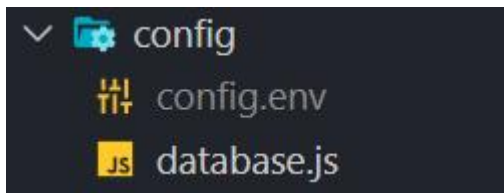


# FOLDER STRUCTURE

## Config

**Config.env** config files to store all env variables

**Database.js** database file contain connectDB function to connect database



## Controllers

**CourseController.js** contains all controllers related to the course & watcher

**UserController.js** contains all controllers related to the user & watcher

**PaymentController.js** contains all controllers related to the payment & subscription

**OtherController.js** contains remaining controllers like contact, course request, dashboardstats.



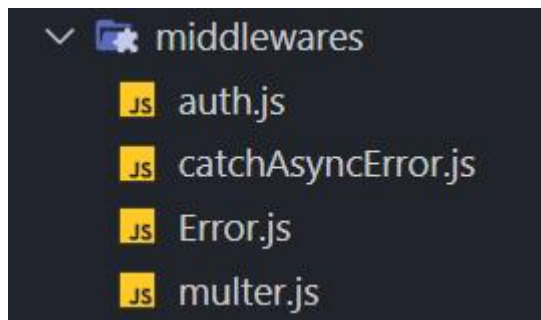
# Middlewares

**Auth.js** contains middlewares related to authenticated

**CatchAsyncError.js** contains catchAsyncError middlewares

**Multer.js** contains multer middlewares

**Error.js** contains custom error middleware



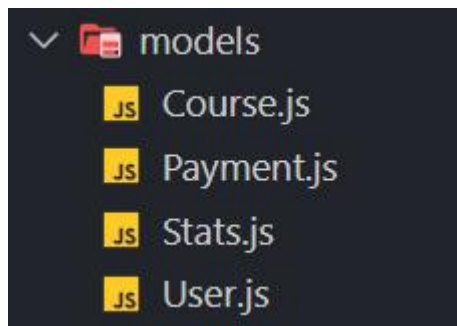
# Models

**Course.js** course schema & model

**Payment.js** payment schema & model

**Stats.js** stats schema & model **User.js**

user schema & model



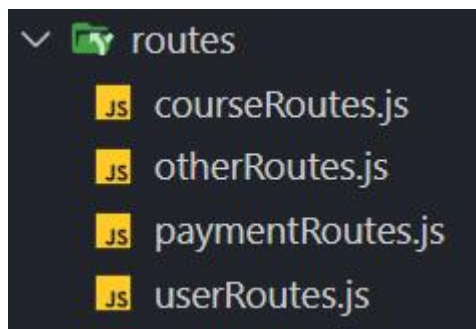
# Routes

**CourseRoutes.js** routes related to course

**PaymentRoutes.js** routes related to payment

**OtherRoutes.js** remaining routes

**UserRoutes.js** routes related to user



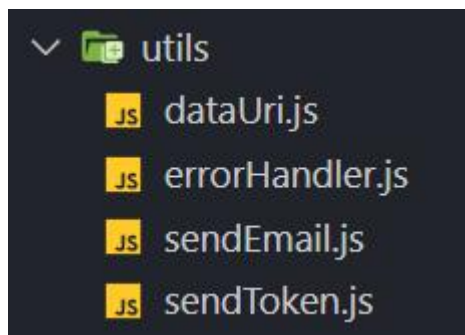
# Utils

**DataUri.js** contains getDataUri function

**ErrorHandler.js** custom error handler class

**SendEmail.js** contains sendEmail function with nodemailer

**SendToken.js** contains sendToken function to set jwt token in cookie



App.js  
Sever.js

# DEPENDENCIES

**bcrypt**  
**cloudinary**  
**cookieparser**  
**cors**  
**datauri**  
**dotenv**  
**express**  
**jsonwebtoken**  
**mongoose**  
**multer**  
**node-cron**  
**nodemailer**  
**razorpay**  
**validator**

```
"dependencies": {  
  "bcrypt": "^5.0.1",  
  "clouddinary": "^1.30.0",  
  "cookie-parser": "^1.4.6",  
  "cors": "^2.8.5",  
  "datauri": "^4.1.0",  
  "dotenv": "^16.0.1",  
  "express": "^4.18.1",  
  "jsonwebtoken": "^8.5.1",  
  "mongoose": "^6.4.4",  
  "multer": "^1.4.5-lts.1",  
  "node-cron": "^3.0.1",  
  "nodemailer": "^6.7.7",  
  "razorpay": "^2.8.2",  
  "validator": "^13.7.0"  
}
```

# SEVER

**Cloudinary** Set Cloudinary configuration

**Razorpay** Create instance of Razorpay

**ConnectDB** Connect database

**NodeCron** Set a schedule to create new Stat each month

**Listen** Listen to a port

# APP

**Config** configure dotenv

**UseMiddlewares** express.json, expres.urlencoded, cookieParser, cors, ErrorMiddleware

**UseRoutes** user, course, payment, other

# API's

# User

**Register** To register a new user

**Login** To login

**Logout** To logout

**UpdateProfile** To update name & email

**UpdateProfilePicture** To change profile photo

**ChangePassword** To change password

**ForgetPassword** To send reset password link

**ResetPassword** To reset password with token

**AddToPlaylist** To add a course to the playlist

**RemoveFromPlaylist** To remove a course from the playlist

**UpdateUserRole** To change user to admin

**GetAllUsers** To get all available users

**GetMyProfile** To get logged in user data

**DeleteUser** To delete a User

**DeleteMyProfile** To delete my profile

# Course

**CreateCourse** To create a new course

**GetAllCourses** To get all courses

**AddLectures** To add lecture in course

**GetCourseLectures** To lectures of a course

**DeleteLecture** To delete lecture in course

**DeleteCourse** To delete Course in course

## Payment

**BuySubscription** To create/start a subscription

**PaymentVerification** This is the callback api to verify payment & save in database

**GetRazorPayKey** To get RazorPay API key

**CancelSubscription** To Subscription of a user

## Other

**Contact** To send contact form mail **CourseRequest**

To send course request form email

**GetDashboardStats** To get Dashboard Stats

# Middlewares



## Auth

**IsAuthenticated** To make APIs accessible by logged in User

**AuthorizeSubscribers** To make APIs accessible only by Subscribers

**AuthorizeAdmin** To make APIs accessible only by Admin

## CatchAsyncError

**CatchAsyncError** To use API in try-catch block

## Multer

**SingleUpload** To get access to req.file

## Error

**ErrorMiddleware** Custom error handling middleware

# UTILS

## DataURI

**GetDataUri** To get uri of file received from req.file

## ErrorHandler

**ErrorHandler** Creating a custom error class extended from default error class

## SendEmail

**SendEmail** Function to send email using nodemailer

## SendToken

**SendToken** Function to set cookie in header

# MODELS

## User

**Name** type, required

**Email** type, required, unique, validate

**Password** type, required, minLength, select

**Role** type, enum, default

**Subscription** id, status

**Avatar** public\_id, url

**Playlist** [ courseId, poster ]

**CreatedAt** type, default

**ResetPasswordToken** type

**ResetPasswordExpire** type

## User Methods

**PreSave** hash password before saving user

**GetJWTToken** To generate jwt token

**ComparePassword** To compare hashed password and given password

**GetResetToken** To generate token for resetting password

## Course

**Title** type, required, minLength, maxLength

**Description** type, required, minLength

**Lectures** title,description,videos { public\_id,url }

**Poster** public\_id, url

**Views** type, default

**NumOfVideos** type, default

**Category** type, required

**CreatedBy** type, required

**CreatedAt** type, default

## Payment

**Razorpay\_Subscription\_Id** type, required

**Razorpay\_Payment\_Id** type, required

**Razorpay\_Signature** type, required

**CreatedAt** type, default

## Stats

**User** type, default

**Subscription** type, default

**Views** type, default

**CreatedAt** type, default

# CONFIG VARIABLE

**PORT** Port to listen

**PLAN\_ID** Subscription plan id created from razor pay

**RAZORPAY\_API\_KEY** Razorpay API key

**RAZORPAY\_API\_SECRET** Razorpay API Secret

**JWT\_SECRET** any random secret for jwt

**FRONTEND\_URL** frontend hosting address **MONGO\_URI**

database URI **CLOUDINARY\_CLIENT\_NAME** Cloudinary

username **CLOUDINARY\_CLIENT\_API** Cloudinary API

key **CLOUDINARY\_CLIENT\_SECRET** Cloudinary API

secret

**RESET\_PASSWORD\_SECRET** any random secret for password token

**SMTP\_HOST** Email service host

**SMTP\_PORT** Email service port

**SMTP\_USER** Email service userId

**SMTP\_PASS** Email service password

**MY\_MAIL** Mail to receive contact form & request form data

**REFUND\_DAYS** Refund applicable days