Project final tutorials

Tutorial 1 :-

Hello Welcome to first tutorial of your full stack development course.In this series we are going to implement all the concepts which we have learned in this course.so let’s with seeing the final output of the project which we can see here .

[Shows the project]

So we have seen the final output which we are going to achieve .So lets start with basic structure of the project and the folders which going to be there in the project.

So let’s open our code editor which is visual studio in my case . first we have to create an configuration folder which will have the configuration of the database which we are going to use in this project.Then next let’s create another folder which is going to be the views which is going to store the pages which will shown in the web browser using the handlebars templates.next we are oign to create public folder where we will be storing the css and external jaavscript files for some of our functions.net we are going to create an folder namely route where we will store our api which will interact with our database and does the crud operation on the database.So that’s it for this tutorial in the next tutorial we are going create the the configuration file for our database and routes for our api.

And I will see you in the next tutorial.

Tutorial 2:-

Hello and welcome to the second tutorial of you full development project.In this video we are going learn about creating files for database connection and api development for the project.So let’s begin with creating the database file for the project.before we start with the creation of files we are going to install the dependencies for the project.for this let’s open our terminal and go to the folder of our project. Now we are going to initialize by running the command npm init .

[Runs the command]

Next we are going to install the express framework for creation of our api and the virtual server,express-handlebars for creation of front end of our project and lastly mysql2 for connecting our api to the mysql database.now we are again going to our terminal to install these dependencies and type the command “npm install express,express-handlebars,mysql2 --save” and then hit enter

[Runs the command]

After running the command we are going create a file namely database.js for storing our configration file for the database in order to connect it to the database.for this first we going to import mysql2 dependency which have installed and then next we are going declare and constant namely db and we are going to assign the value using mysql.creatConnection method available in the dependency and going to declare the host which the url where the database is located which is localhost in this case.Next we are going to declare on which our database is running then the user and password which we will use to connect it to the database and lastly the database which we are going to use in this project which is digitalprofile in this case . after creating this constant we are going to export the constant in order to use it in our project and connect it to the api using module.exports method in express.

Next we are going to create namely skillsFunctions.js which will handle the data interaction of our skills table inside our api.so let’s create the file and let’s start writing code for this file.Now in order to the rest api for this first we are going to first import the express in this file and then create router by create an router namely router from express.Router class .Next we are going to import the db constant from the configuration folder from the database.js file.After this we are going to create the post from router in order to insert the skills of the user into the skills table.to create this route first we create the endpoint of our api which going to add and write in single quote with froward before the the add keyword.Next we create the an arrow function using the es7 syntax which we have learned before which will accept two input namely req and res which are request and response.next we going to create two constants id and name which we read from the request’s body.enxt we are going to declare an constant sql which of string type which has value of insert mysql of skill table.next we declare the constant body which will contain the key value pair having database field name and it’s corresponding value as shown in the code.After this we are going create an variable namely query which execute the query and insert the value in the table by using the db.query method which accept two input namely sql and body which we have declared earlier and next we are going to create an function which accepts two variables namely err and result and inside the function we are going to create an if else in which in case of error we are going to display the error in the json format and in case of success we are going to display an message telling that skill is added.

Next we are going to create an delete method using the router and we are going to take the input of skill id from the url or endpoint parameters. So we are going to use router.delete method and going to take from url name id and then going create an method as before which takes two variables namely req and res using the arrow function and inside the function first we are going to read the id from the url using de structuring using the req.params method and like last function we are going to create an constant sql whose value will be the sql query for deleting the skill row using the id and then we are going to create the body constant which will be an array with id as it’s member.And like last time we are going declare query using db.query method and going to input two vlaues which will be sql and body and then going to create an arrow function which will take two values err and result and then inside the function we are going to create an if else case which will handle the error and displays the error in case of any error and also displays success in case of successful query.

Now after creting these two methods we are going to export the router using module.exports inorder to use in the main entry file.

Next we are going to create next file which is going to be userFunctions.js which will handles the crud operations for users table.now in order to perform the crud operations on users table we are going import express and create an constant router using the express.Router() class and then we are going to import the database from the configuration and then start creating the post method first like we did in the last file then we are going to create the delete file as we did last time using the id .Please take notes of the field and its corresponding values as it is important we don’t miss any fields and can cause errors and after creating this two methods we are going to create an update method which will take input from both url and form body so let’s create this method for updating the user table we are going to use the put method from router using the router.put and then going to create the endpoint which is going to take input of id from the url while the field from the body which has to be updated in the table .next like the same we had done earlier in course we are going create an function with two input values req and res respectively.Then we are going to get the values from using de structuring from both url and body.then we are going create an sql statement and then the body as array with the values which has to updated in the same order as the fields are mentioned in the sql query and then we are going to create the query and in similar fashion we are going create whole method using db.query with sql and boy input and lastly create an function with with err and result with if else case in the body of the function in order to handle the error and after creating the method we are going to export the router using module.export in order to use it in the main entry file.

So that’s it for this tutorial in the next video we are going to create the main entry file and also the database and test api using postman and I will see you in the next video.