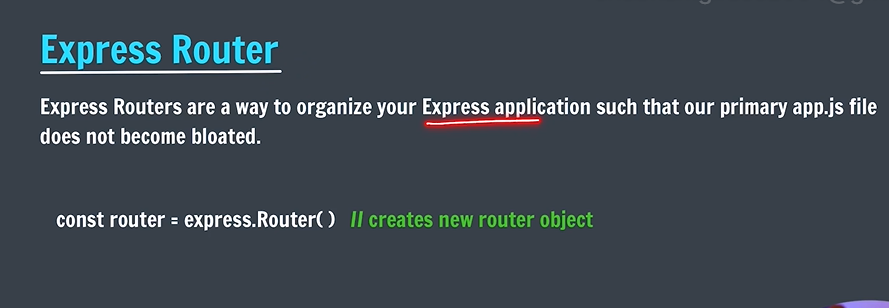
**Major Project module 2**

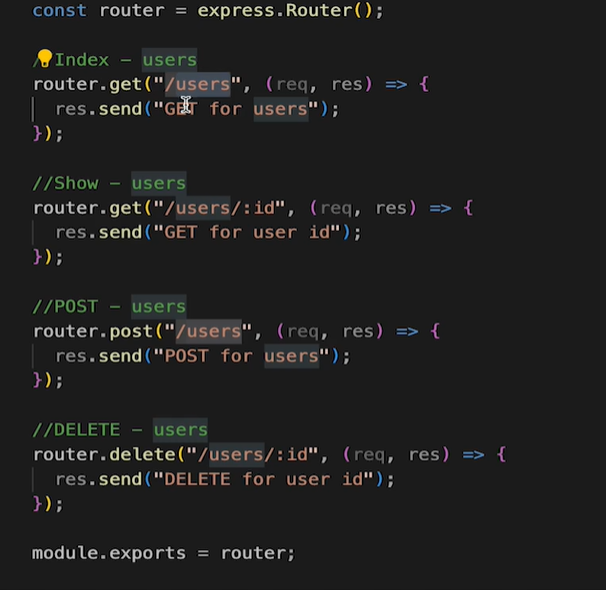
****

**Use to segregate routes..for programming convenience**

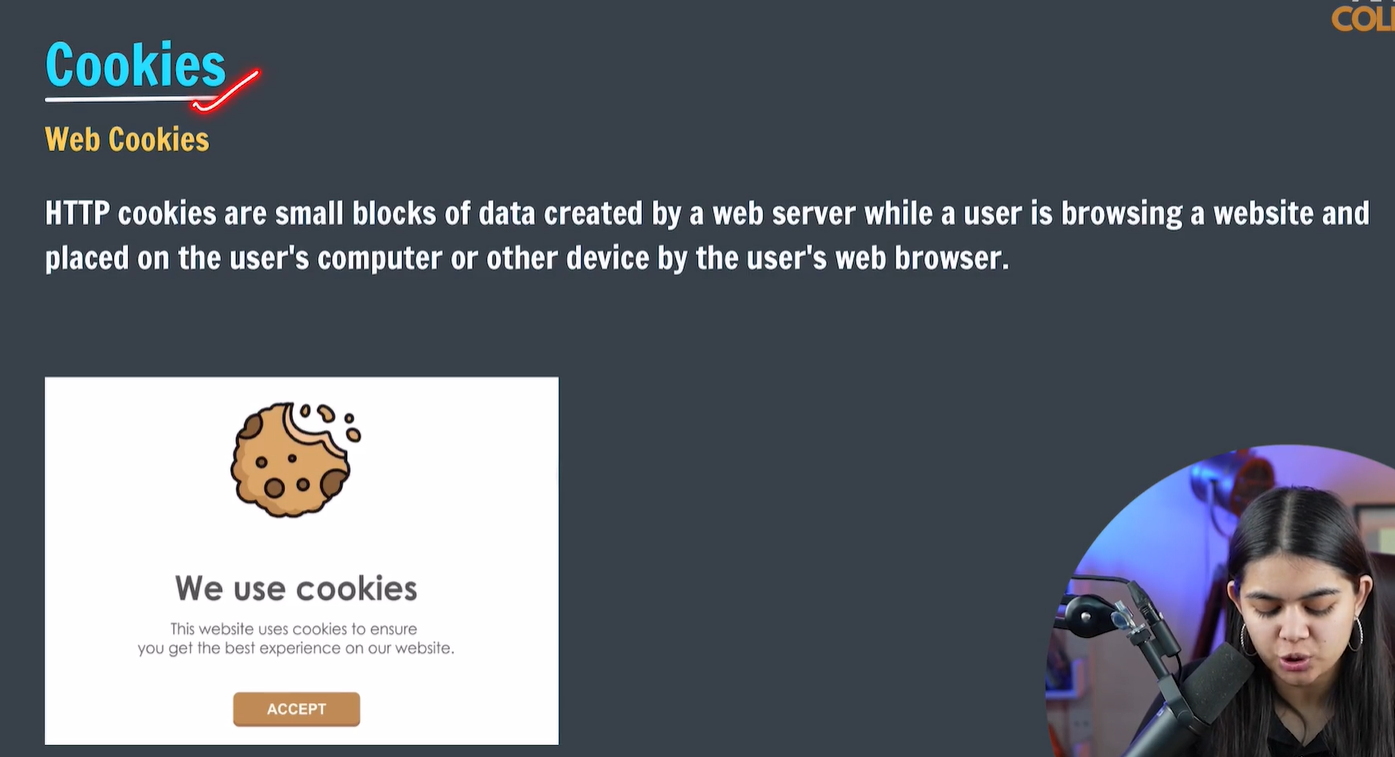
**Require it in app.js .**

**Common path is used in app.js**

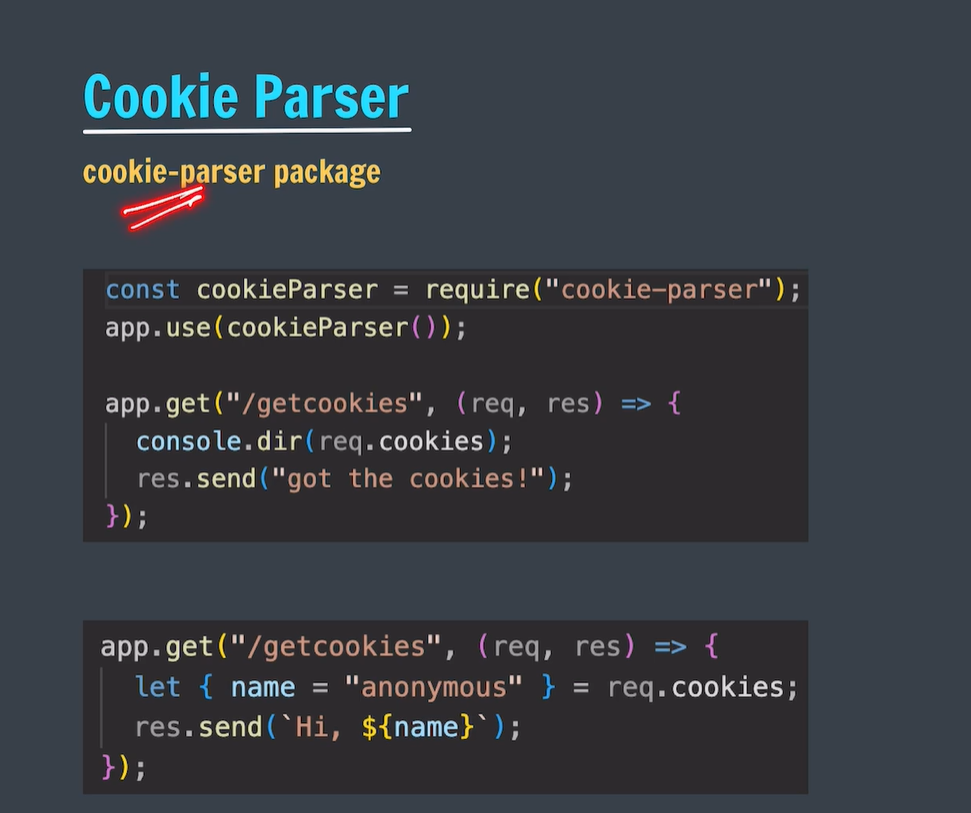
****

****

**Web cookies**

****

****

****

**To access cookies of another route, we need to parse them for that purpose we use cookie parser middleware.**

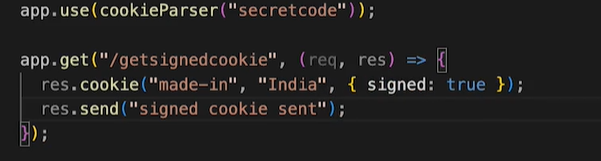
**Npm I cookie parser before using it.**

****

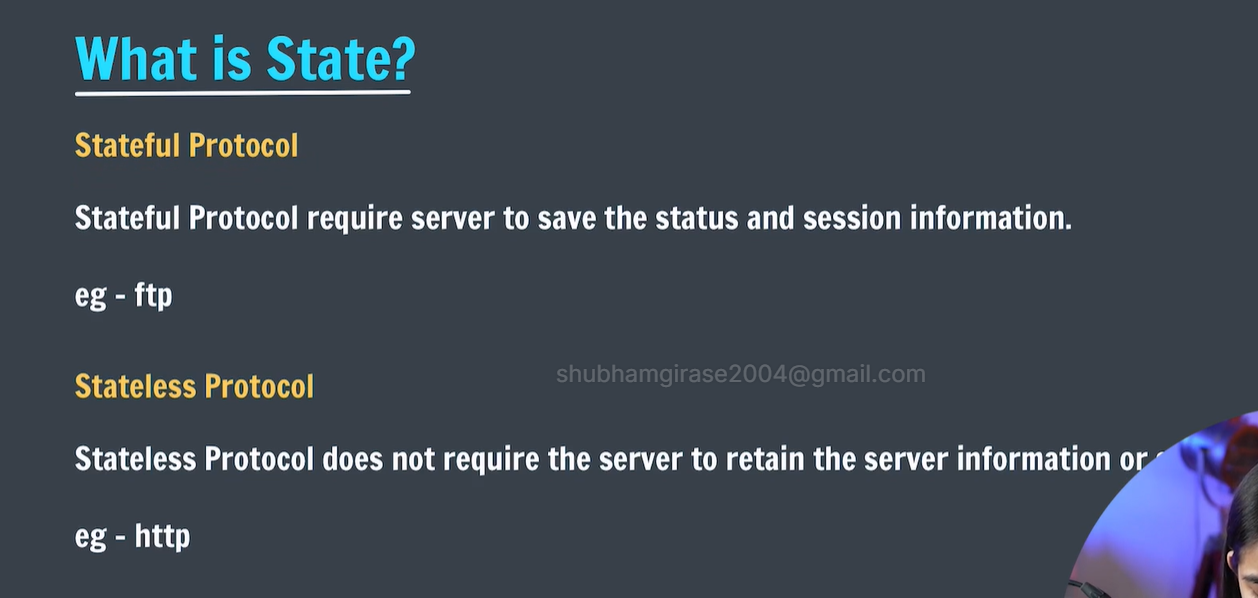
**To use it :   
**

****

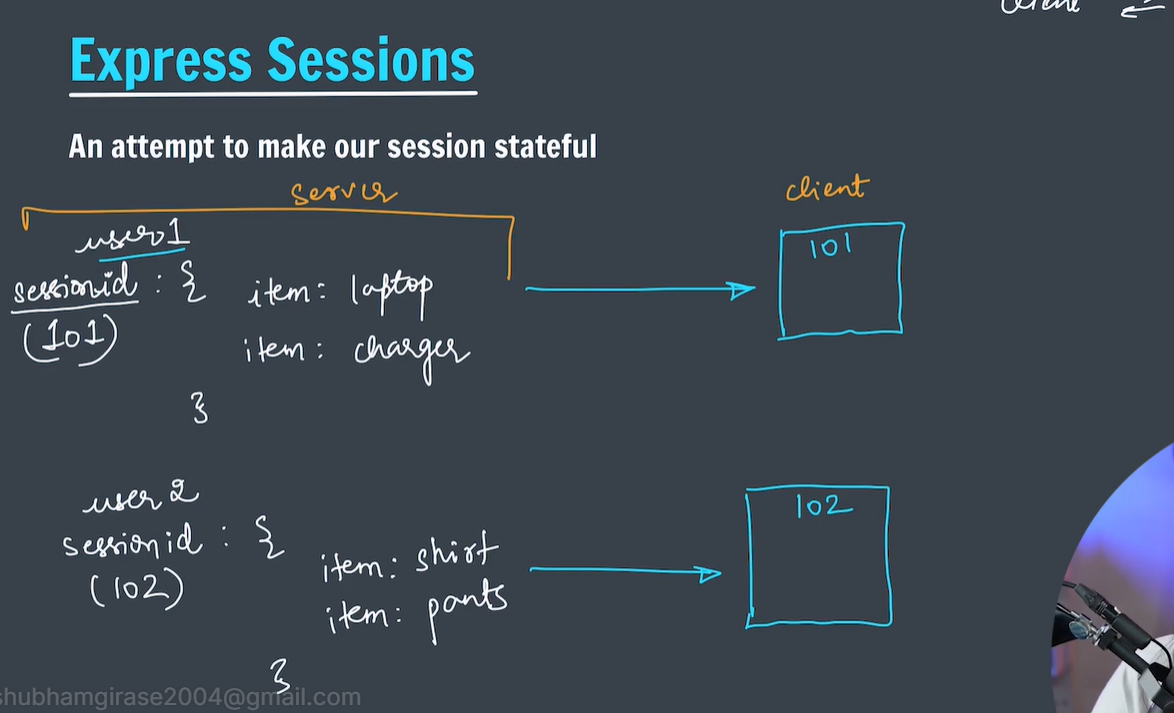
**Signed cookies make sure that receiving cookies are not manipulated and tempered.(maintain integrity).**

****

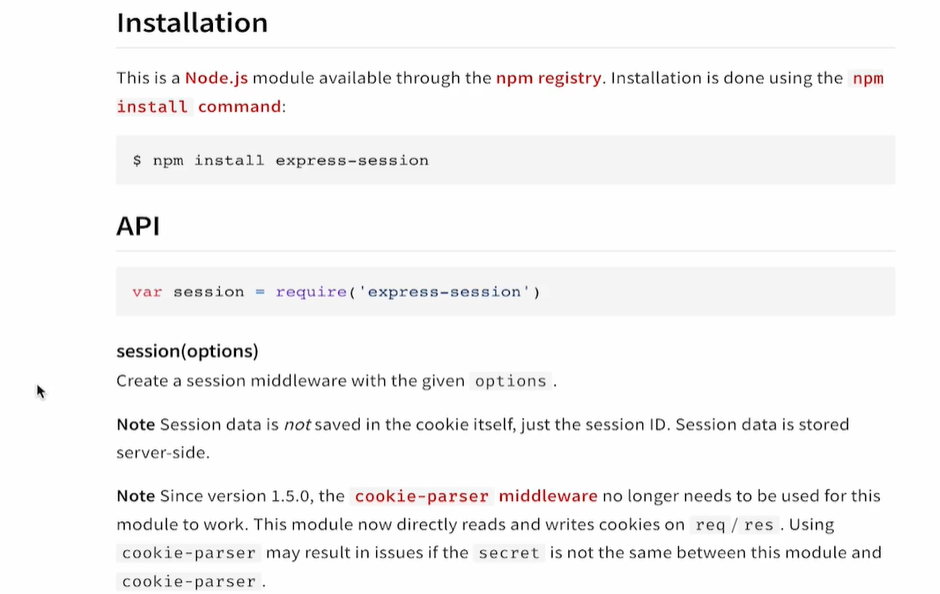
**State**

****

****

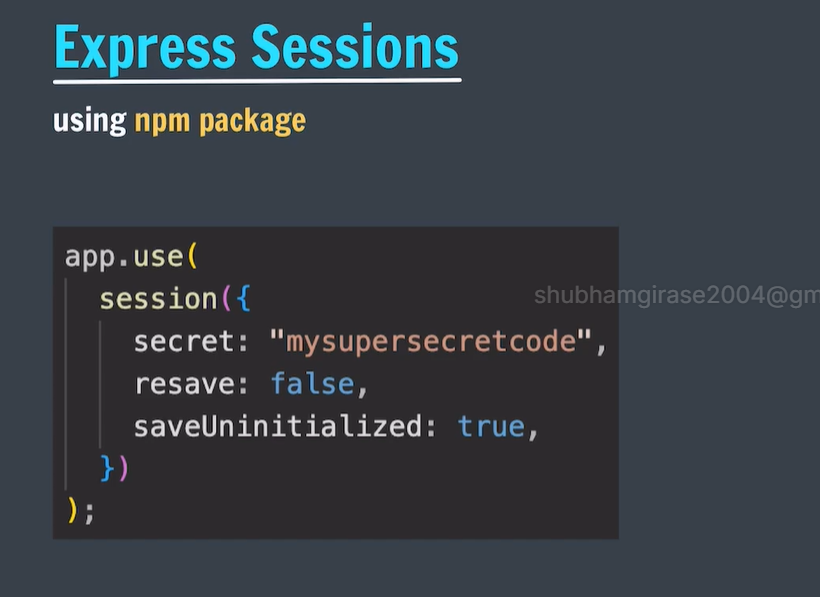
****

**Information is stored in temporary storage and session id is created it is further sent to client. (it makes browser remember our important information such as added items in cart on amazon)**

****

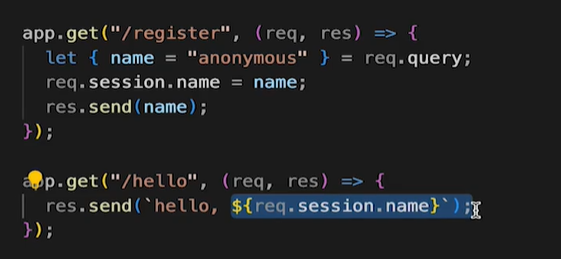
****

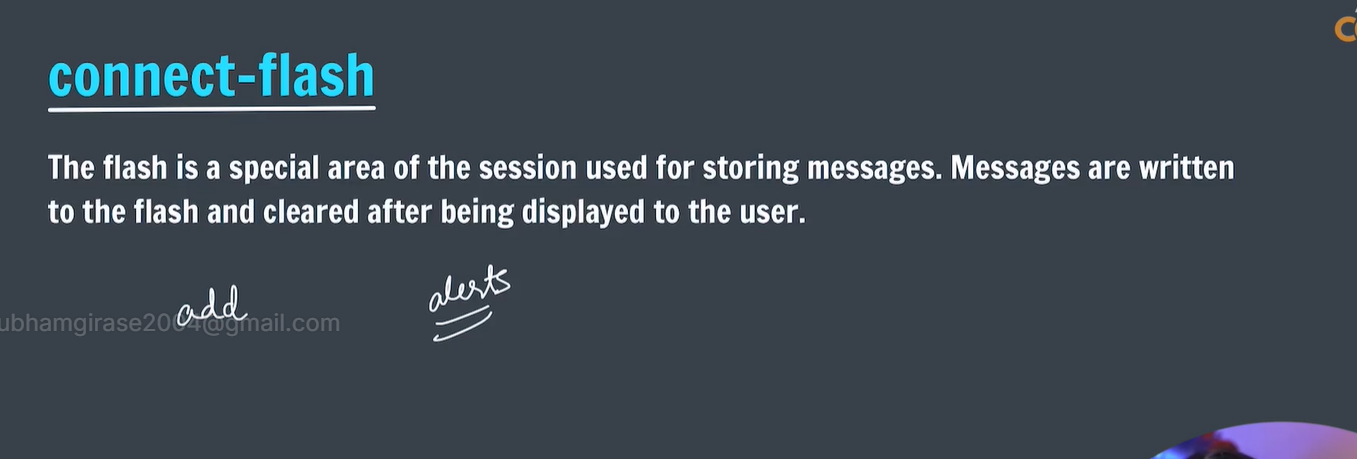
**Header**

****

**Session stores are used as temporary storages in production environments . however we used local memory for local use.**

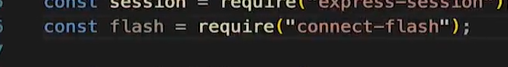
****

**storing and accessing information**

****

**Connect flash is npm package use to pop up messages on web page top whenever we perform particular action.**

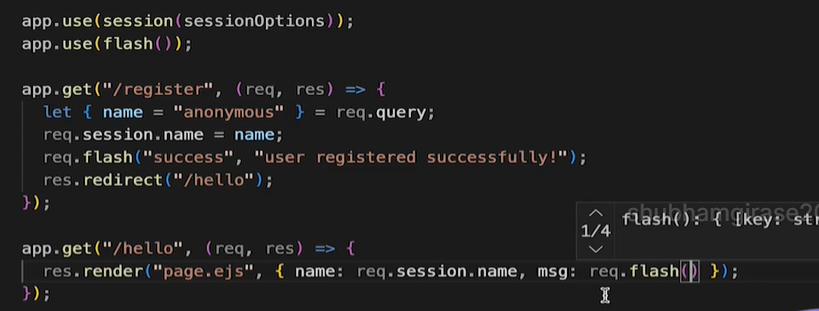
**Npm I connect flash to install it .**

**Require it @ top .  
**

**Usable only with sessions.**

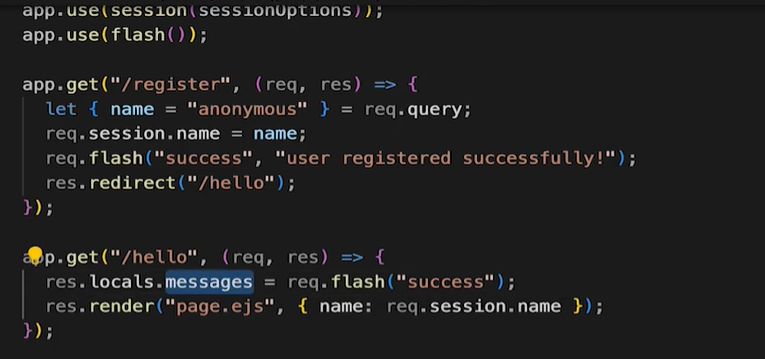
****

**that’s how we write message.**

****

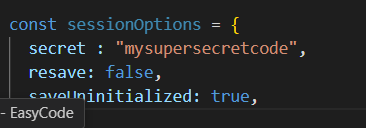
**To use flash**

****

**message got saved in response.local**

**Starting Project..with session and cookies:**

**Installed session and require it**

****

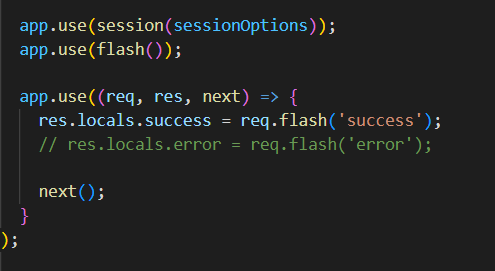
****

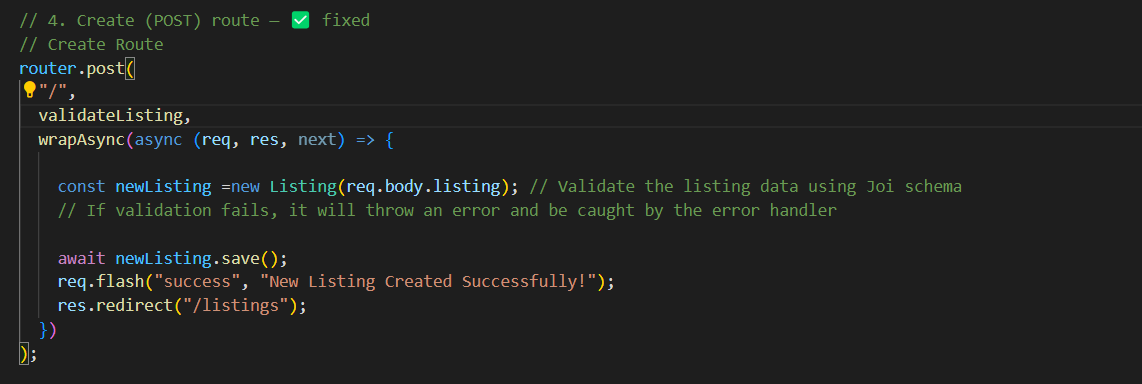
**Flash:**

**Installed by npm I connect-flash**

**Require and**

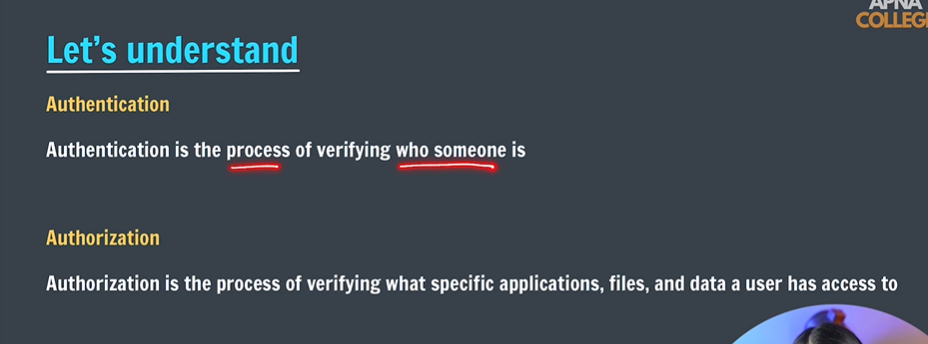
****

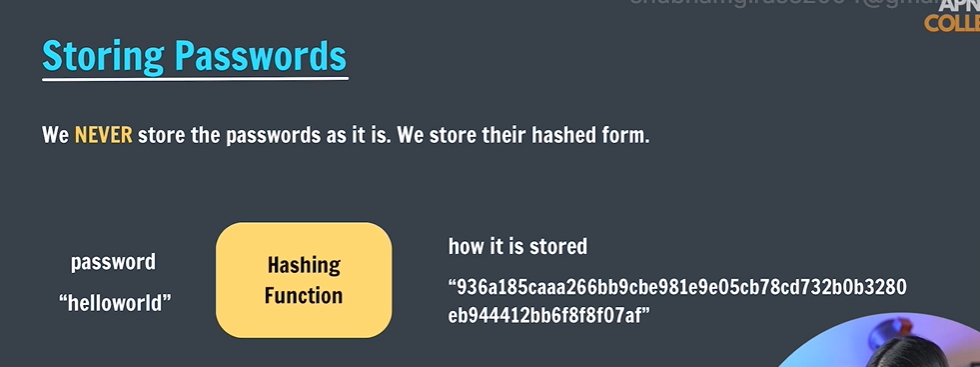
**Defined middleware .** ****app.js

**  
added in index.ejs and displayed on page**

****

**Authetication and authorization …**

****

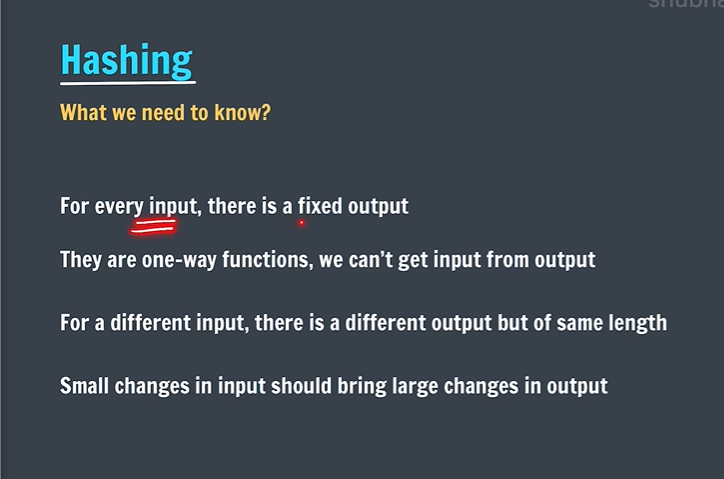
****

**Password wont saved as it is .. it converted to hash form:**

**Reason: 1. Privacy of user**

**2. user might have same passwords for another website too ..it could affect other web if single instance is leaked.**

**Whenever user will try to login . his entered password will be again pass to hashing function and output will be compared to existing str of hash.(if current hash == existing hash). Login successful.**

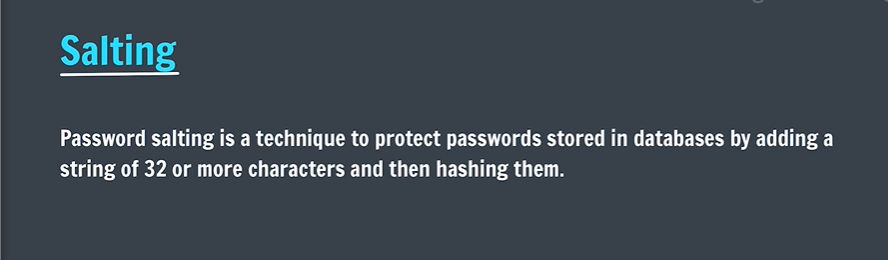
****

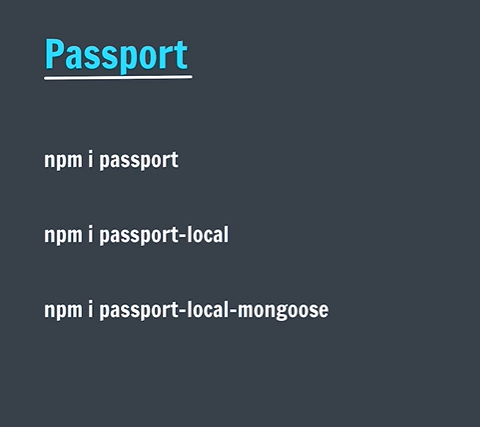
**Ex SHA256**

**MD5**

**CRC**

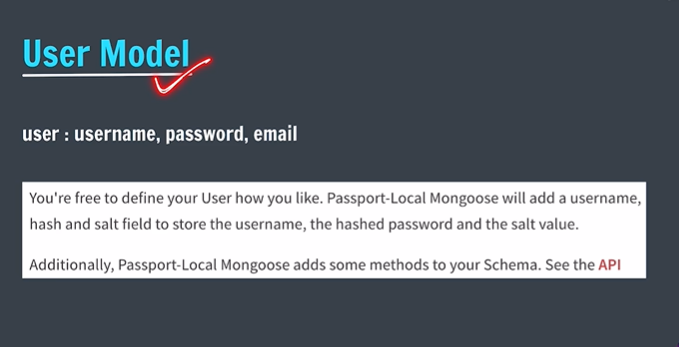
**BCRYPT**

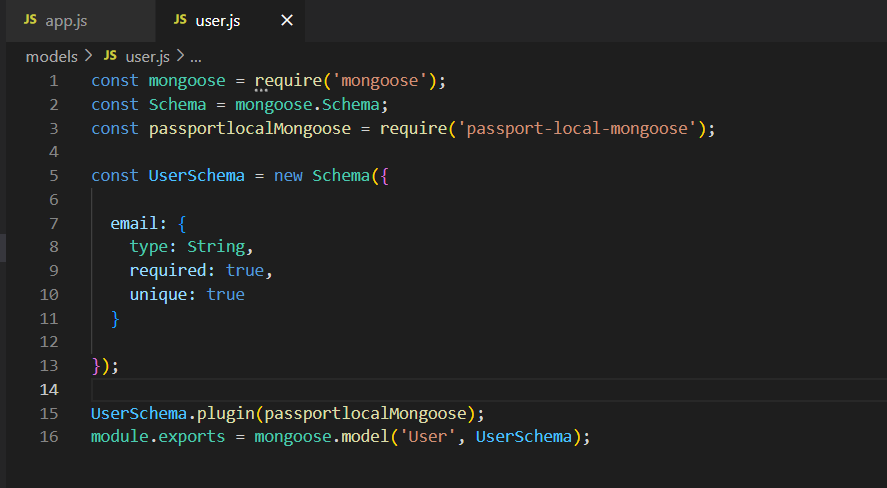
****

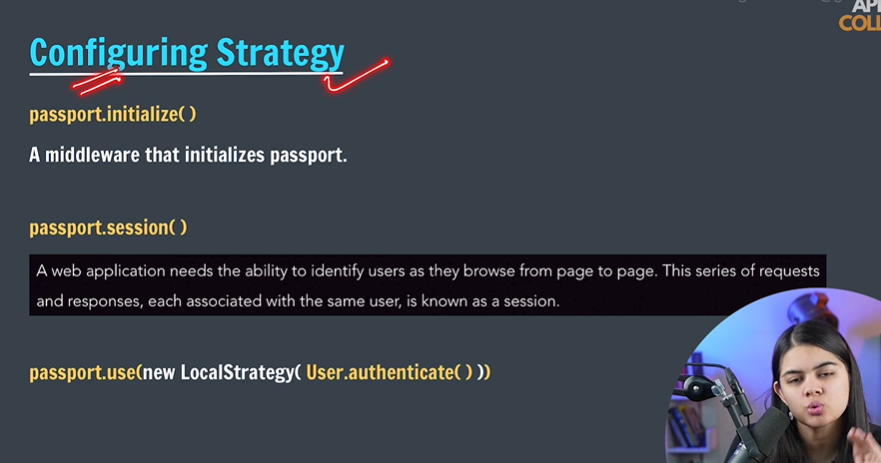
****

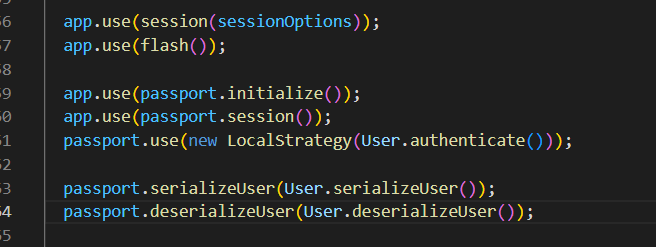
**For project signin/signup**

**Passport is express compatible authentication middleware for node.js.**

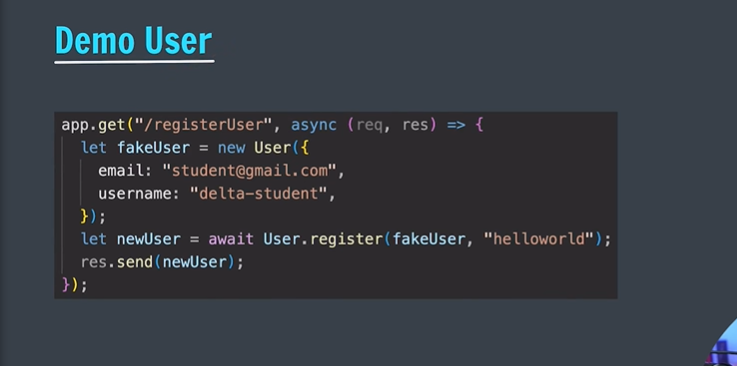
****

**  
no need to explicitly define username and password as passport automatically does that for us .**

****

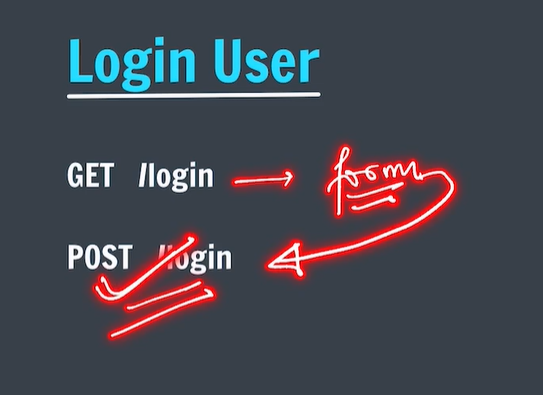
****

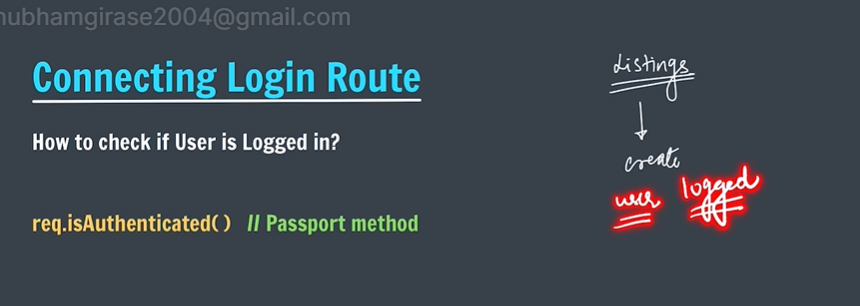
**App.js implementation by passport documentation**

****

**Passport uses pbkdf2 hashing algorithm.**

****

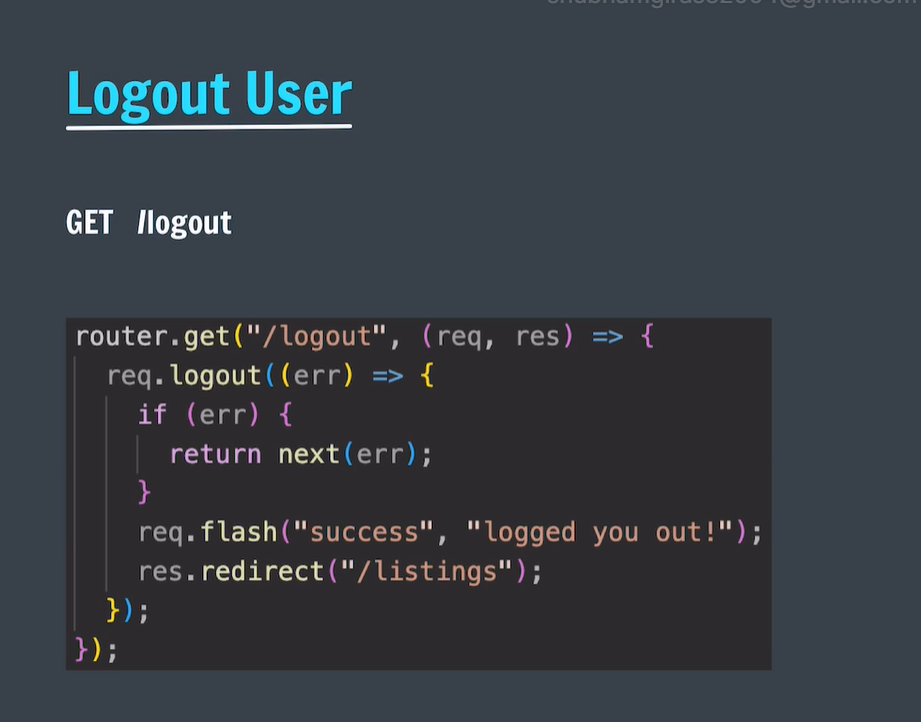
****

****

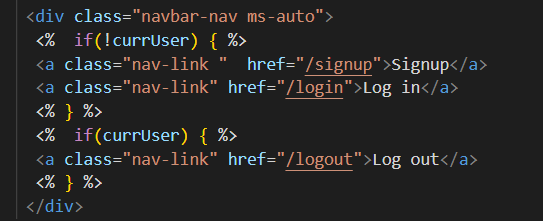
**We r creating a functionality for user authentication before adding new listing.**

**We created middleware.js to ensure user must be logged in before adding,updating or deleting listing.**

**Called it in routes..**

****

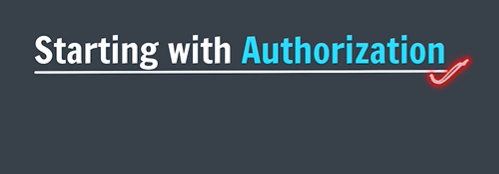
**Adding login ,signup options on Navbar ..**

****

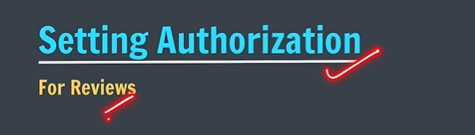
**We cant directly access req.user in ejs .. so we use res.locals ( created curr User in app.js and added local variable here)**

****

****

****

****

****

****

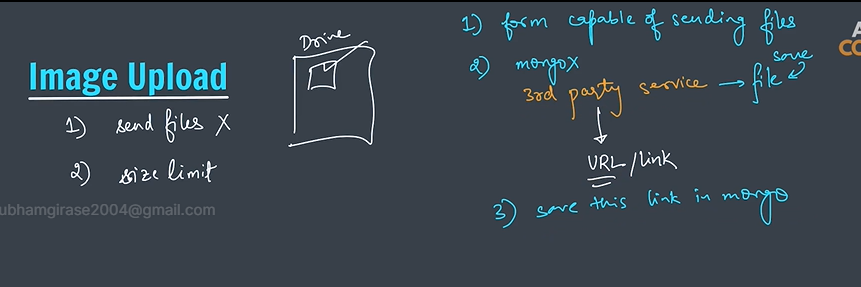
**Mvc is design pattern use to arrange full stack codes.**

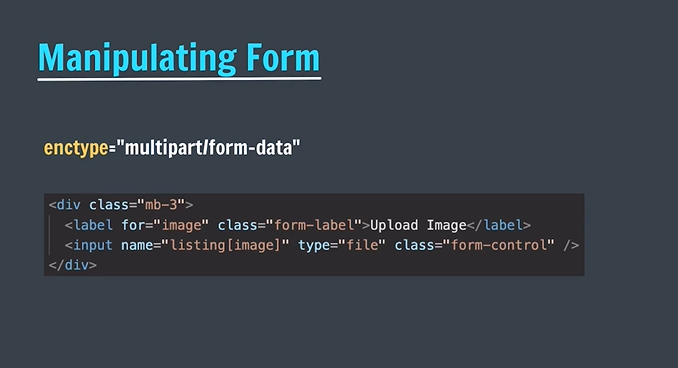
****

**Router.route to write routes in compact form .**

****

**Starabillity property to apply star format ratings..**

**Image upload   
**

****

**Enctype is use to send file type data instead of urlencoded data.**

**We r using multer .. for upload image functionalities**

****

**Used cloudinary .**

**We cant access files directly from file.env ..to peform this we will use dotenv third party service from npm.**

**Installed cloudinary and dotenv dependencies**

**Later**

****

****

****