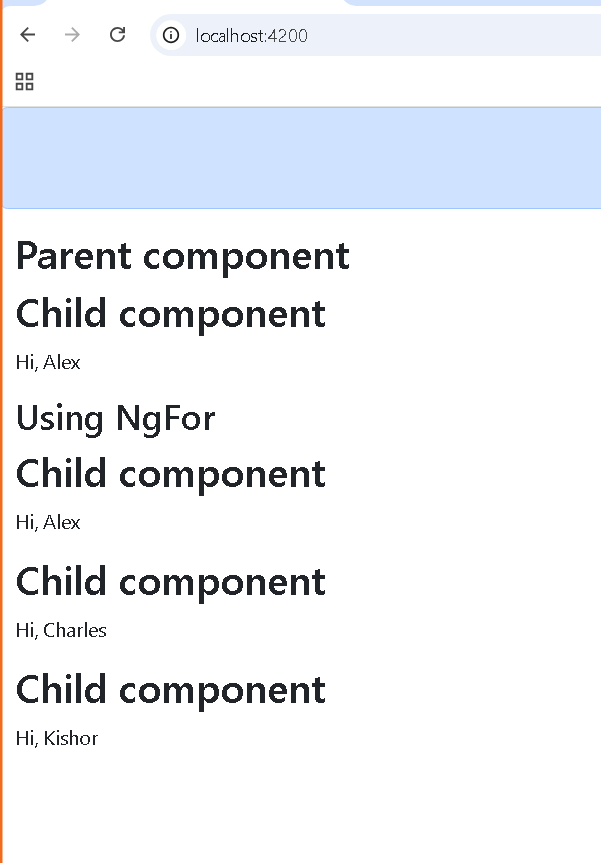
Create an array in the parent.ts



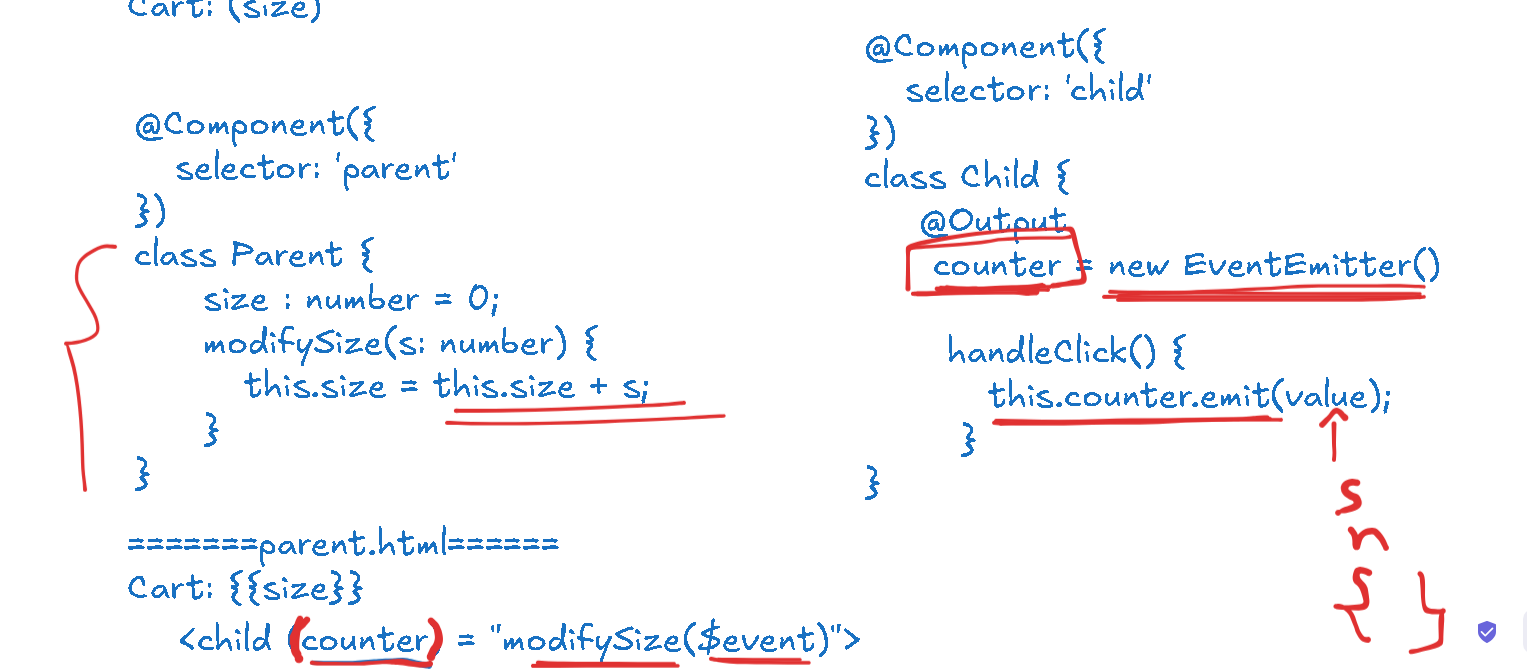
pass the element to the child component user property in parent.html



Output:



Passing the data from child to parent using @Output

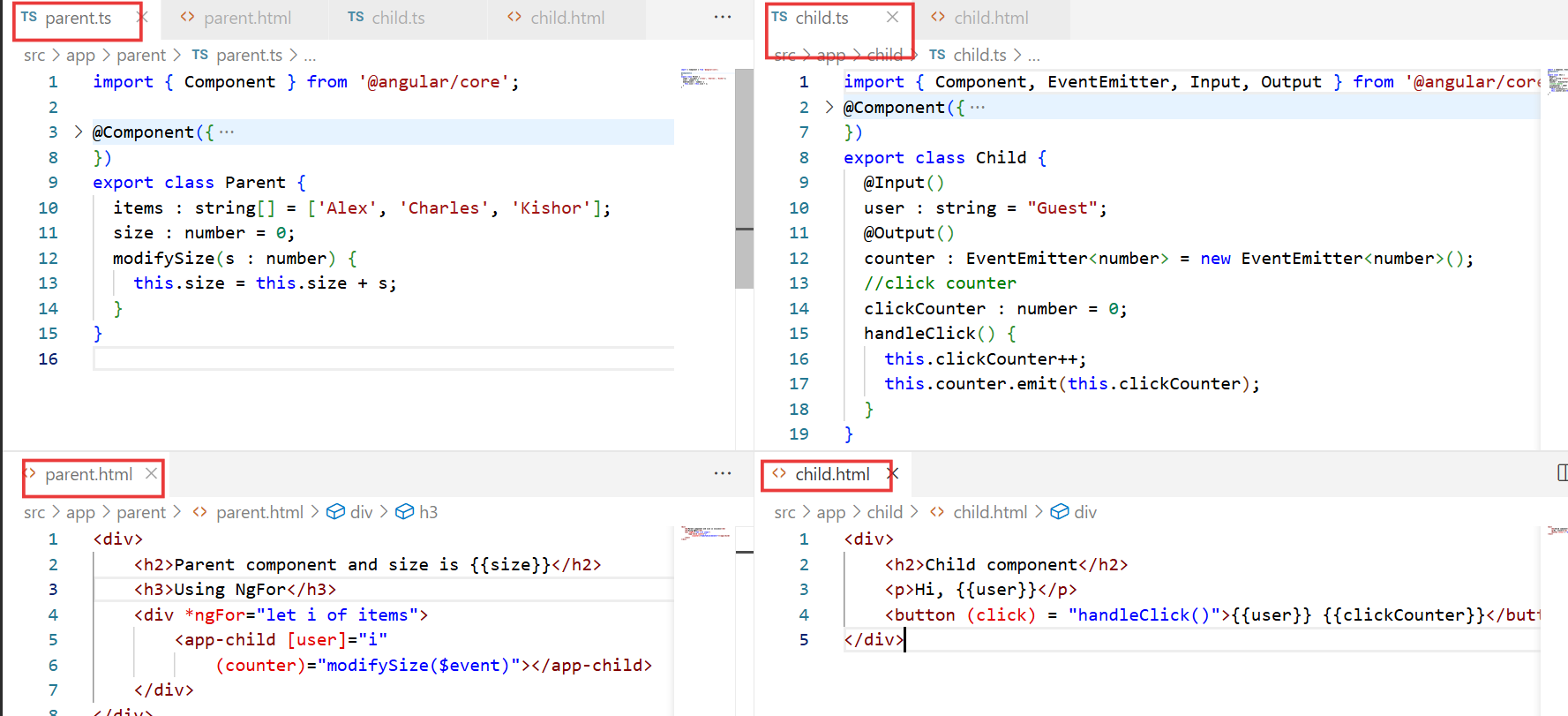


If the child component wants to share the data to the parent, then the child component must have a property of EventEmitter that must emit a value(a simple or complex type), that value is represented as $event which you can assign to the parent component

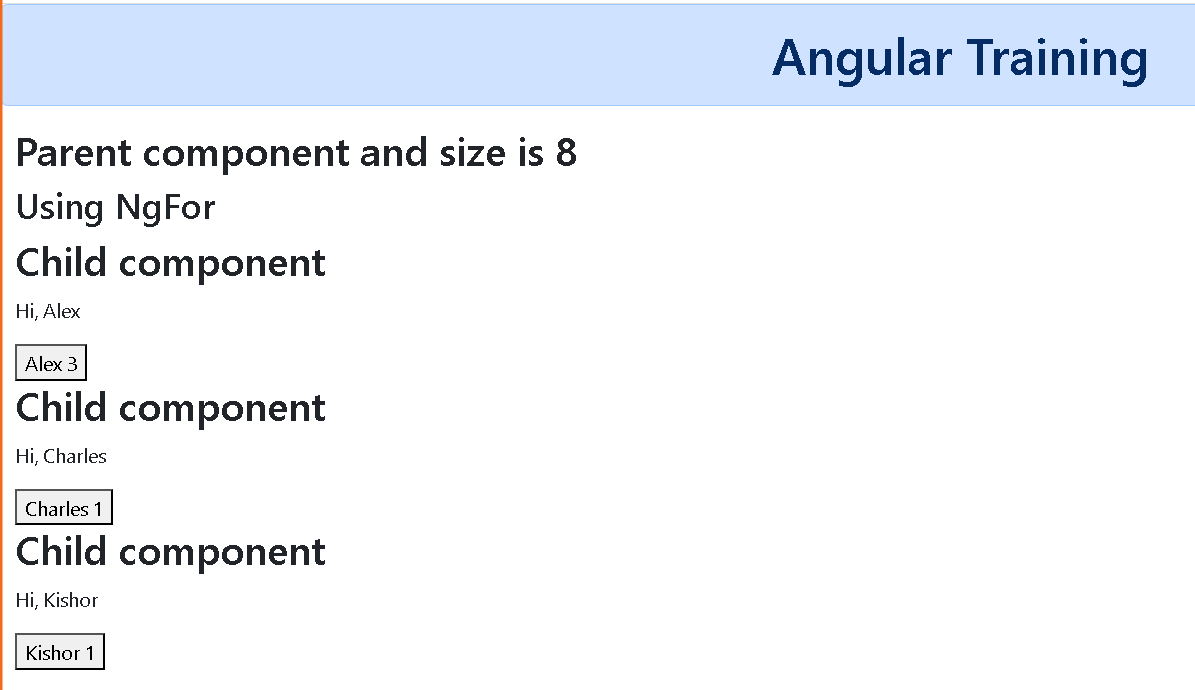
<child (counter) = “modifySize($event)”> or

<child (counter) = “size = $event”> # this might not increment with previous value

parent and child component code

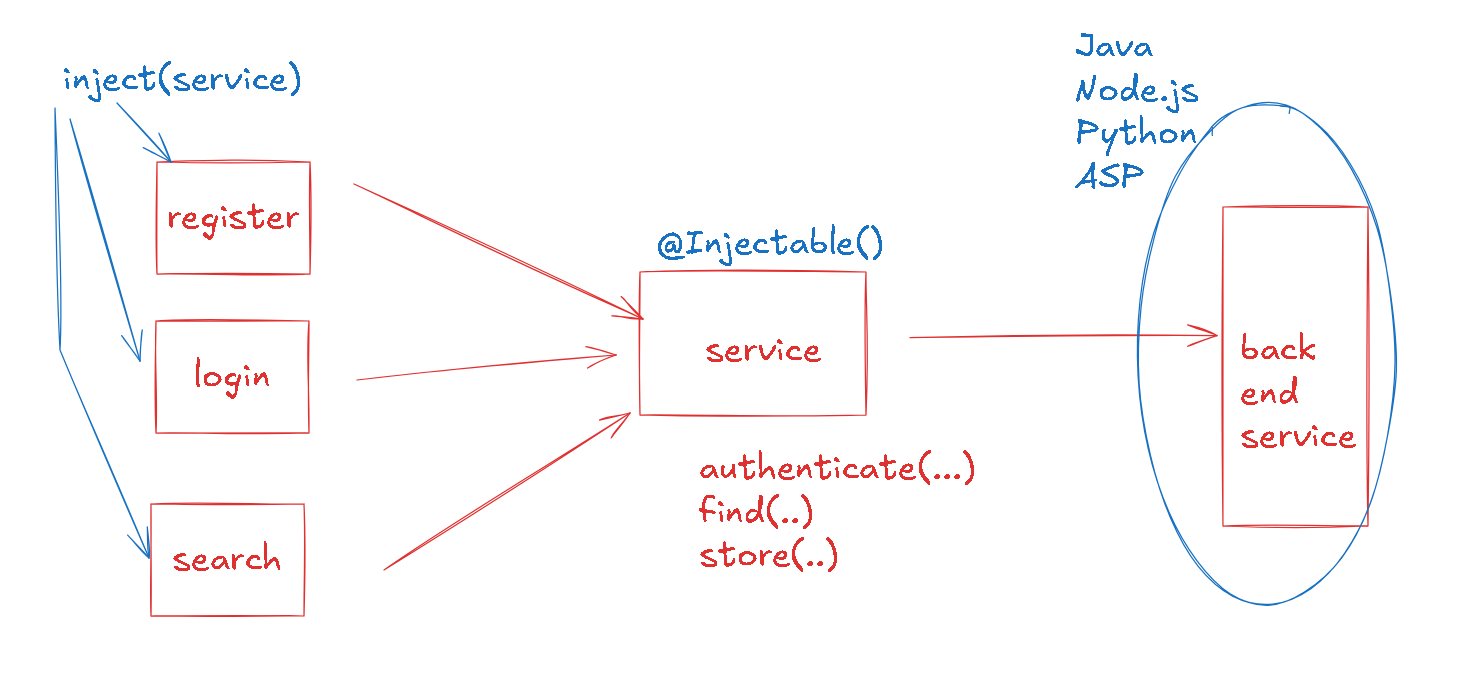


Output:

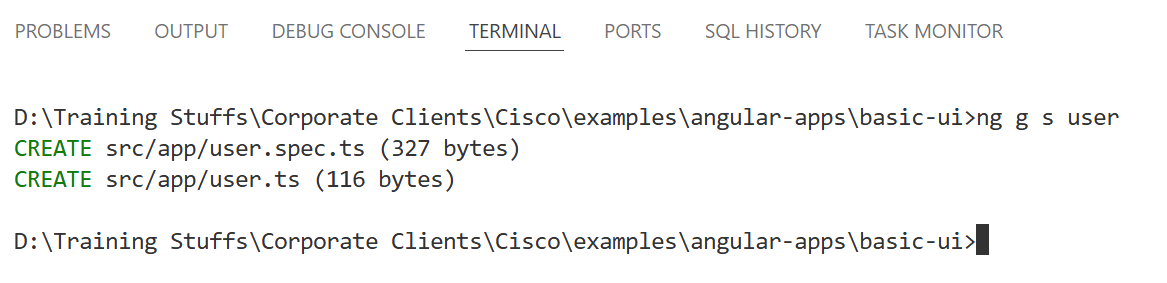


Services

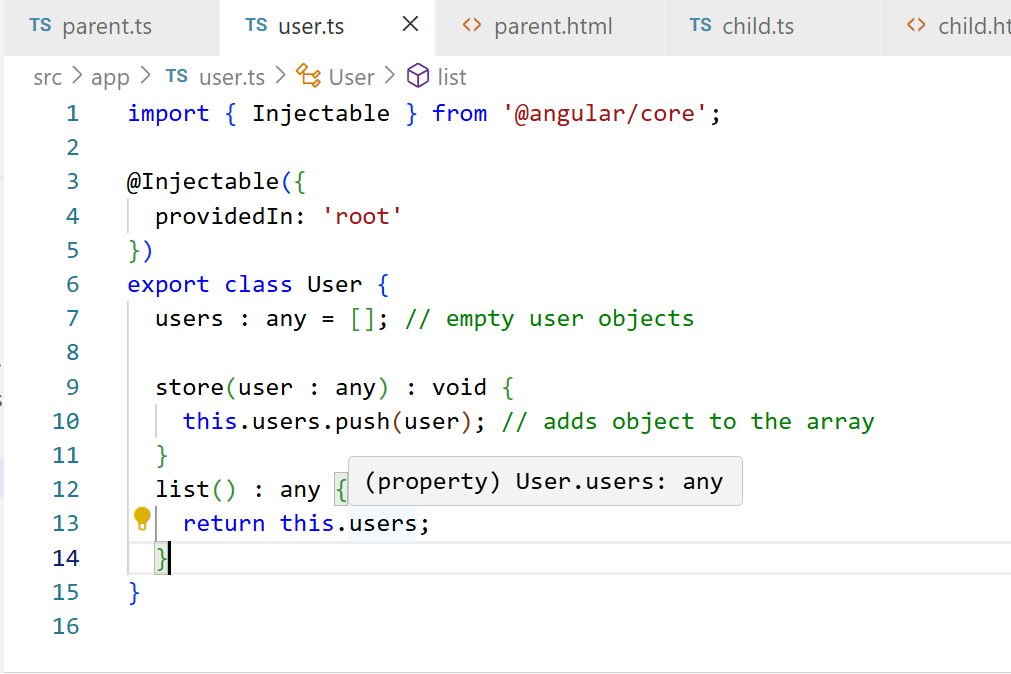
These are reusable logics which multiple components call, they are classes with @Injectable() decorator



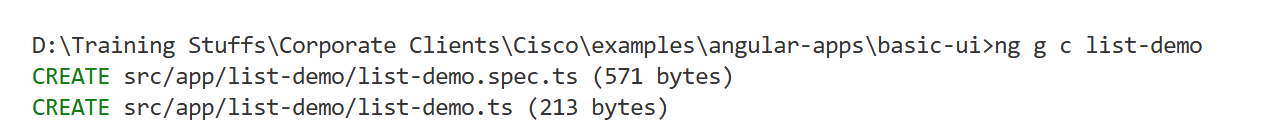
ng g s service-name creates a service class with @Injectable decorator.



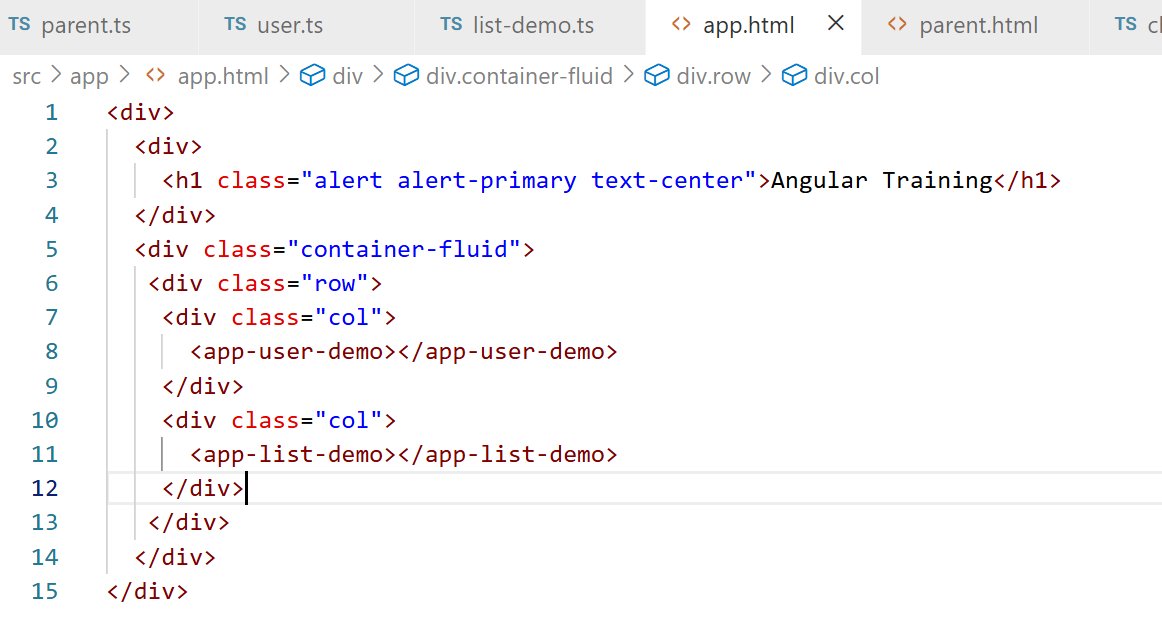
user.ts



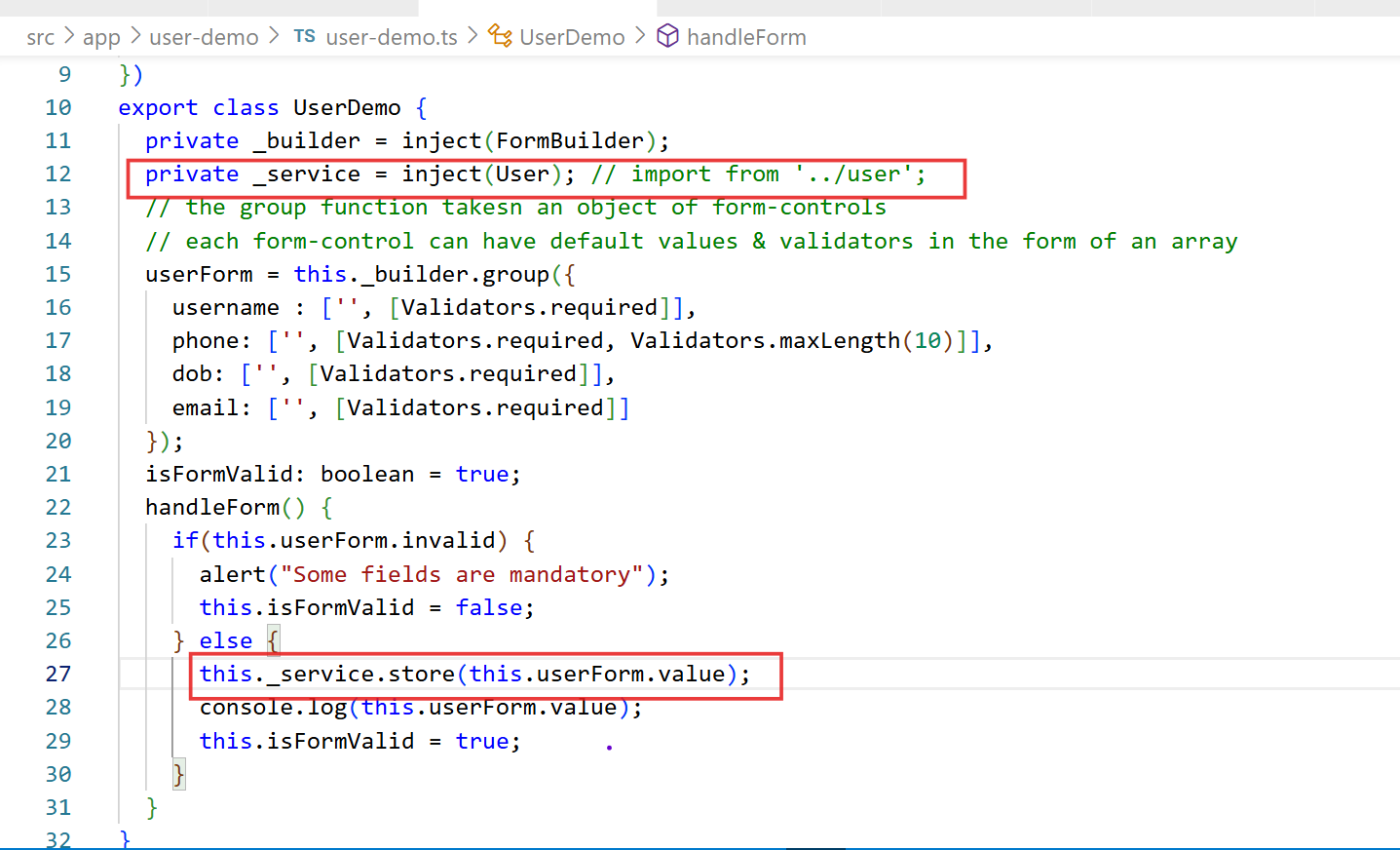
user-demo is already taking user input which can call the store method, however we don’t have a component to call the list method hence we will create list-demo component.



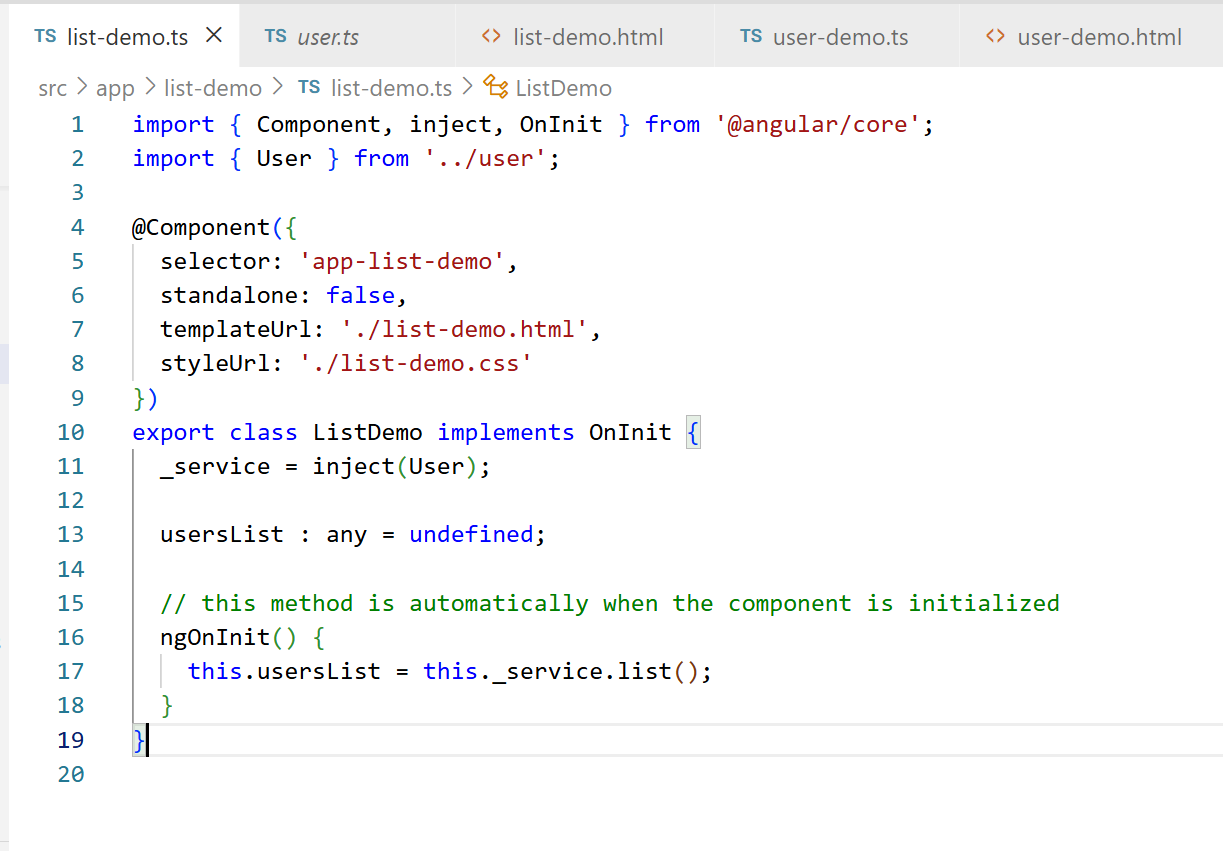
Use user-demo & list-demo both in the root component



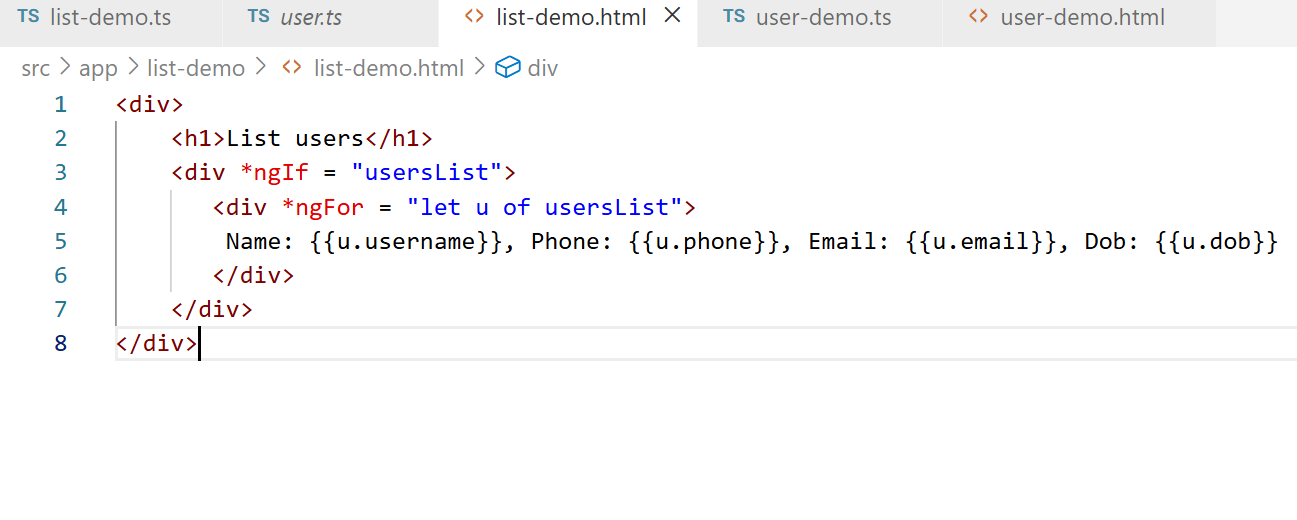
make changes in user-demo.ts file to inject(User) and call the store method



list-demo.ts



list-demo.html



Output:

