1)what is ExpressJS=============

a)express is a backend web applocation framwork for nodeJS.

b)it is used to build web and mobile applications and API.

c)to develop web application we should install and import express

d)and aslo we should write expresss() method

2.what is ODM

\*ODM stands Object Data Modeling .it work like bredge b/w nodejs seerver and mongodb

3.what is mongoose

\* Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js.

\* It manages relationships between data, provides schema validation,

and is used to translate between objects in code and the representation of those objects in MongoDB.

4.what is Middleware

Middleware functions are functions that have access to the request object (req),

and response object (res), and the next middleware function in the application’s request-response cycle.

==here next is used for accepting the anthor middleware ,if we are not write this it will be struced

===>uses <==

->Execute any code.

->Make changes to the request and the response objects.

->End the request-response cycle.

->Call the next middleware function in the stack.

5.type of Middleware

->Application level Middleware

->Roter level Middleware

->error handlling Middleware

->built in Middleware

->Third part Middleware

1.Application-level middleware

application-level middleware to an instance of the app object by using the app.use() and app.METHOD() functions,

where METHOD is the HTTP method of the request that the middleware function handles

(such as GET, PUT, or POST) in lowercase.

ex:

var app = express()

app.use(function (req, res, next) {

console.log('Time:', Date.now())

next()

})

2.Error-handling middleware

simillar to application-level middleware .use of this handling error

it is having 4 argumets are objects .

ex:

app.use(function (err, req, res, next) {

console.error(err.stack)

res.status(500).send('Something broke!')

})

3.Built-in middleware

use of this middileware accepting the static accests

it is having 3 type of middlewares

->express.static serves static assets such as HTML files, images, and so on.

->express.json parses incoming requests with JSON payloads.

->express.urlencoded parses incoming requests with URL-encoded payloads.

4.Third-party middleware

use of this module add functionality to Express apps.

like athentication,possport, storing into database Execute

5.Router level middleware work just like application level middleware except

they are bound to an instance of express.

Router() . The middleware system created at the application level in

the example above, can be replicated at the router level using the following code. ...

var routes = require('./routes/index'); app.

5)we have many built in modules whereever we used to creating web application

examples

a) "body-parser": "^1.19.0",---: used to parse incomming request before your handlres.

with helpl od req.body.it is based on user inputs

example

// create application/json parser

var jsonParser = bodyParser.json()

// create application/x-www-form-urlencoded parser

var urlencodedParser = bodyParser.urlencoded({ extended: false })

b)"bootstrap": "^4.6.0", ---:it is used to give bootstrap framwork

c)"connect-flash": "^0.1.1",--:it is used flash messages mainly used when page redirecting to anotherpage

it working on session based.

example

const flash=require('connect-flash');

"express-session": "^1.17.1",

app.use(session({

secret:"secret",

resave:true,

saveUninitialized:true,

}));

app.use(flash());

"dotenv": "^8.2.0",--: it ised to load enviranmetn variables .in .env files proses.env.

storing the configaration and saparate the code.

ex:

DB\_HOST=localhost

DB\_USER=root

DB\_PASS=s1mpl3

"express": "^4.17.1", --:it is used create express applicatioin

"express-handlebars": "^5.2.1",--:it is uesd to handlng the html code with handlebar view engin

we can write any name for extention like home.handlebars or home.hld exceptit is following starcture

view-inside folder again 2 and one file, folders loayouts, partials and file is home.handlebars

ex:

const exphbs=require('express-handlebars');

app.engine('handlebars', exphbs());

app.set('view engin', 'handlebars');

"method-override": "^3.0.0",--:HTTP verbs such as PUT or DELETE in places where the client doesn’t support it.

ex:

app.use(methodOverride('\_method'));

"mongoose": "^5.11.18",:schema validation connecting to the node and mongodb

"multer": "^1.4.2",--:muter is used to the uploadinf file into the database

"popper.js": "^1.16.1",