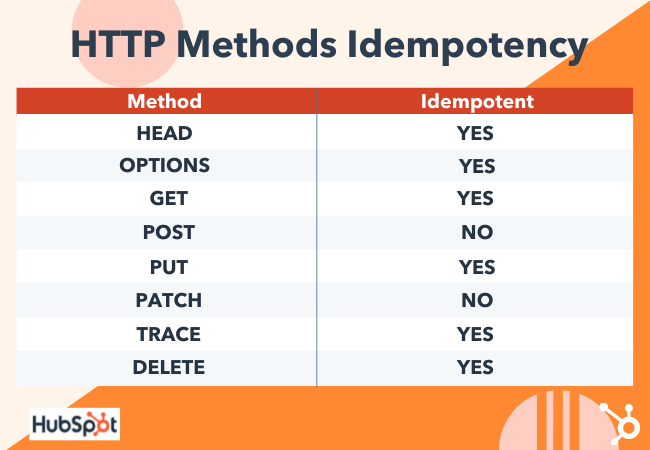
**Backend Development Notes**

1. Introduction:-  
   Link pe click kiya -> chrome -> IP address (Domain Name Resolution ) -> server ip address -> connection estd (TCP) -> communication done  
     
   HTTP Methods :
   1. GET : to fetch the data
   2. PUT : update the data
   3. POST : submit the data
   4. DELETE : remove the data

Idempotent: An HTTP method is **idempotent** if the intended effect on the server of making a single request is the same as the effect of making several identical requests.



**Class- I**

* npm init -y : to create package .json file
* npm i express : to install the express dependicies
* server ko listen kiya at port 3000 but web page pe show hone ke liye hume route lagega
* POST – post karne ke liye pehle req mein data chahiye
* Postman use karke data ko send kiya lekin error destrauctred property s  
   o used the middle ware jo request ki body ka data so use body parse
* BodyParser :   
  const bodyParser = require('body-parser');  
  app.use(bodyParser.json()); Hum ye keh rahe hain ki

1. MongoDB  
    How to link server and mongodb 🡪 Use mongoose (ODM) Library : Object data modeling

**Class – 2**

* File module separate rakho
* Routes -> isme saare ke saare routes rakho ab jab ye route hit hoga toh koi na koi operation perform hogi and who logic pada hoga
* Models : **{structure}**for example car ki properties like weight color route mein hum define karenge ki path kis controller se map hain
* Controller : **{logic}**yesi file jisme exact logic likh hoga ki karna kya hain   
  har ek path controller se map hoga  
  like create , get , post
* Config :- isme function like operation bar bar use hone wali static chize isme rakhenge jab need hogi tab call karlenge
* .env :- file isme hum define karte hain ki konsa variable use karne wale hain like konsa port
* 127.0.0.1 🡪 ye loopback address bhi bolte hain just like localhost
* Ab mongoose ka function banaya and waha process.env.datbase\_url ko diya but ye yaha aaya kaise isko feed karna hoga na   
  so isko use karne ke liye hum use karte hain .env library  
  so npm i dotenv  
  require(‘dotenv’).config() 🡪 isse humara isse humn jo env mein define kiya hoga who load ho jayega process object ke andar
* Ab ek model mein todo.js se ek schema banao and usko export kardo jo controller use karega badme
* Create todo ka controller is like todo type ke object ki entry database mein save karlo
* CreateTodo.js  
  Jab database mein kuch access karna ho toh motive ye hain ki main thread block nahi hona chhaiye so use async await   
  yaha hume data ko fetch kiya and directyly insert kar diya db mein ab for success ek response generate kiya and if some error occurs then usko catch kiya (500 ->internal server)
* Ab route define karna hain sab karliya todo ka route banao   
  ab route means map karna hain controller ke sath so require controller  
  ab isko post ke liye hi banaya hain so post ka likho and export kardo
* Now sab kuch ho gaya ab time for index.js  
  yaha middle ware ki need hain toh parse the data  
  import the routes and set them
* Hum url mein custom define kar sakte hain back part kaisa dikhega   
  isko mount kiya toh localhost || api v1 || createtodo
* Ab db connect ko define karo and call kardo
* Default Route : likhna hoga nahit toh path dena hoga

**Class – 3**

* **a)** its time for get all todos controller but before that we have created mogoose library provides diffeenre type of fuctions like crate, findbyid, find one, findbydeleteandupdate

**a**b muze all todos at one time chihiye the so run the code  const todos = await Todo.find({});const todos = await Todo.find({});

* then api get request mari and all data got liste**d**
* now its time for the single to do item on the basis of id   
  so pehle id ko accesss karliya also route mein likh liya ki hua hain ya nahi
* **ab time for put request updateTodo** toh ek route banaya ab yaha ek function hota hain findbyupdate and id
* **ab time for delteTodo**

**Class -4**

* Blog Page Routes  
  Post, Get Posts , Create Comment , Like post , unlike post  
  so pehle to sara basic basic setup karo like database connet folder strucyure and controllers and modles
* Then for post it will have title , desc, likes array , comment array   
  acchi pratice yeh hain ki likes mein sarri info nahi dalenge sirf id dalenge and baki chize object mein dusri jagah likhenge
* Ab commnetmodle ko likhete hain toh yaha route lihne ke baad schemea define kiya but here referencing is used so new syantax is there for mongoose

**Backend II & Class 2**

* Pehle toh data lelo req.body se email and password
* Validat karo ki empty ya phir format thik hain kya nahi
* If user exist -> call the db
* If not exist then return error and first sign up
* Then validate password if it is right then yaha se game chalu hoga

JWT Tokens implementation:-

* Sign functin isme payload, secret, options, call back function
* To verify password bcrypt libray is there
* Ab login karna hain toh jwt token lagega and uko create karna hoga
* Ab ye hua sab token create then
* Response mein user ko send krna hain but ek gadbad hain object mein toh password bhi hain toh password nikal do
* Ab res ko as a cookie store karo
* 3 parameter cookie na name , cookie ka data , options

Middlewares :-

Ye ek utility hain jo intercept karti ongoin route mein if any misbehave is there student trying to access admin route or vice versa

3 Type of middlware:

* It will check autheticity
* Ab pehle toh user.js mein import kiya ab procted route define karne hain to usme param change honge (path, middlwares, handlers)
* Ab actual middle ware likhte hain uskeliy auth .js mein jao and ab 3 parameters   
  req , res and next (where to go now)
* Ab token ko nikalo 3 ways hai from header, from token , from cookie
* Admin and student middleware are used for the authorization
* Hw ->cookie parser kya hota hain

**Backend Class II & Class 3**

* BodyParse : express.json() : req ki body mein json parse karne ke liye hum ise use karte hain
* Cookie Parser :-
* 3 ways to access the token
* Most secure way is of header ( token is send using bearer token )
* Body ke liye req.body kiya but uske liye isko bodyParser lagta hain
* Cookie -> req.cookie.token -> iske liye cookie parser lagta hain
* Header -> key – Authorization value -> “Bearer <Token Key> “
* Ab ek ek karke try kare dekho cookie delte kardo toh bodyse gaayi  
  agar body undefined kardi toh cookie se gayi
* Expirtion time khtm hote hi relogin karna hota hain
* Cookie hijacking , token hijacking hw

**Backend III + Class I**

* Image and video upload karnihain  
  cloudinary ke through acess karni hain
* Iskko pehle import karlo then use .config() fuction isme 3 chize pass karni hoti hain
* Kya express file se inteact kar sakta hain so that we have to use  
  multer , expresss file upload , etc  
  so import express file upload -> ye server pe data dalta hain hain  
  cloudinary ->pehle server pe then uske server pe dalthahain and apne server se delete kar deta hain
* Ab models mein datamodeling karte hain then controllers mein logic likhte hain
* We use files hierachy using req.files.files
* \_\_dirname -> repressents the current directory
* Ab cloudinary pe file kko upload karna hain so hum ek function banayenge pehle toh validation lagayenge then using cloudinary functions file ko upload kar denge i
* Video Upload :-> data fetch , validation , upload to cloud , db mein entry , response successful

**Mega Backend Class**

* Pehle pura rough structure create karlo and then sab map karo kiske under kya aata hain
* Toh ab first config folder mein database.js ko setup kiya
* Now User route likha hain so uska schma define kiya
* Ab same yese 8 modles banao
* Jo otp hain uska code pre middleware mein likhna hain as humne abhi tak db mein entry nahi ki hain ye sara mamala db ke bhaar chaal raha hain so we are going to use **pre-model**
* Ab mail send karne ki service Utils mein banao file kko
* To mail karne ki liye hum transoporter ka use karte hain  
  uske liya hume host, auth declare kiya and send mail functtion use kiya to send the mails
* Code ko schema ke baad and model ke pehle likho
* Kabhi bhi code mein agar console.error() likh te hain toh waha ek stringapne aap se diya karo for good pratice
* Ab hume controller liknnahina sfor signup , login , auth
* And one more for reset password isme ek link send hoga user ko
* User dicebear to create image with dynamic input text

**Mega Backed Class 3**

* Sabse pehle tag create karna hain jisme course bana skate like create post
* Section like remove update add section
* Then subsection lilke add delete remove
* Ab create tab banao usme aayge  
  name ,desc -> validation -> tag entry in db -> return response
* Get all tagas  
  user find keyword hamare pas koi parameter thoh hain nahi jaha bata sake ki toh all entries le aao but jisme data ho present
* Ab course ka handler function banate hian usem create karte aana chahilye and fetch all courses
* Create course keliye  
  data fetch -> file fetch -> validation lagadi ->instructor ->tags(postman) -> image(cloudinary) -> add course entry in user schema -> add course entry in tags -> db mein entry kardi ->return response
* Agar hum course create kar rahe hain toh hum pehlse se hi login hain toh ab hum id chahiyye so who aaye gi payload se  
  humne account type ko hata diya ab bolo kya karenge..so Id access karne ke liye db ko request marni hi padegi  
  instructor lagega as course mein hum dalte hian instructor kon hain

**Mega Backed Class -4**

* Sections and subsections flow:-  
  pehle toh course se section create hoga then again wo created section course mein update hoga and then subsection banao usme store huyi hain subsection ki ID\
* Subsection banaya
* Profile section banaya
* Db mein entry karne ke two ways using create and save

**Backend Class 5**

* Payment gate way integration ab agar razor pay ki api ko integrate karna hain toh orders create karna padenge { Order api } use karni hogi
* Joh amount hogi who \* 100 bhejo tabhi who correct dikhega   
  currency mandatory
* Store some fields in the sever  
  payment\_id , order\_id , razorpay\_signature
* Verify payment signature   
  order -> created , attempted , paid
* Created -> authorized -> captured
* Payment integration : 2 step process hain  
  1. Capture payment :- payment initial , order create kardo baki razro pay dekh lega  
  ab agar tume order capture karliya then user payment kardega razp pay ko then rp bank se interact karegea then rp se website ko kaise pata chlegaki payment ho gyai hain course dena hain so there is conceptcalled  **webhook** isko use kiya jata hain   
  jab successful transaction hoga toh api route ko hit kardo and yeconfirm karne ke liye ki ye rp hi hain so ek secret key dedi webhook ko
* Config mein razorpay setup karte hain
* Toh to encrypt the webhook use crypto in build module and sha ( secure hashing algorithm ) use kana hain and ek hashing functin create karna hoga  
  hmap works on top of hasing algo
* Jab hashing algo ko run karte hai ho output aata hain usko bolte hain digest (hexadecimal format)
* Ab sab sort hain user ne payment kardiya , razor pay ne payment authorize bhi kar diya ab output dena hain user ko i.e course mein enroll karna hain but kaise as mere pass toh koi id hain hi nahi req.body mein aayi hain toh ye kaha se lau  
  ab yaad hain notes nam ki chiz isme jan bhuz ke userid pass ki thi
* Average Rating find karni hain toh phele kya access kare
  + 1. Kisko deni hain course toh couse id ko accesskaro
    2. Phele toh entries ko access karo db using db if they are in string convert them to object
    3. Group karlo sari entries ko and average karlo kis basis pe karna hain pata nahi toh null set karlo
* Category page details :- fetch karo different courses

**Meaga FrontEnd Development**

* First of all humne concurrently package dowload kiya then server and src concurrently run kiya
* To use icon we have used react icons package