**Day 9: 12 April 2025**

**Connection mongo Db database using Node JS (Express JS)**

**Mongodb module:** it is a low level or native api module which help to connect mongo db database using node js. This module doesn’t provide schema features.

**Mongoose module :** it is a third party module which support features as ODM (Object Data Modelling). Which internally use mongodb module but provide few extra features as. It support schema concept( we can mention collection name with number of attribute with their data types, few attribute are option). It allow to do validation. It support few middleware features. In mongos we use schema concept to interact with Database like deal with object rather than native query

Generally while developing the application we need to divided the code into different files base upon that code functionality.

**Creating user defined modules**

**Module** is a collection of variable, function, classes which have same but different purpose.

We can create user defined module using

ES5 style : require and export : by default type is consider as commonjs. So in ES5 style we need to use module.exports and require function.

ES6 style : import and export : if we want to use import and export we need to change type as module.

**Product – CRUD Operation using Express and mongoose module**

**Products** -🡪 Collection

**MVC (Model View Controller)**

**mkdir product\_crud\_operation\_using\_mongoose**

cd **product-crud-operation**

**npm init -y**

**npm install express mongoose**

**open the project in VS code**

**config -🡪resource details**

**db.js :** this file provide database connection.

**models -🡪Folder**

**products.model.js :** model is a type of file which provide schema definition details (name of collection, number of attributes with their data types and optional attribute) as well as collection details connect with db.

**repositories🡪 folder -🡪 database logic**

**products.repository.js :** this file contains more than one database related function like store, delete, update and retrieve etc. (pure database code).

**services -🡪 folder -🡪 business logic**

**products.service.js :** this file contains business methods. Before store data in database we need apply some condition. If condition true then only call repository layer else return the message we can’t do that operation. This file contains pure business logic.

**controllers 🡪folder**

**products.controller.js :**  this file ready to receive request from client and base upon request pass the value to service layer and base upon response it send the result to client application. Response can be in json or text format.

**routes-🡪folder**

**products.routes.js :** this file contains routing details. It provide http method details like get, post, put and delete. It provide sub path details and base upon path it will call controller layer.

**app.js or server.js or main.js**

Which is responsible to load the module, use any middleware if required, then provide main path for applications, call the route file and provide the port number details and run the application.

Node js provided one the pre defined module is **nodemon**

this module keep the track of the application whenever we do any changes automatically your project refresh. This module we need to use in development.

**npm install nodemon -g**

after installed you need to run project using **nodemon** rather than **node**

**node app.js**  : need to stop the project and re-run if we do any changes in project.

**nodemon app.js**