



ATHENTICATION

BY : C FOR CODING

CREATE A SERVER

01 - EXPRESS

Express.js is a fast, flexible and web framework for Node.js. It's effectively a tool that simplifies building web applications and APIs using JavaScript on the server side.

02 - LISTEN

Express `app.listen()` method creates an http server object and then configures it to receive incoming TCP connections on a specific port and IP address so that when clients request a connection to that port and send an http request, the server can receive that http request and process it, sending a response.

03 - PORT

You can specify the port number on which your Express server will listen. This is usually done by defining a variable for the port and then using it in the `app.listen` method.



CONNECT TO DATABASE

01 - MONGOOSE.CONNECT

The `mongoose.connect()` function in Mongoose is used to establish a connection between your Node.js application and a MongoDB database.

`mongoose.connect(uri)`

uri: A connection string that specifies the location of the MongoDB database you want to connect to.

MODEL:USER SCHEMA

01 -SCHEMA

To define a schema in Mongoose, you use the Schema class provided by the Mongoose library. Here's a basic example:

```
import { mongoose, Schema } from "mongoose";
const userSchema = new Schema(
{
  name: {
    type: String,
    require: true,
    trim: true
  }
})
```

02-CREATING A MODEL

After defining a schema, you create a model from it. The model provides an interface to interact with the MongoDB collection:

```
mongoose.model('User', userSchema);
```

ROUTES

In Express.js, a "route" is a specific URL pattern that an application responds to with a function. It determines how HTTP requests are handled by the server.

```
// Router object  
const router = express.Router()
```

01-REGISTER ROUTER

```
// METHOD POST  
router.post('/register', registerController)
```

02-LOGIN ROUTER

```
// METHOD POST  
router.post('/login', loginController)
```

HASH PASSWORD & COMPAREPASSWORD

bcrypt: bcrypt is a popular library used for hashing passwords in Node.js applications.

Hashing: A one-way function that converts the original password into a fixed-length string.

Salting: This ensures that even if two users have the same password, their hashed values will differ.

01-HASHPASSWORD

use in Registration controller

```
bcrypt.hash(password, saltRounds)
```

02-COMPAREPASSWORD

use in Login controller

```
bcrypt.compare(password, hashed pass)
```

CONTROLLER

A "controller" is a JavaScript module that contains functions to handle requests for specific routes.

01-REGISTER CONTROLLER

- Access req
- validate
- check user for existence
- Register user

01-LOGIN CONTROLLER

- Access req(email,pass)
- validate
- check email
- check pass
- Login

TOKEN

JSON Web Tokens (JWT) are compact, URL-safe tokens used for securely transmitting information between parties as a JSON object. They are commonly used for authentication and information exchange

```
const token = jwt.sign(id, secretKey, { expiresIn: '1h' });  
return token;
```

01-LOGIN CONTROLLER

- Access req(email,pass)
- validate
- check email
- check pass
- Login



THANK YOU

C FOR CODING