

deploy your server with easy step

Step-01: go to your index.js and organize your code in this order:

1. All your required file

```
1 const express = require('express')
2 require('dotenv').config()
3 const cors = require('cors')
4 const { MongoClient, ServerApiVersion, ObjectId } = require('mongodb')
5 const jwt = require('jsonwebtoken')
6 const cookieParser = require('cookie-parser')
```

2. Your creation:(app, port etc)

```
1
2 const app = express()
3
4 const port = process.env.PORT || 5000
5 const jwtSecret = process.env.ACCESS_TOKEN_SECRET
6
```

3. All you middlewares:

```
1 app.use(  
2   cors({  
3     origin: ['http://localhost:5173', 'https://enmmmedia-19300.web.app'],  
4     credentials: true,  
5   }),  
6 )  
7 app.use(express.json())  
8 app.use(cookieParser())  
9  
10 // middleware  
11 const verifyToken = async (req, res, next) => {  
12   const token = req.cookies?.token  
13  
14   if (!token) {  
15     return res.status(401).send('Unauthorized access')  
16   }  
17   jwt.verify(token, jwtSecret, (error, decoded) => {  
18     if (error) {  
19       return res.status(403).send({  
20         message: 'forbidden access',  
21       })  
22     }  
23     req.decoded = decoded  
24     next()  
25   })  
26 }  
27
```

4. Database info(uri, client, database connection & your collections)

```
1  
2 const uri = `mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASS}@cluster0.s9x13go.mongodb.net/?retryWrites=true&w=majority`  
3  
4 const client = new MongoClient(uri, {  
5   serverApi: {  
6     version: ServerApiVersion.v1,  
7     strict: true,  
8     deprecationErrors: true,  
9   },  
10 })  
11  
12 const dbConnect = async () => {  
13   try {  
14     client.connect()  
15     console.log('DB Connected Successfully✅')  
16   } catch (error) {  
17     console.log(error.name, error.message)  
18   }  
19 }  
20 dbConnect()  
21  
22 // all database collection  
23 const db = client.db('movieDB')  
24 const brandsCollection = db.collection('brands')  
25 const moviesCollection = db.collection('movies')  
26 const cartCollection = db.collection('cart')  
27 const userCollection = db.collection('user')  
28
```

5. Default route/api endpoint then all of your routes/api endpoint:

```
1 // default route
2 app.get('/', (req, res) => {
3   res.send(
4     "<h2 style='text-align: center; margin-top: 1rem;'>MovieDB Server Running ▶ </h2>",
5   )
6 })
7
8 /* Brands Route */
9 app.get('/brands', async (req, res) => {
10   try {
11     const brands = await brandsCollection.find({}).toArray()
12     res.send({
13       status: true,
14       data: brands,
15     })
16   } catch (error) {
17     res.send({
18       status: true,
19       error: error.message,
20     })
21   }
22 })
23
```

6. App listen

```
1
2 app.listen(port, () => {
3   console.log(`DB Server Running on Port:${ port } 🚀`.cyan.bold)
4 })
5
```

Step-02: create a vercel.json file and paste the following code



```
1  {
2      "version": 2,
3      "builds": [
4          {
5              "src": "index.js",
6              "use": "@vercel/node"
7          }
8      ],
9      "routes": [
10         {
11             "src": "/*",
12             "dest": "index.js"
13         }
14     ]
15 }
16
```

Step-03: open terminal write `vercel --prod`.

Step-04: after the deployment is complete add all your environment variables into your vercel project.

Then check all of your routes working correctly or not.