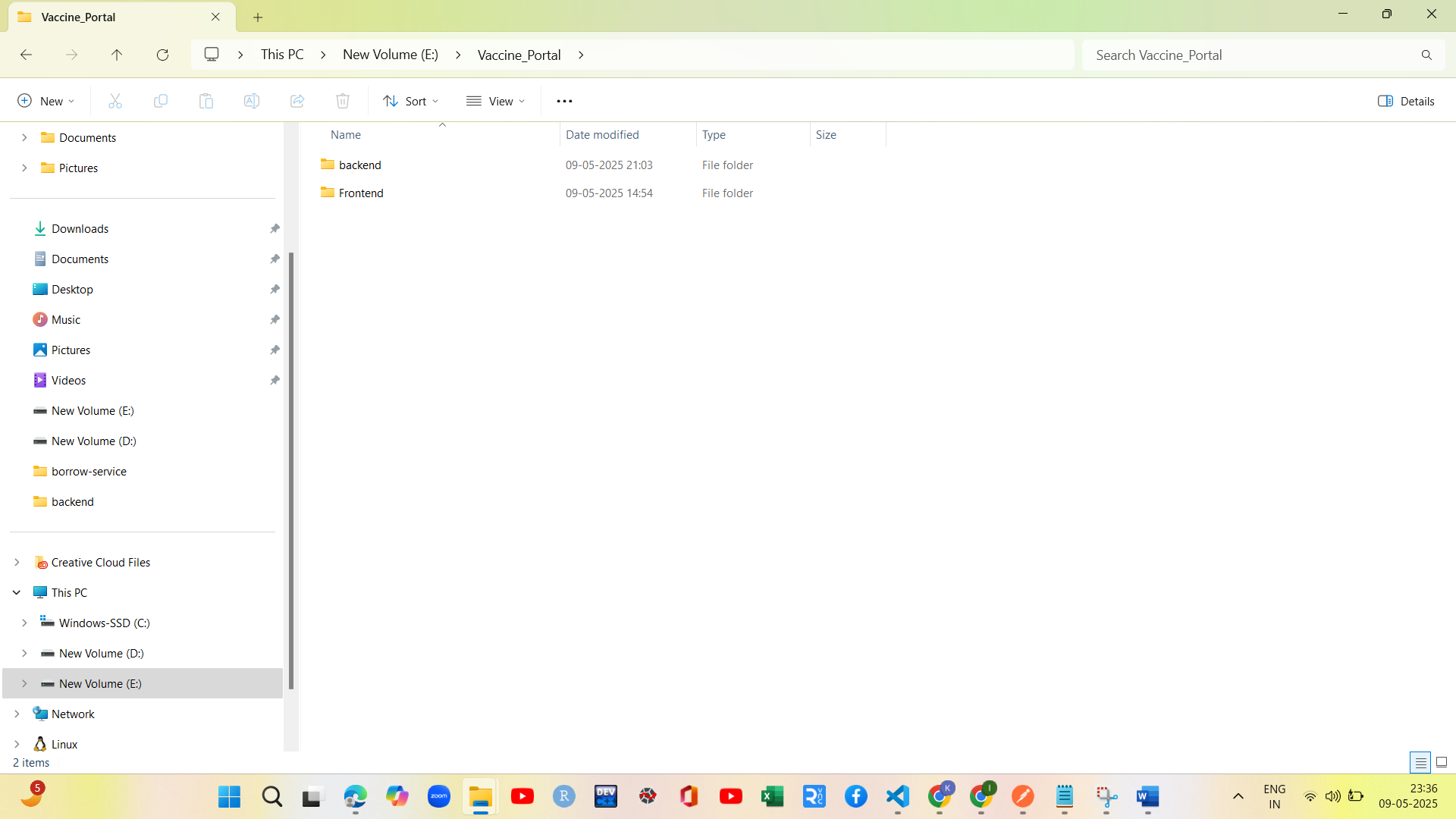
SCHOOL VACCINATION PORTAL consists of both Frontend and Backend Development

**Step1:-** Firstly I have created a folder name Vaccine Portal

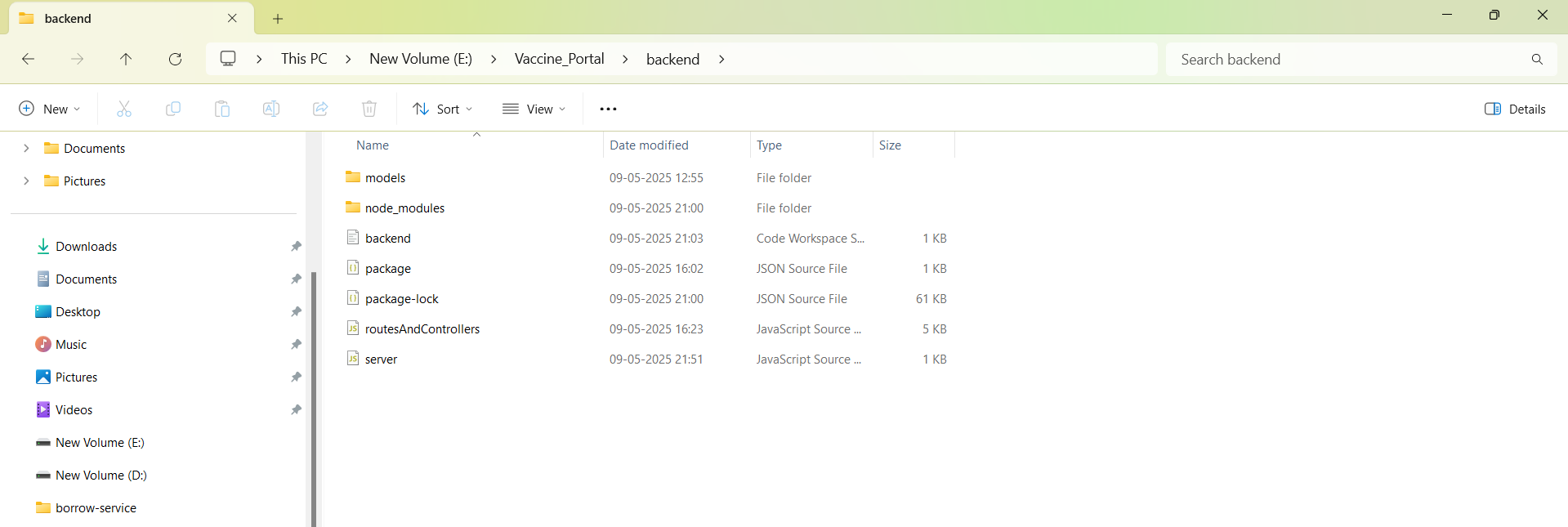
In this I have created two sub folders Frontend and backend. I have frontend folder to handle client side requirements and Backend folder to handle Server Side logic.



I had used following Techstack in backend development

| Layer | Technology |
| --- | --- |
| Runtime | Node.js |
| Web Framework | Express.js |
| Database | MongoDB (likely, based on JSON-style API and schema-less payloads) |
| ORM/ODM | Mongoose (if MongoDB is used) |
| API Format | RESTful APIs |
| Server Port | Running on localhost:3000 (default for Express.js dev servers) |

This is my project structure for backend application in Vaccination Portal



In this created three models Student,Vaccination Drive and Vaccination record

**Student Schema**

Defines individual students eligible for vaccination drives.

* name (String, Required): Full name of the student.
* className (String, Required): The class in which the student is currently enrolled (e.g., "LKG", "5th").
* rollNumber (String, Required, Unique): A unique identifier for the student within their class.
* vaccinated (Array of Objects):
  + vaccineName (String): Name of the vaccine administered.
  + date (Date): Date on which the student was vaccinated.

**Vaccination Drive Schema**

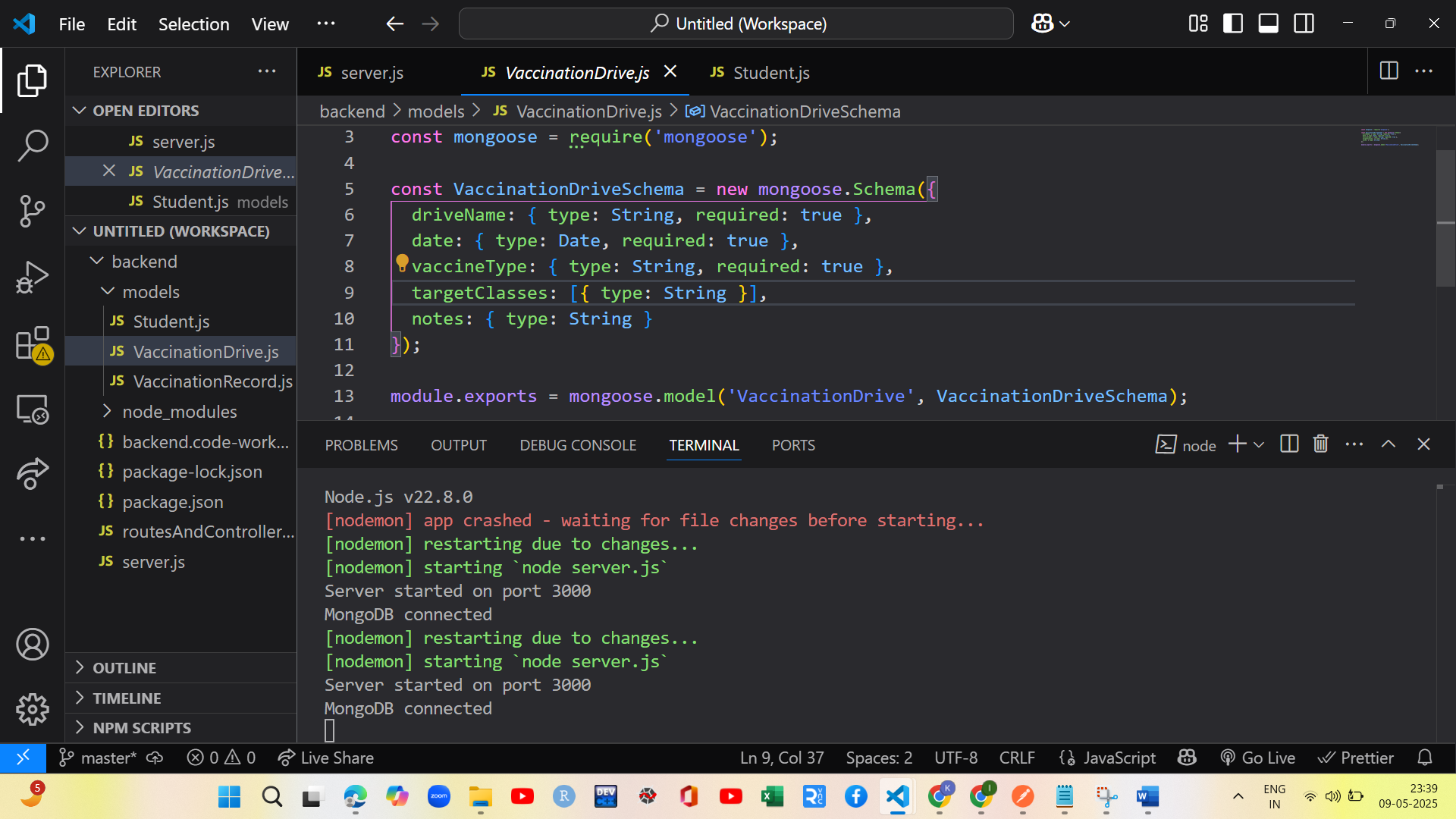
Stores the details of each vaccination campaign initiated by the school.

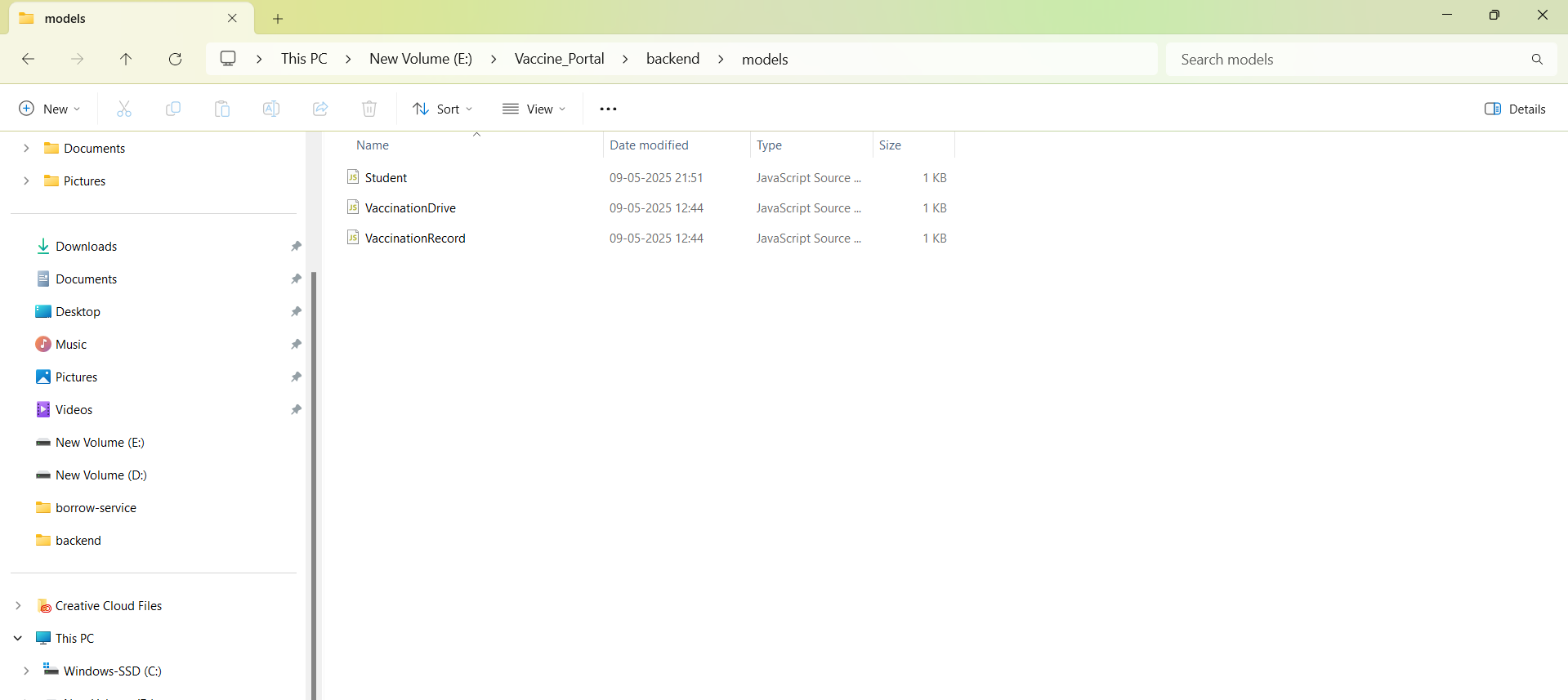
* driveName (String, Required): The official name/title of the vaccination drive.
* date (Date, Required): The date the drive will be conducted.
* vaccineType (String, Required): The type of vaccine to be administered (e.g., "Polio", "MMR").
* targetClasses (Array of Strings): List of class names that are targeted in the drive.
* notes (String, Optional): Any additional information or instructions regarding the drive.

**Vaccination Record Schema**

Captures the status of each student's participation in a specific vaccination drive.

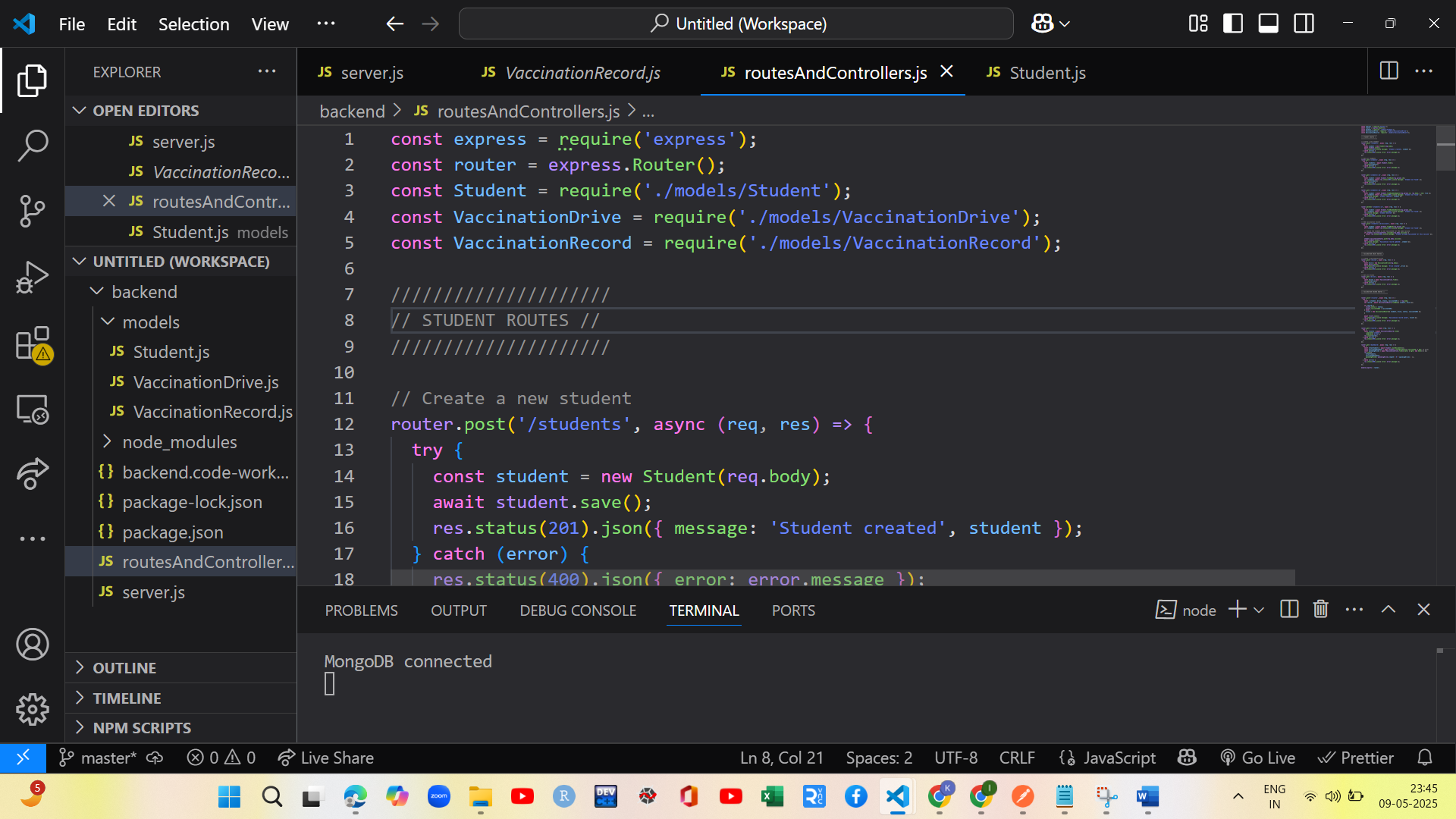
* student (ObjectId, Required): Reference to a student document (Student collection).
* drive (ObjectId, Required): Reference to a vaccination drive (VaccinationDrive collection).
* status (String): Indicates if the student is already Vaccinated or still Pending. Default is Pending.
* vaccinatedOn (Date): The actual date the student received the vaccine, if applicable.



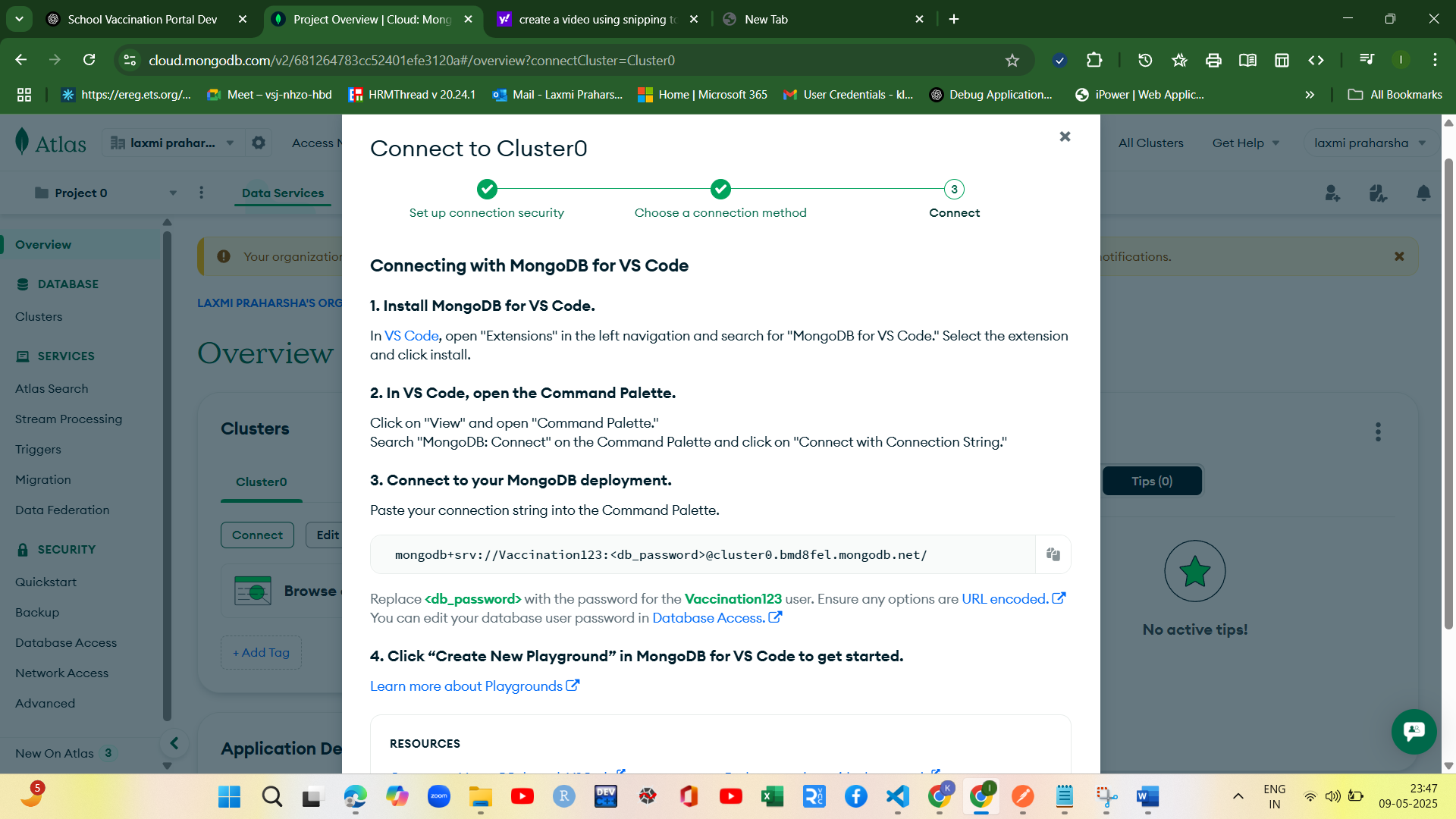


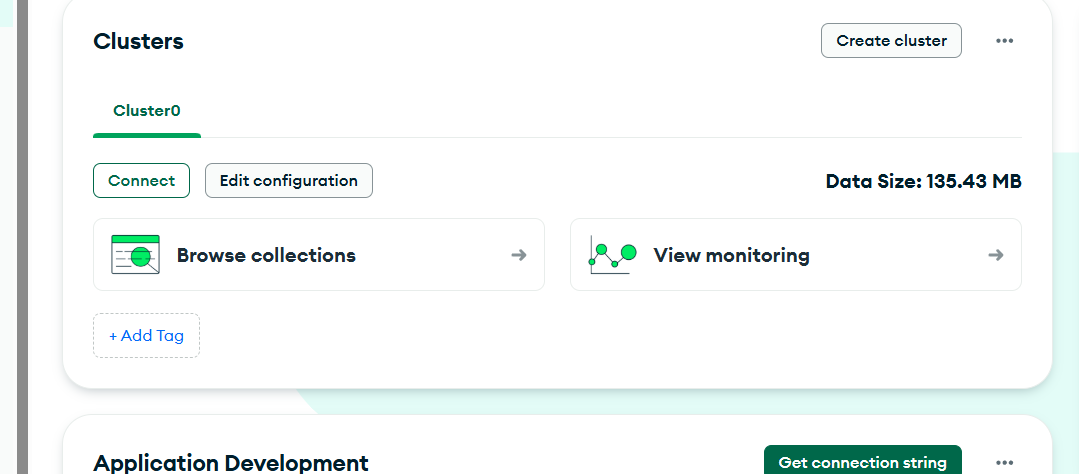
Now we are routesandControllers.js files it defines Vaccinationdrive routes,Student routes and Dasboard routes I had used post and get request.

Post:To add a new record and get is used to fetch all records.

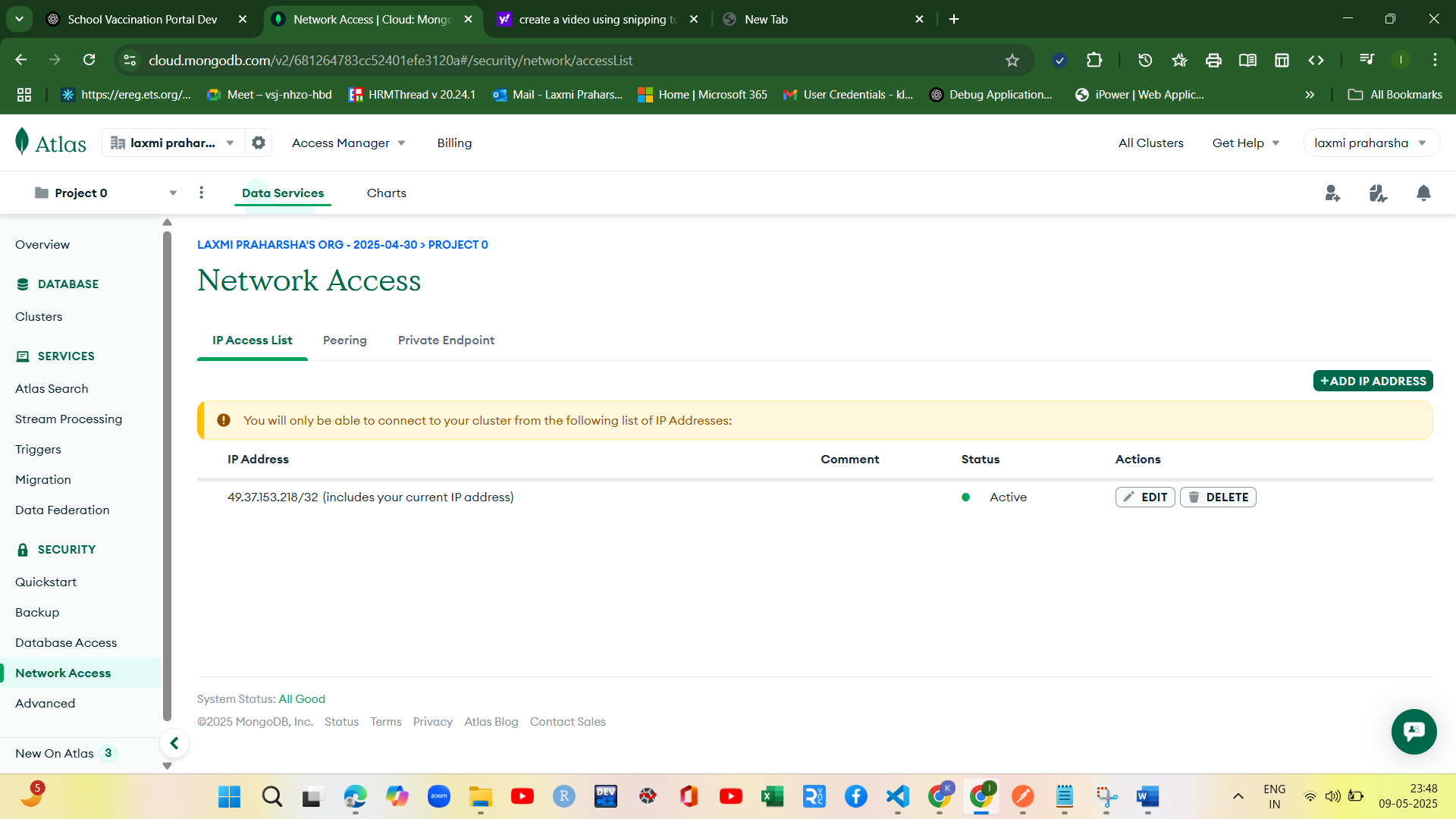


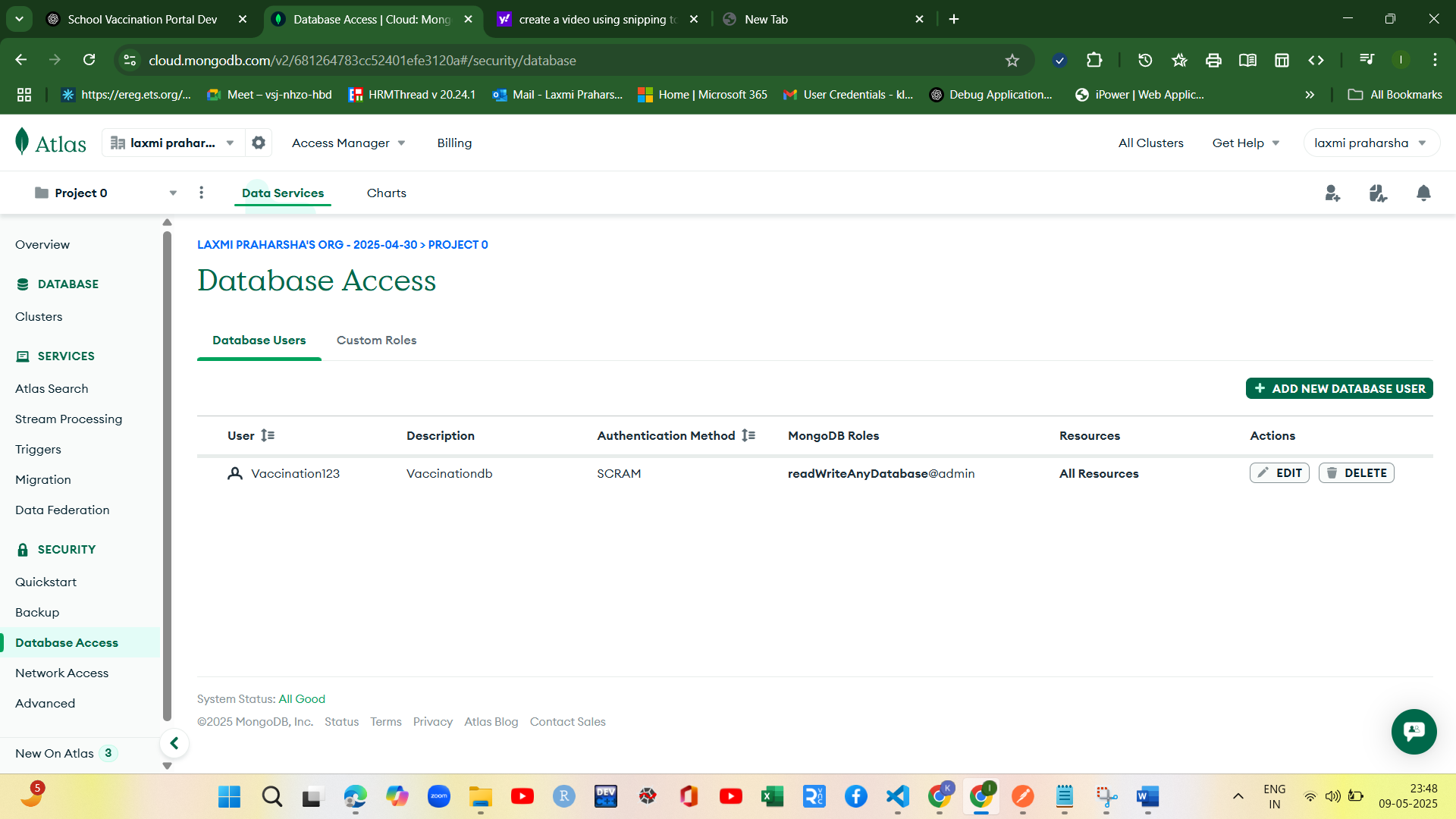
I have created a cluster in mongodb and connected using mongodb





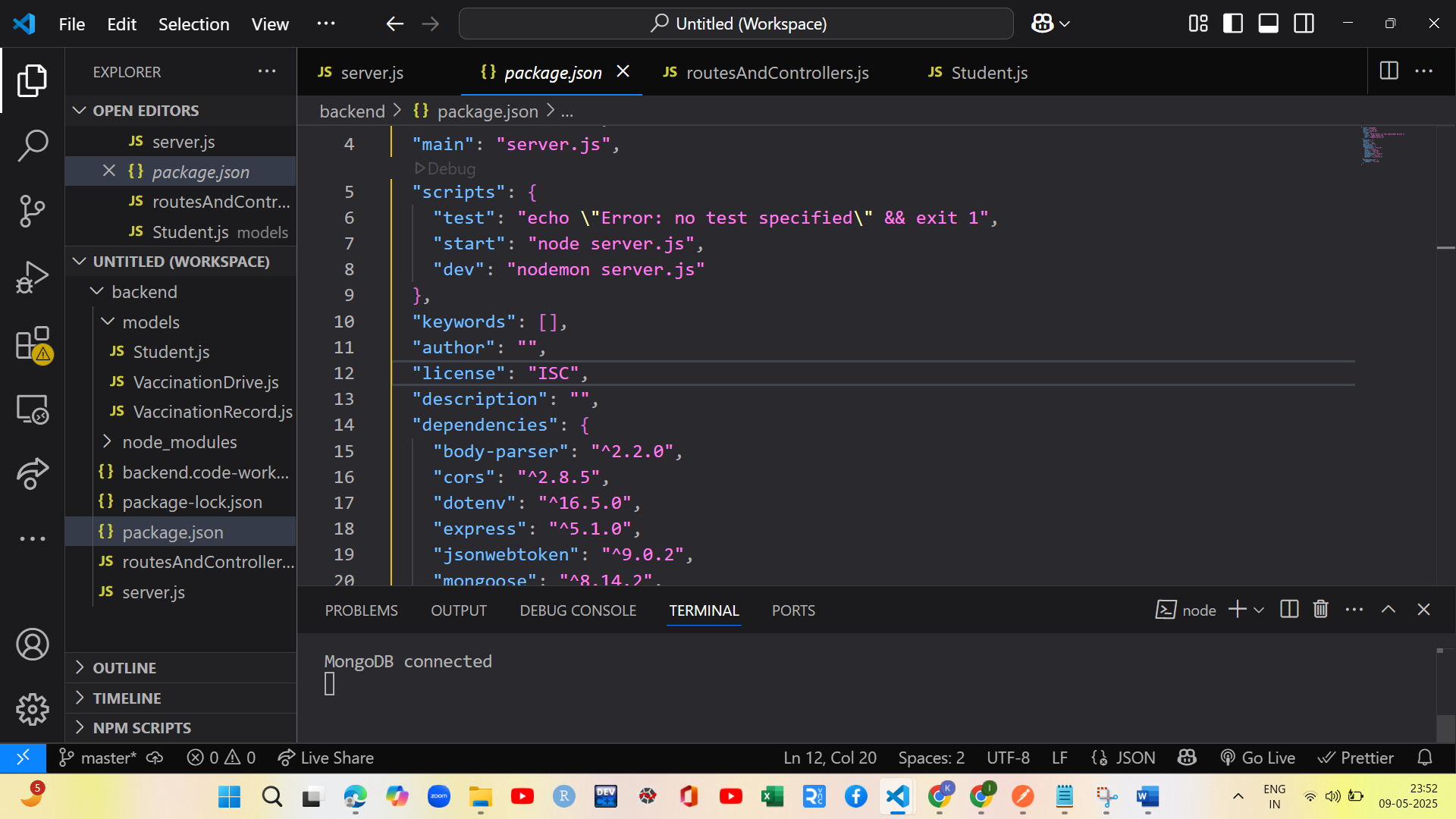
Created the database in mongodb and used appropriate network accesss in mongodb

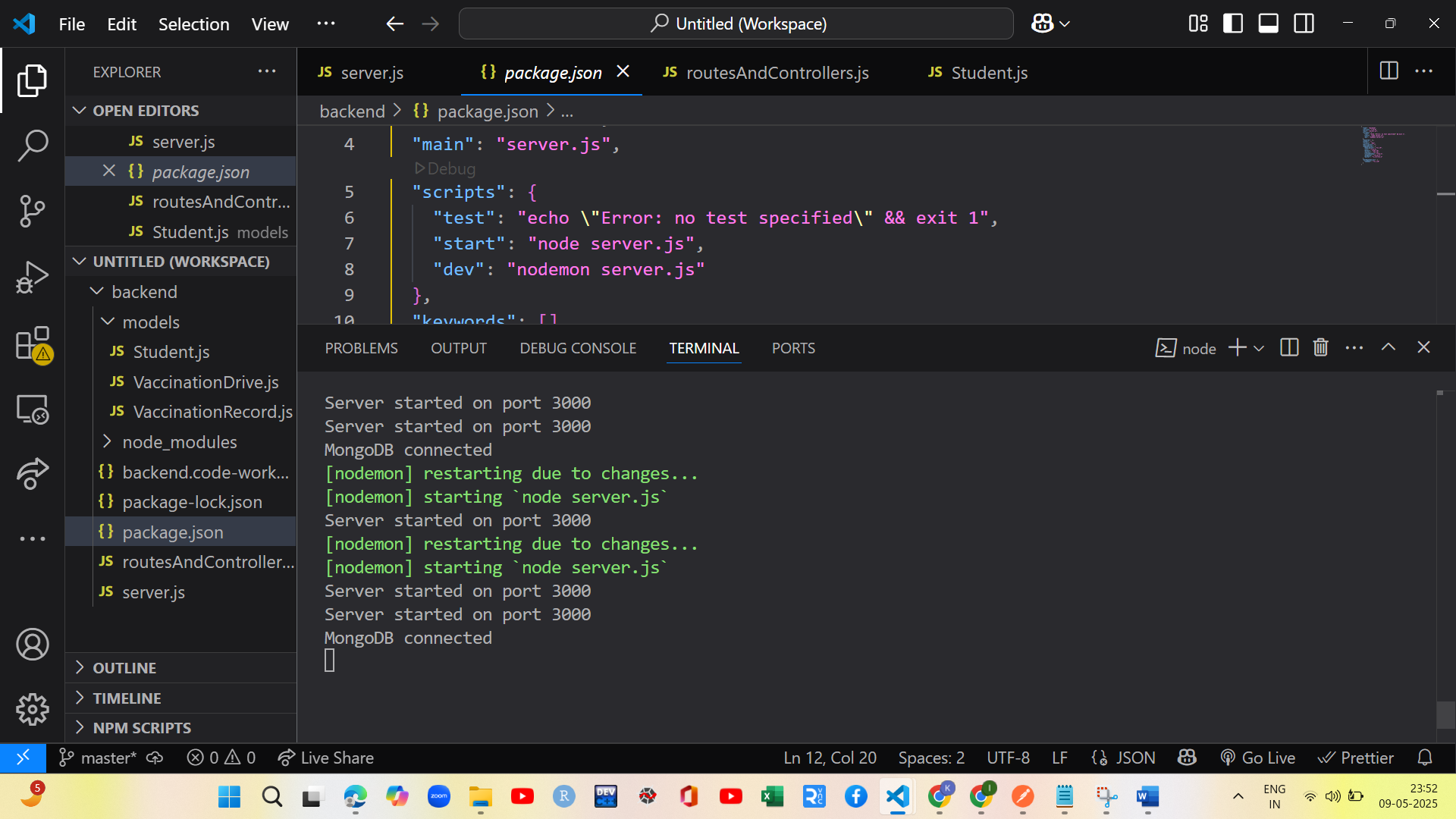




Now I connected to mongodb successfully using mongo.connect() method url is present in cluster connection properties.

Used nodemon to run the server added in the package.json and also start script. The server is running at port 3000.



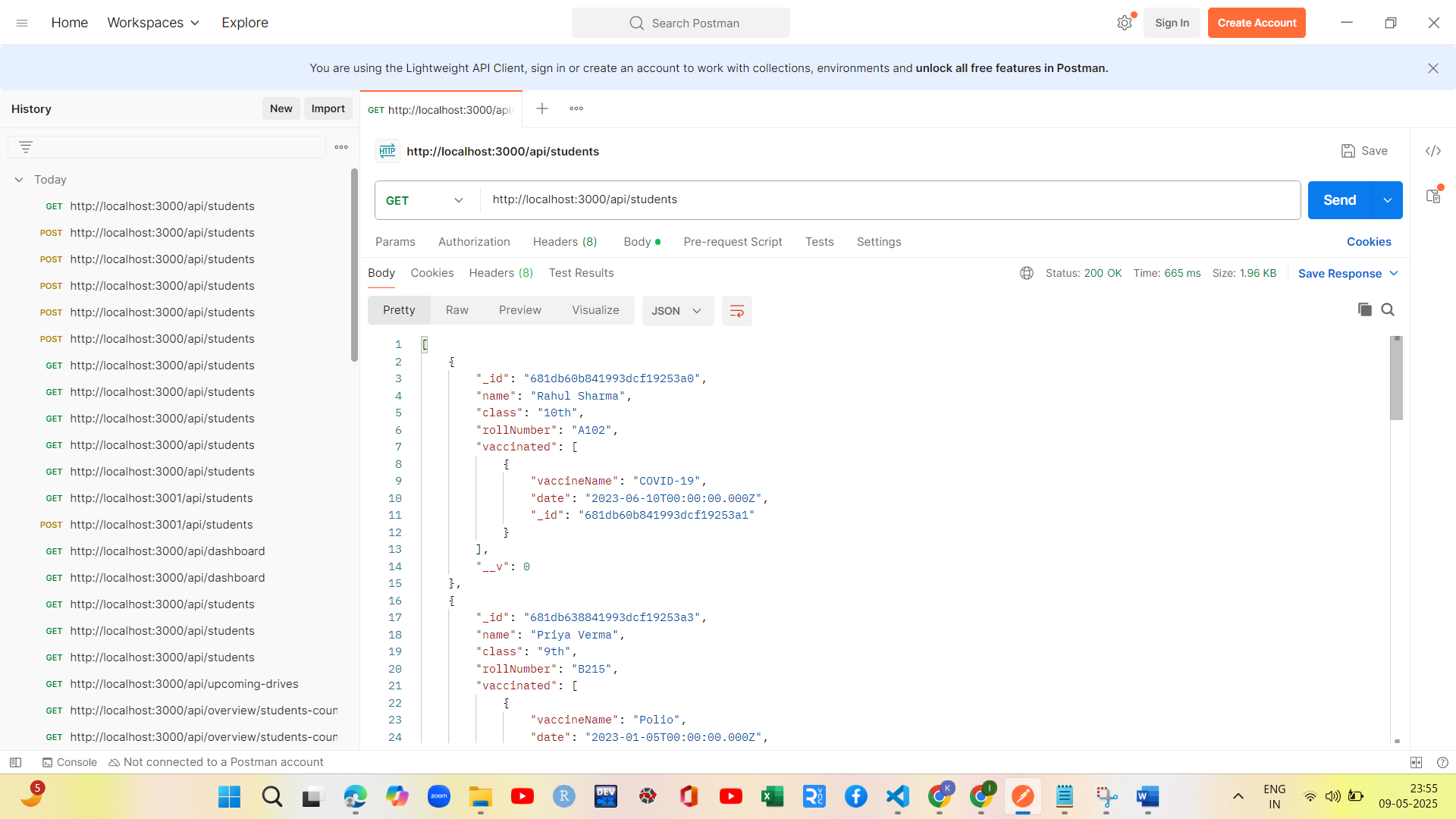


From the image it is shown that mongodb is successfully connected

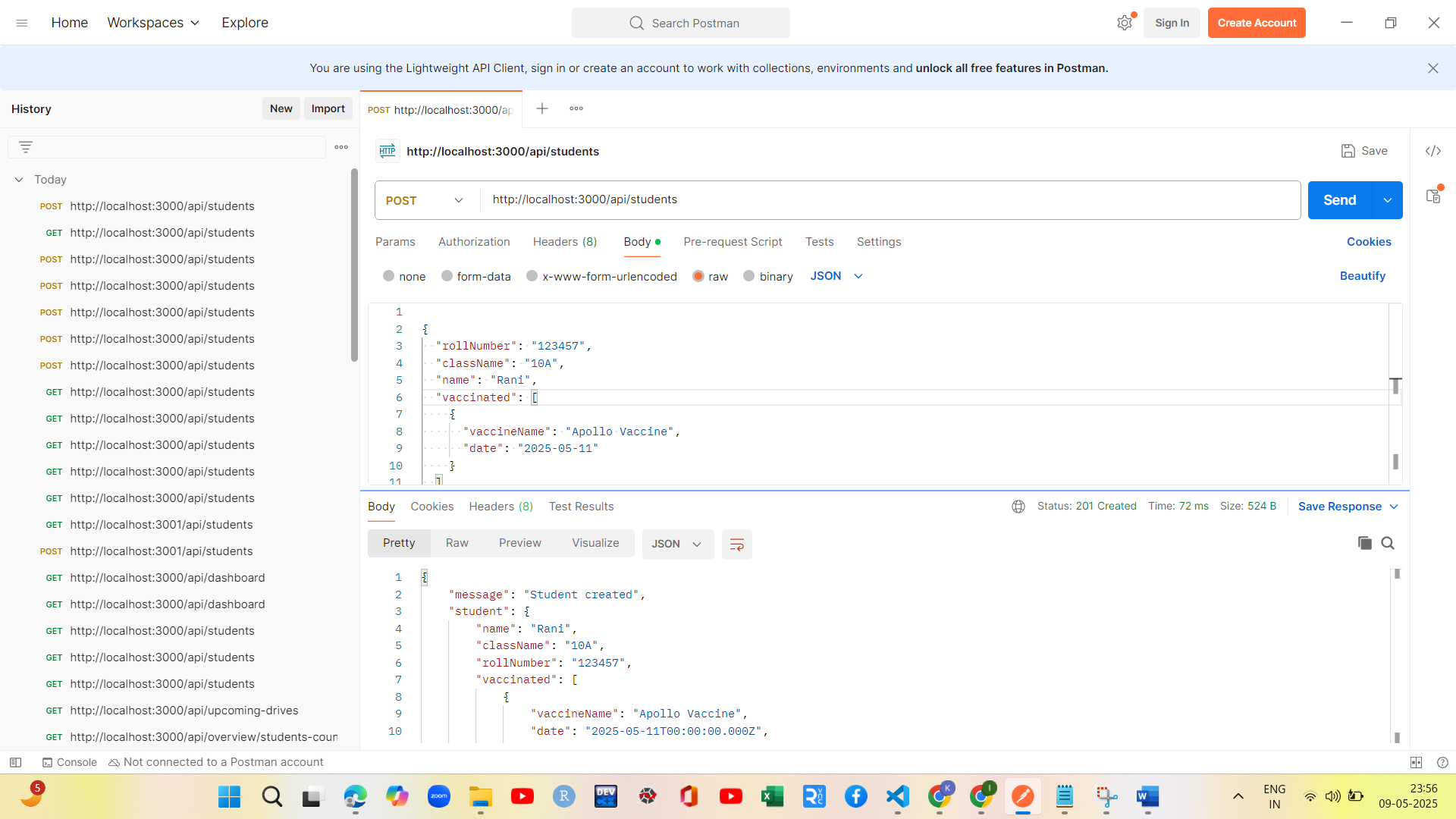
Now Test all the services using postman whether they are working or not

Students

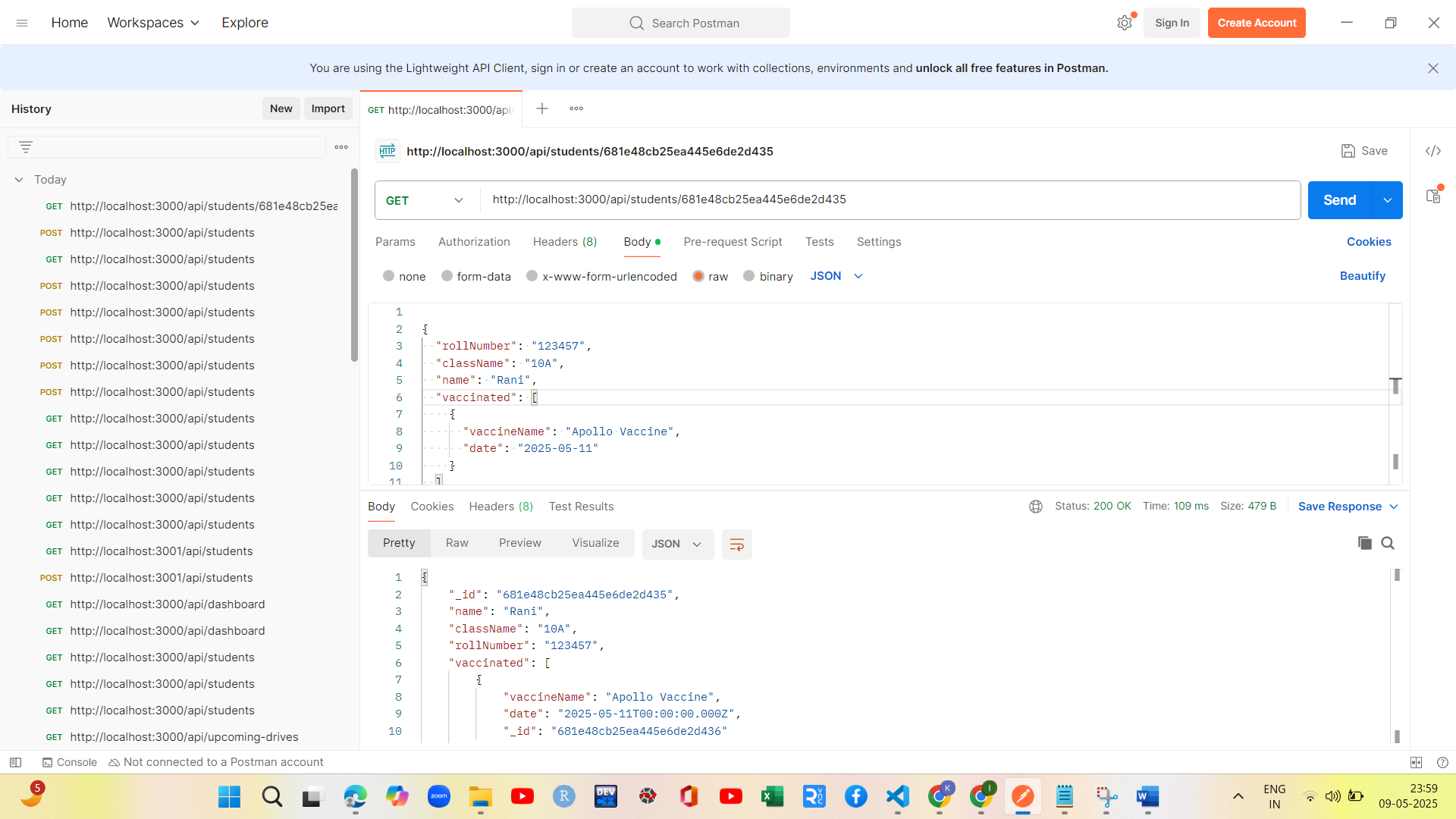
Get: <http://localhost:3000/api/students>



POST:- <http://localhost:3000/api/students>



GetbyId:- <http://localhost:3000/api/students/681e48cb25ea445e6de2d435>



Put:- <http://localhost:3000/api/students/681e48cb25ea445e6de2d435>

Updated apollo vaccine to apollo medicine

