**🏥 MedTrack – Cloud-Based Healthcare Management System**

**📌 Project Overview**

**MedTrack** is a web-based healthcare management system that connects patients and doctors. It allows patients to book appointments and view diagnosis reports, while doctors can manage appointments and submit diagnoses. The project uses **AWS DynamoDB** for data storage and **AWS SNS** for real-time notifications.

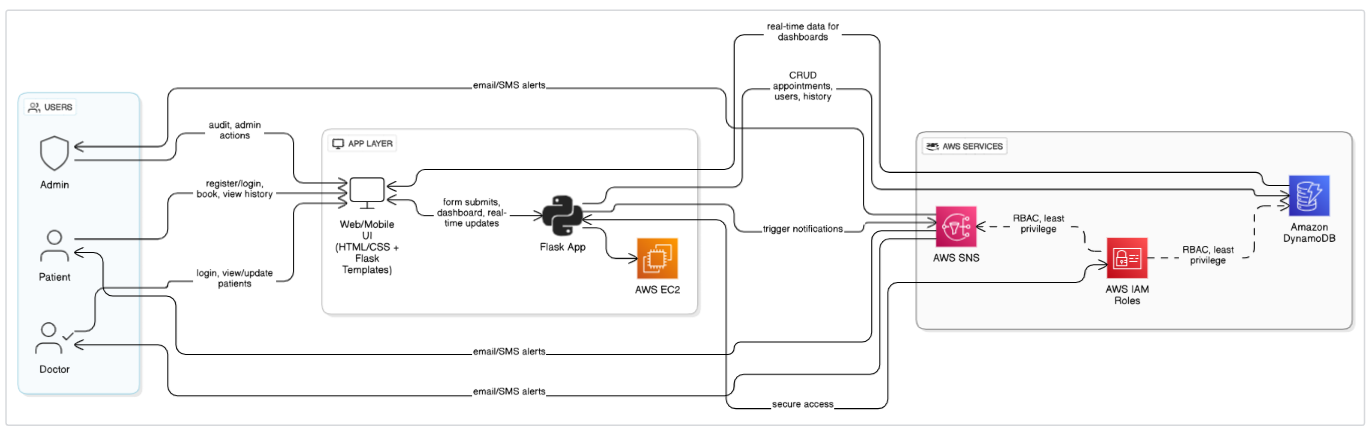
✨ **Key Features**

* 🔐 Secure login & role-based access (Patient / Doctor)
* 📅 Book and view appointments
* 📝 Submit and view diagnosis reports
* ☁️ Data stored in AWS DynamoDB
* 📣 Notifications via AWS SNS
* 🌐 Deployed using AWS EC2

⚙️ **Tech Stack**

* Frontend: HTML, CSS
* Backend: Python (Flask)
* Database: DynamoDB (NoSQL)
* Cloud Services: AWS EC2, AWS SNS
* Version Control: Git & GitHub

**AWS Architecture**

****

**Entity Relationship (ER)Diagram:**

****

**📦 Project Structure**

MedTrack/

├── app.py

├── .env

├── templates/         # HTML pages

├── static/            # CSS, images

├── utils/             # Logic for AWS & data

├── create\_tables.py   # DynamoDB setup (optional)

├── README.md

**🗃️ Database Tables (DynamoDB)**

|  |  |  |
| --- | --- | --- |
| Table | Partition Key | Attributes |
| Users | username | password, role |
| Appointments | appointment \_ id | patient, doctor, date, time |
| Diagnoses | diagnosis \_ id | patient, doctor, notes |

**🧪 Testing and Deployment**

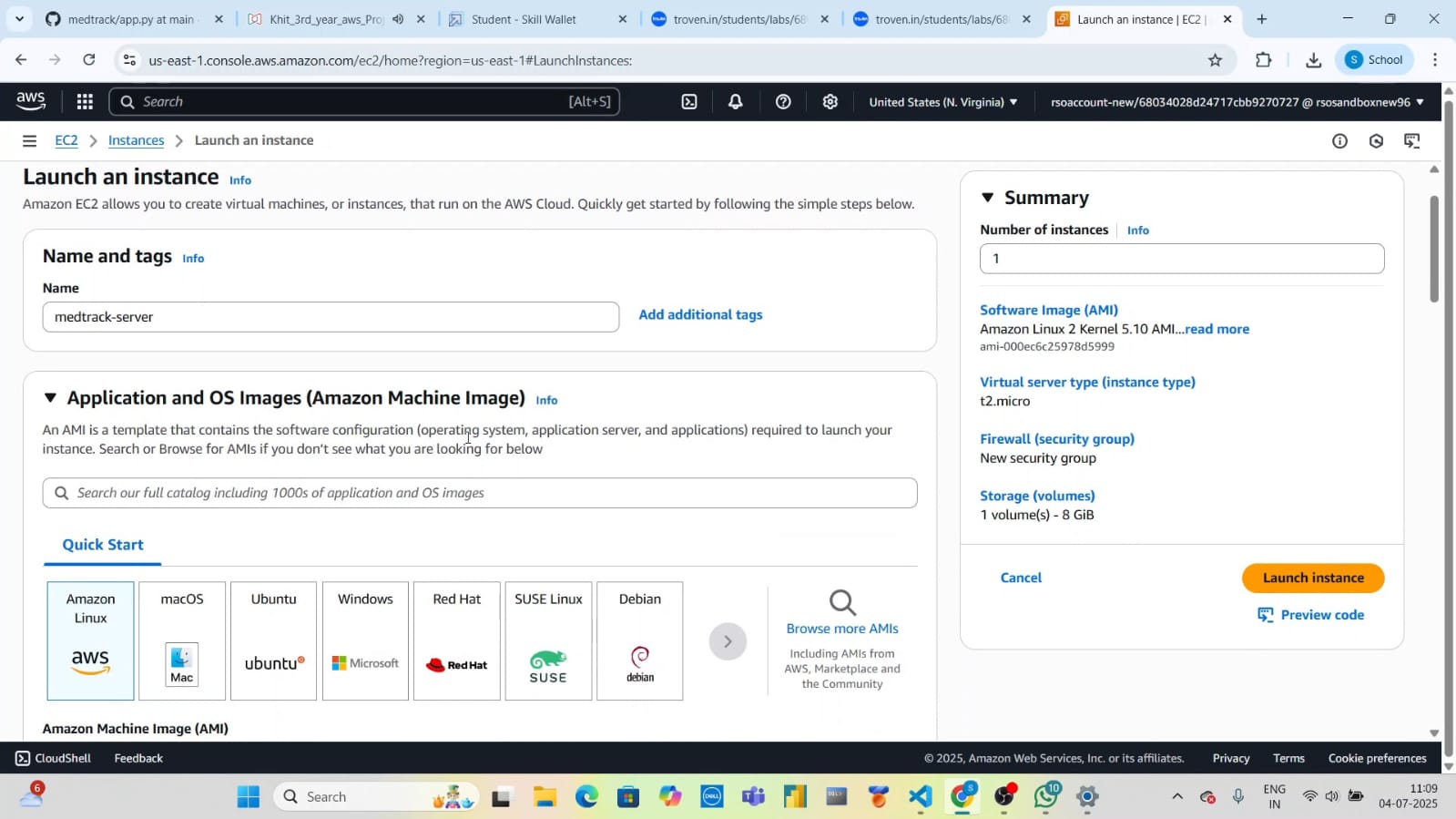
**Local Testing**

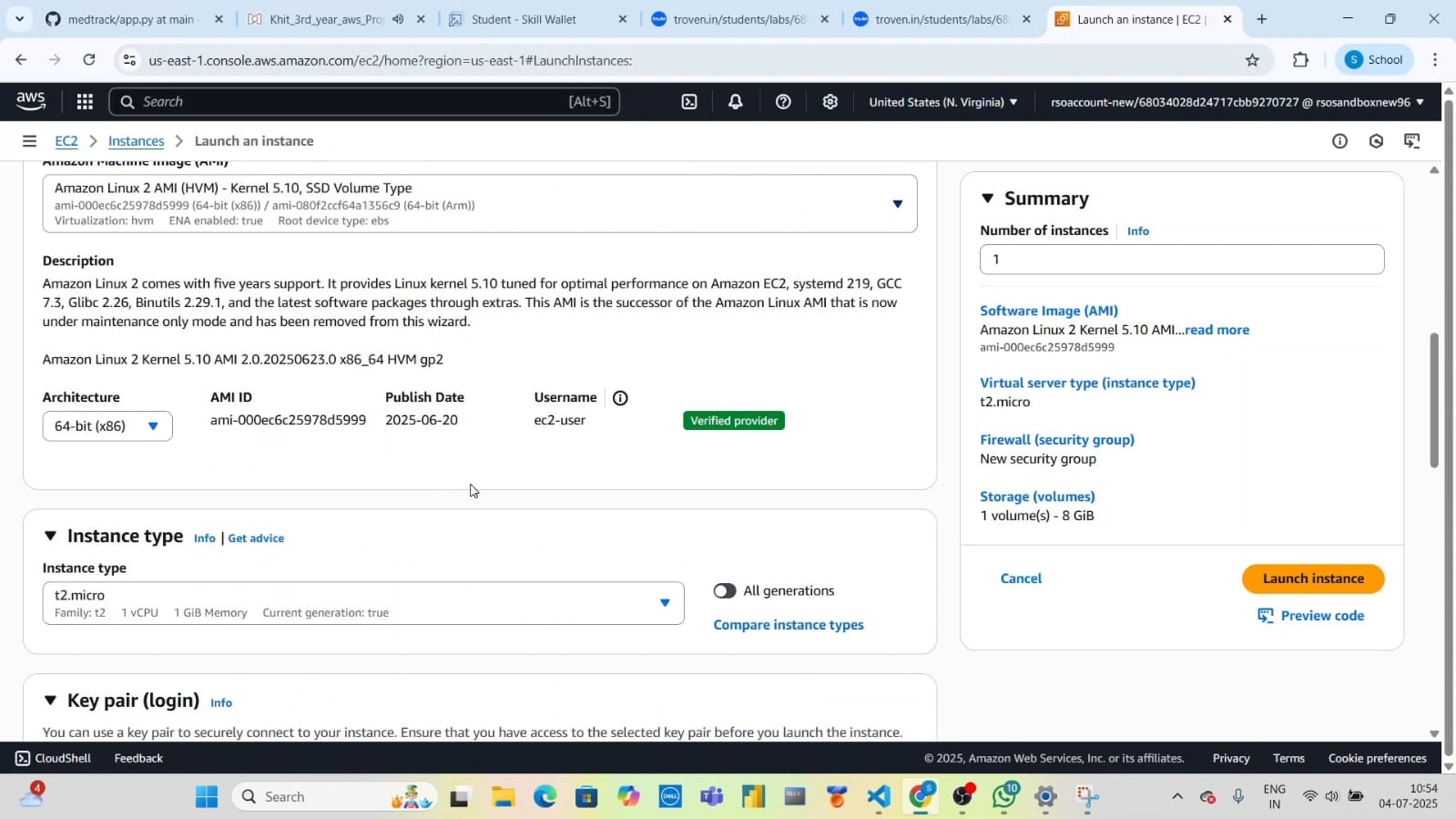
1. Install dependencies:  
   **pip install flask boto3 python-dotenv**
2. Run the Flask server:  
   **python app.py**

**☁️Deployment Steps**

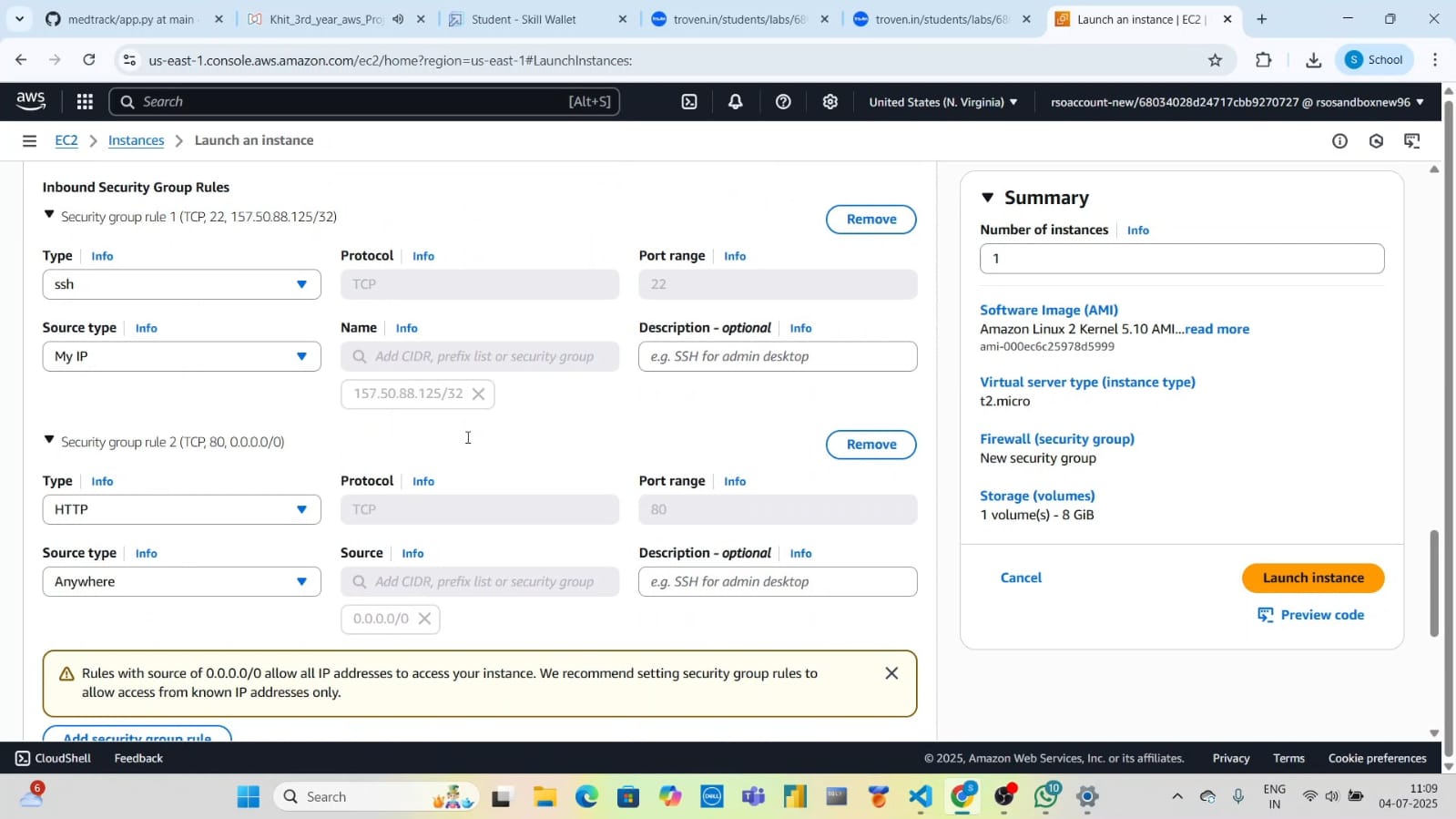
1.Launch EC2 Instance





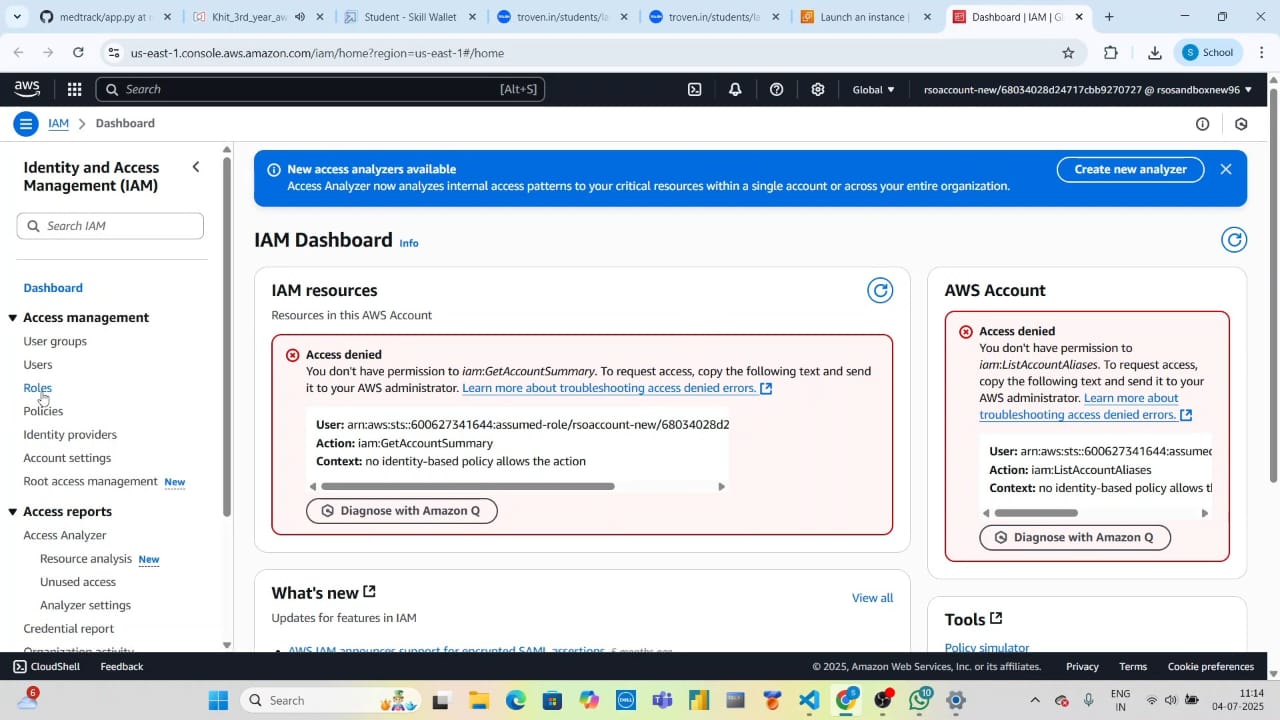


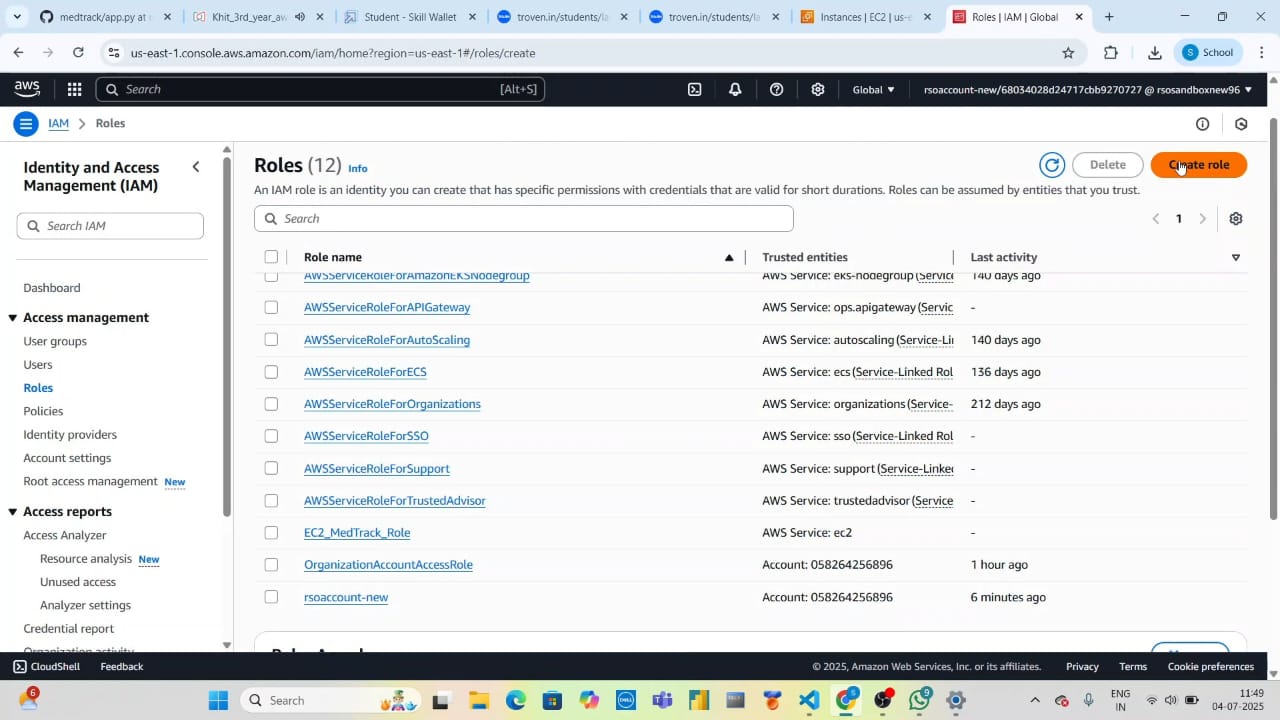


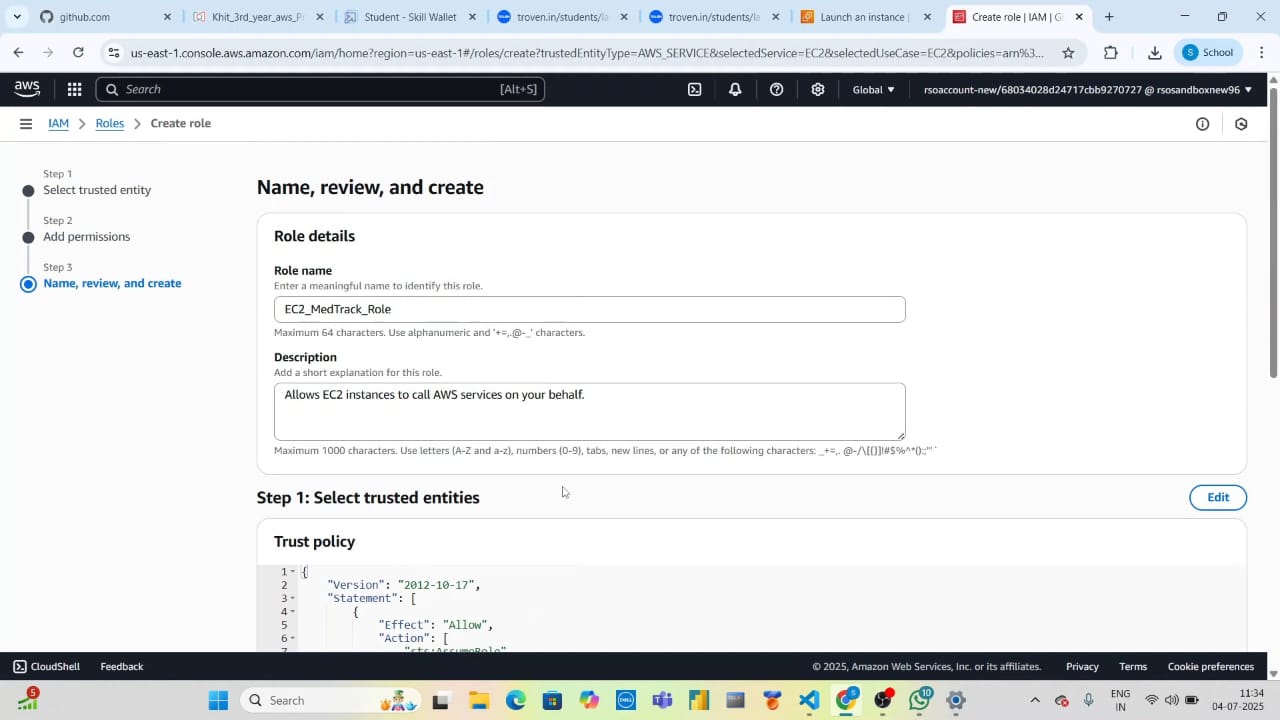


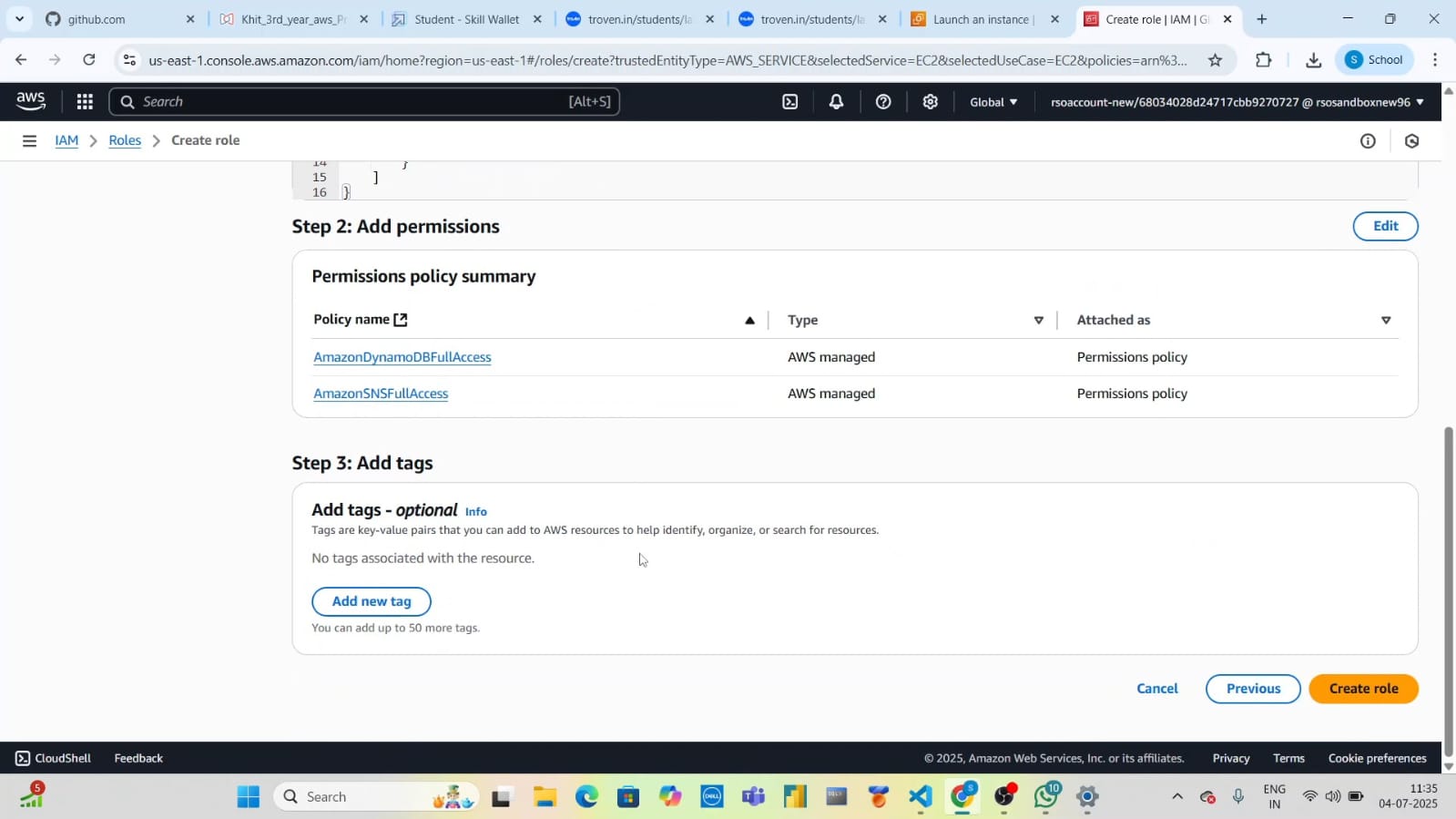


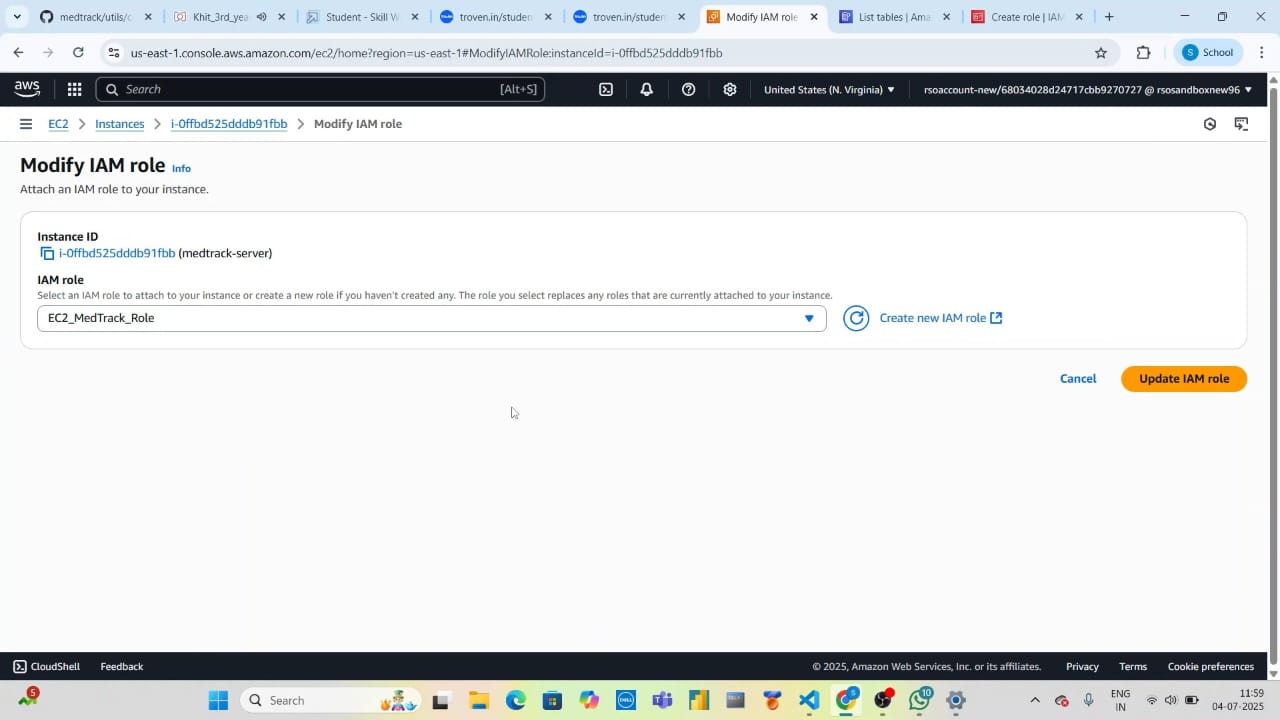
2.Create IAM Role with necessary permissions and attach to EC2

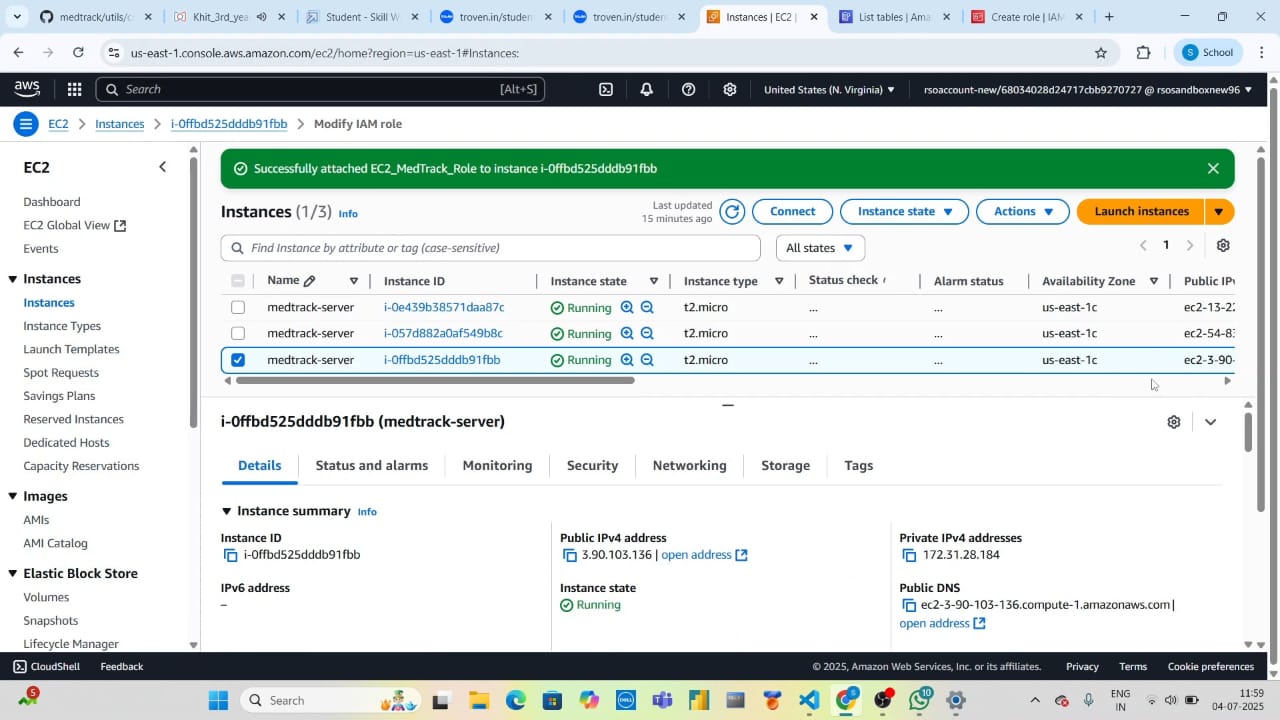




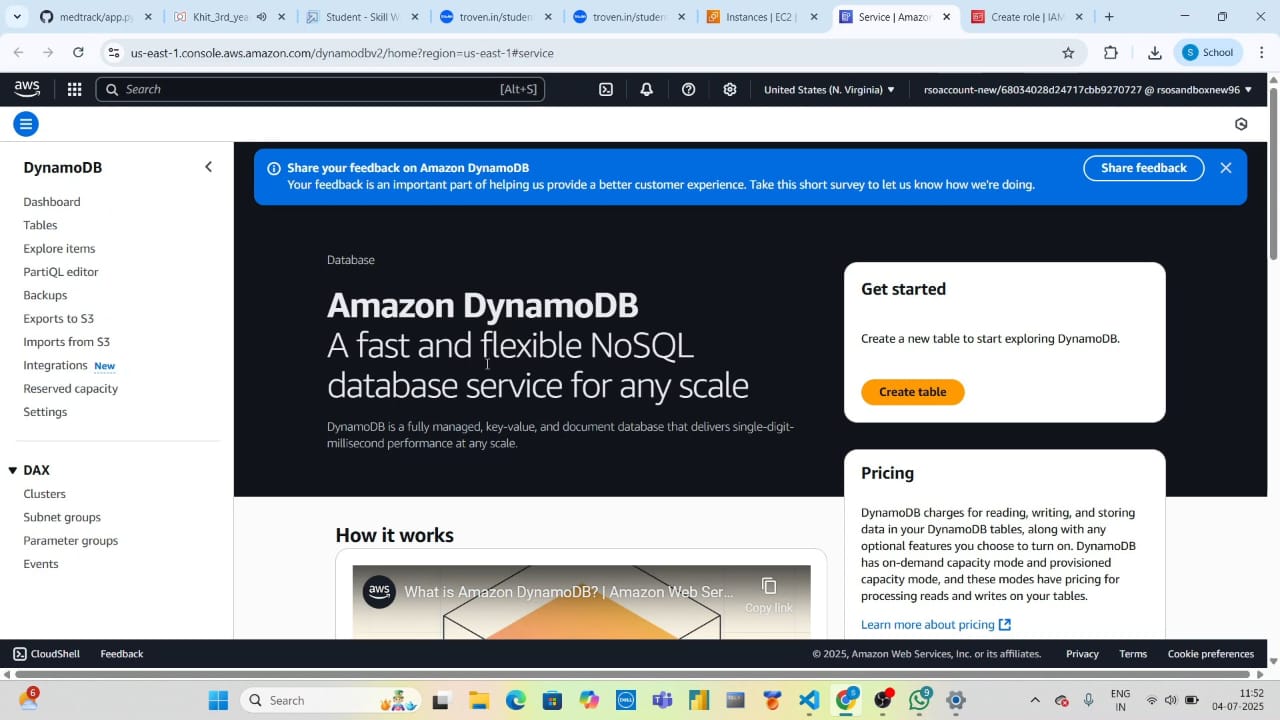


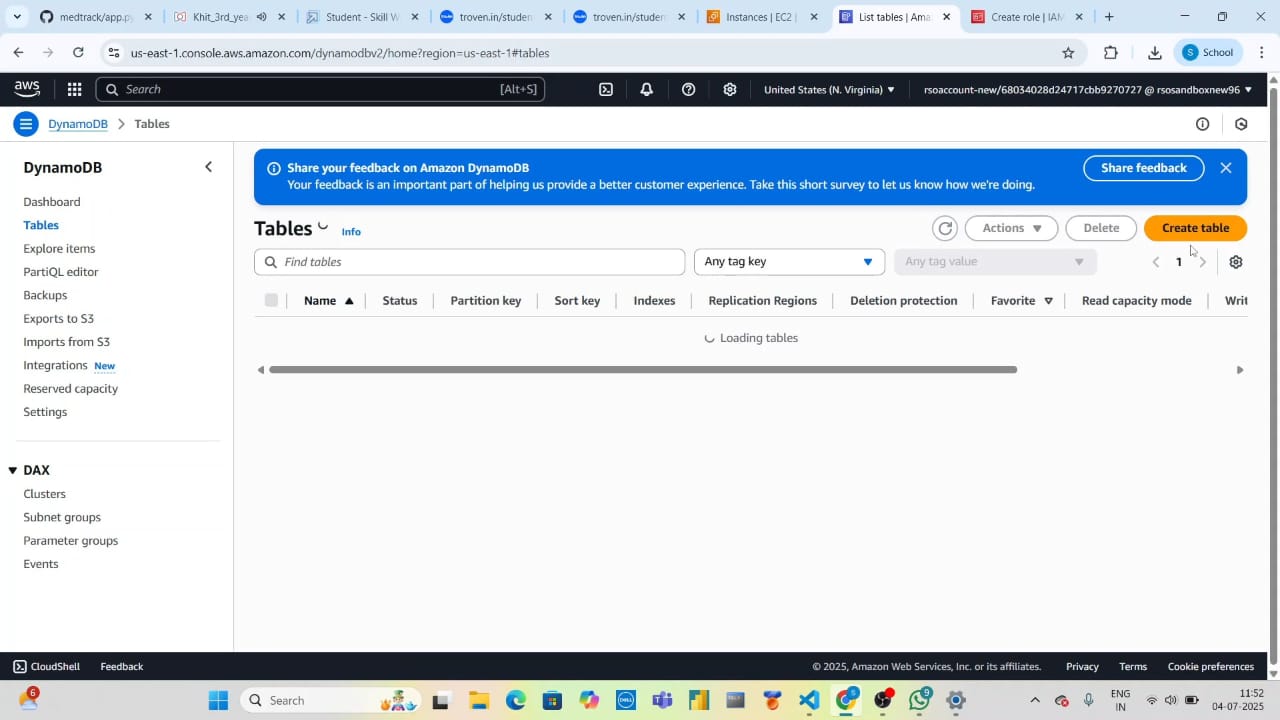


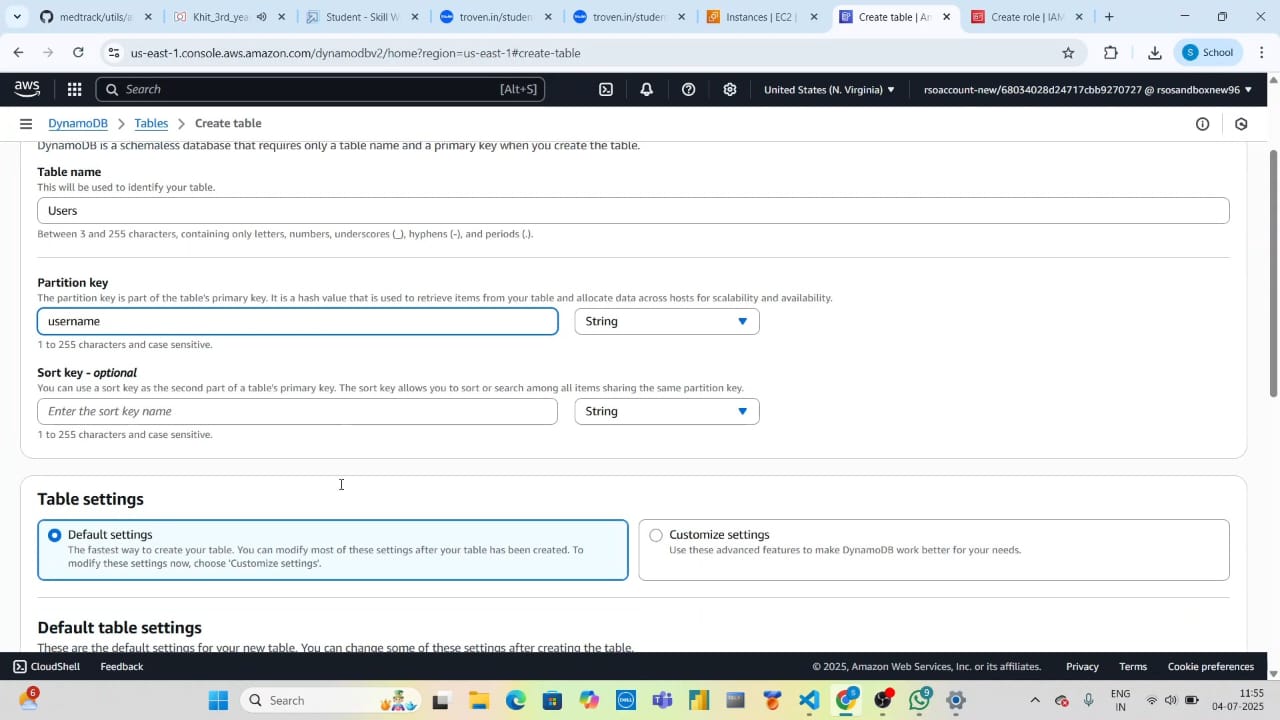


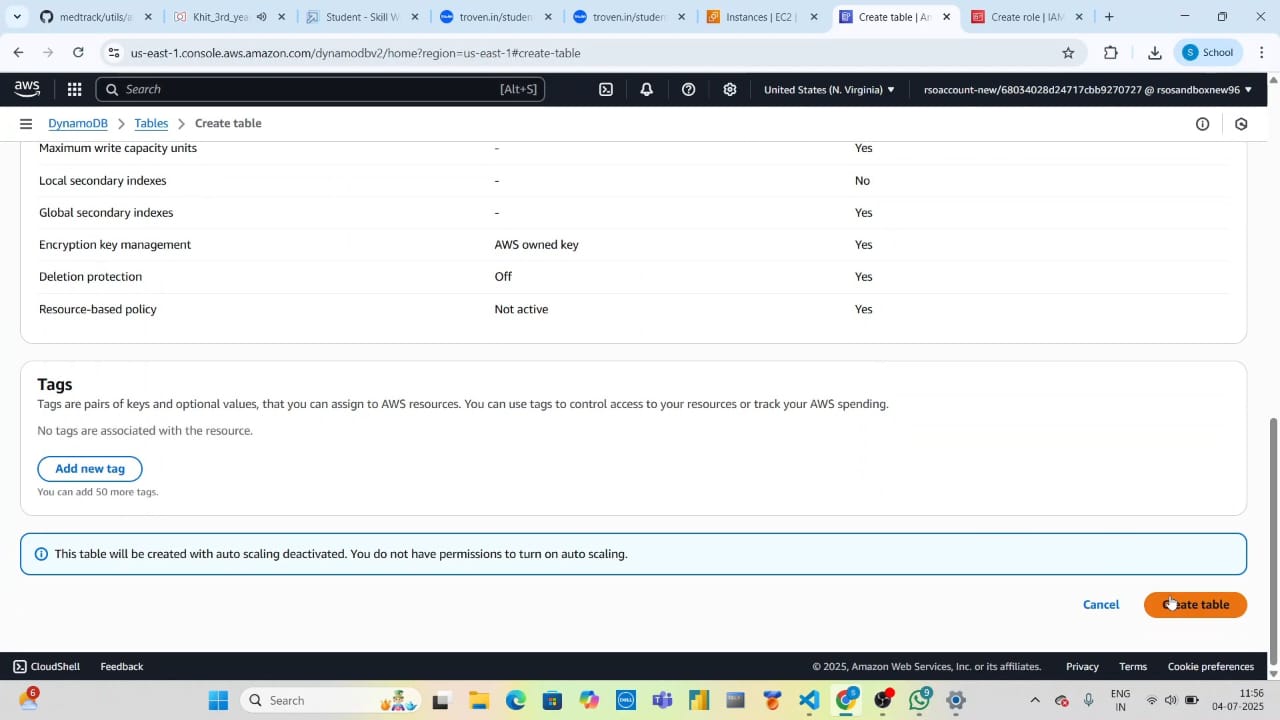


3. Setup DynamoDB Tables: Users, Appointments, Diagnoses

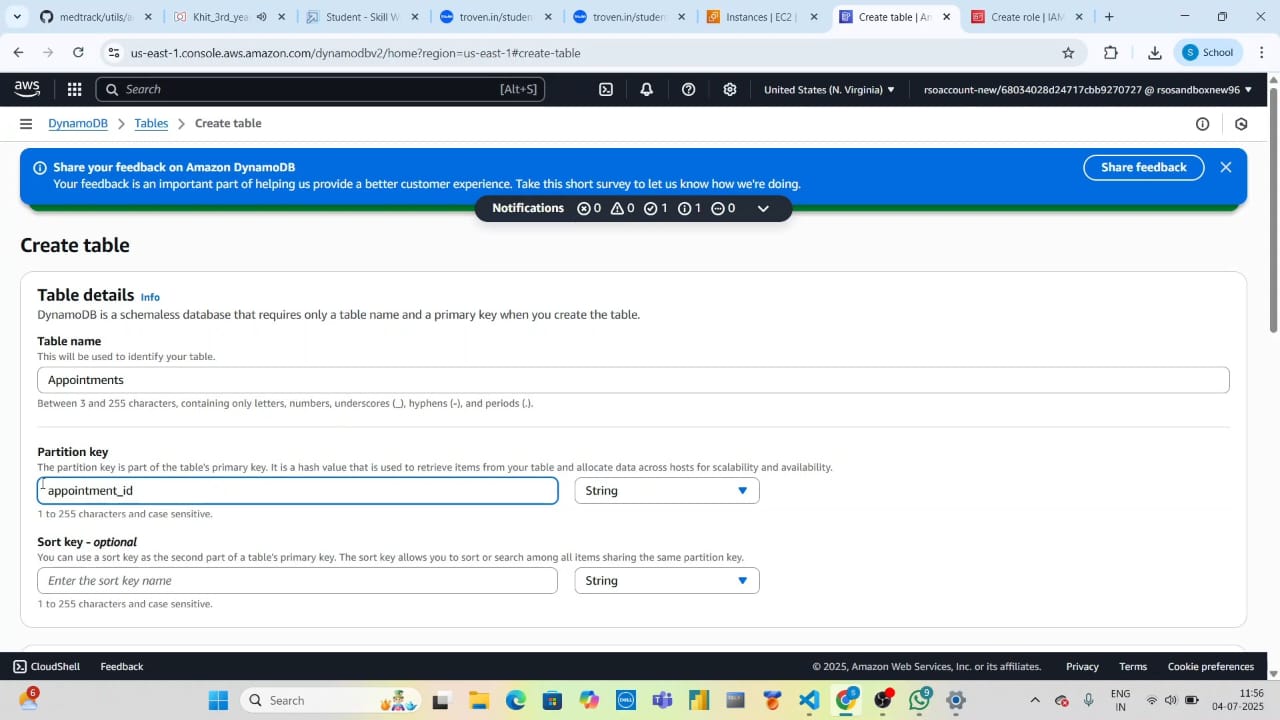


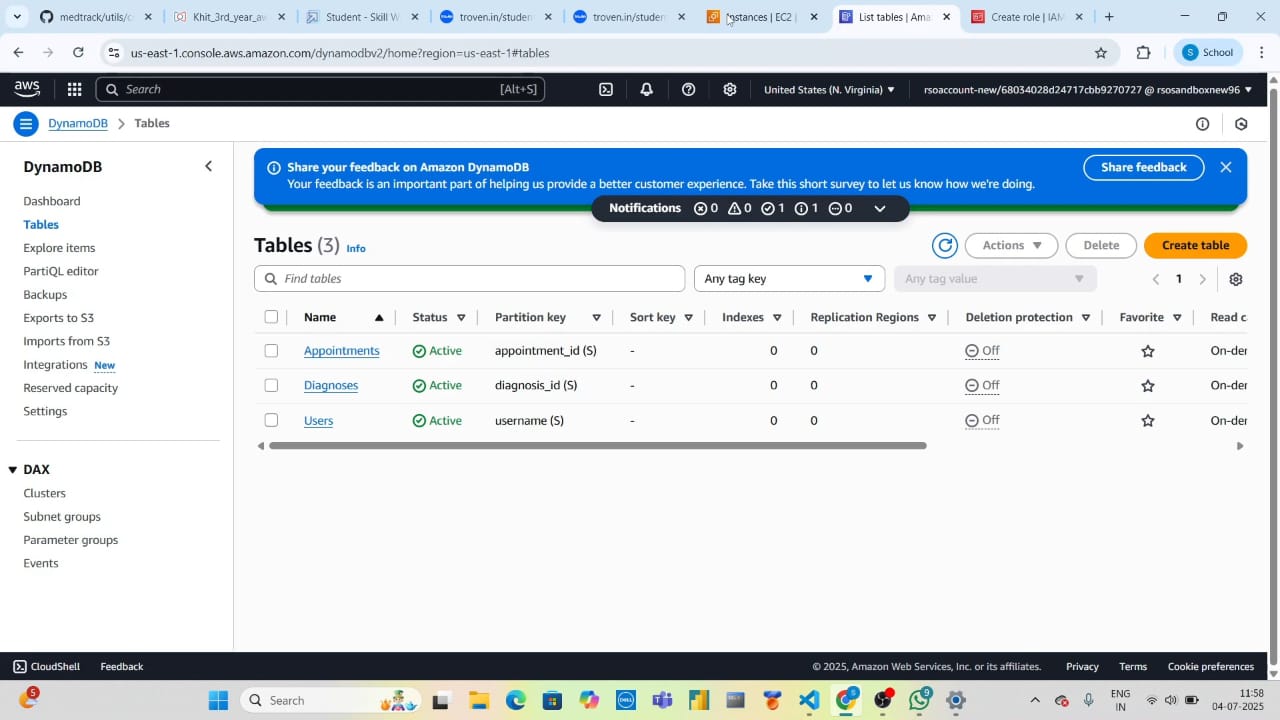




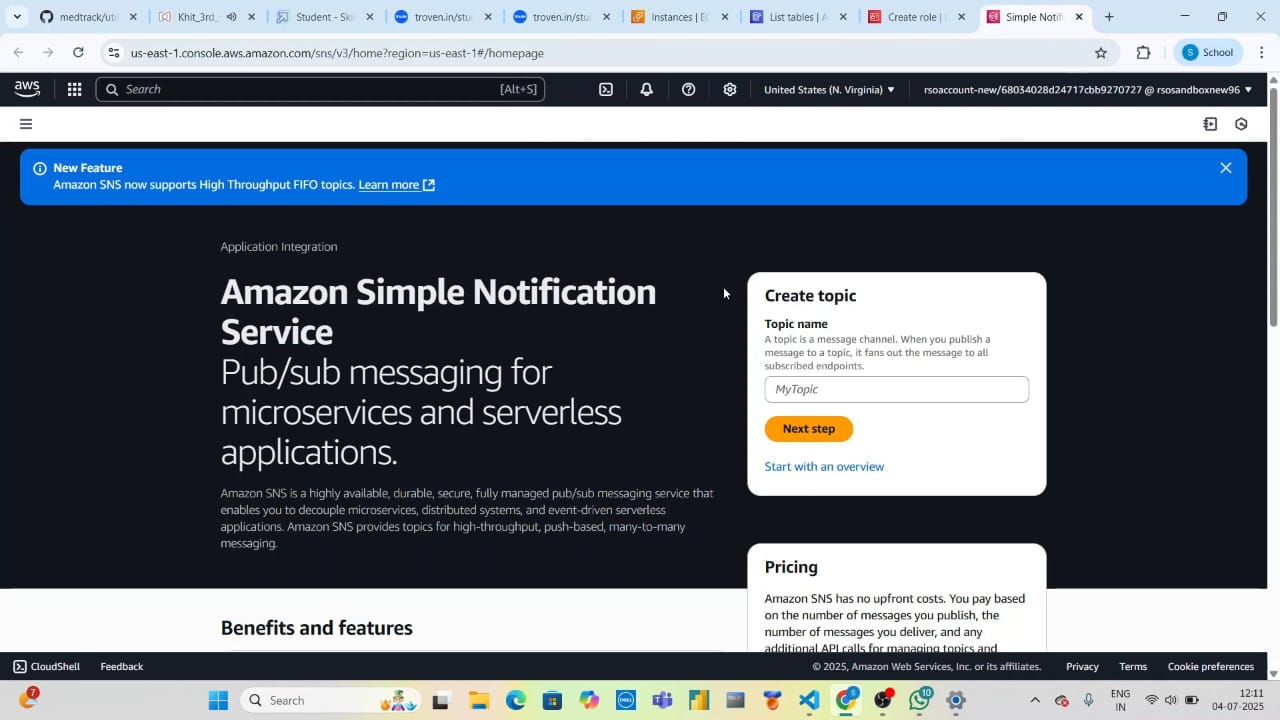


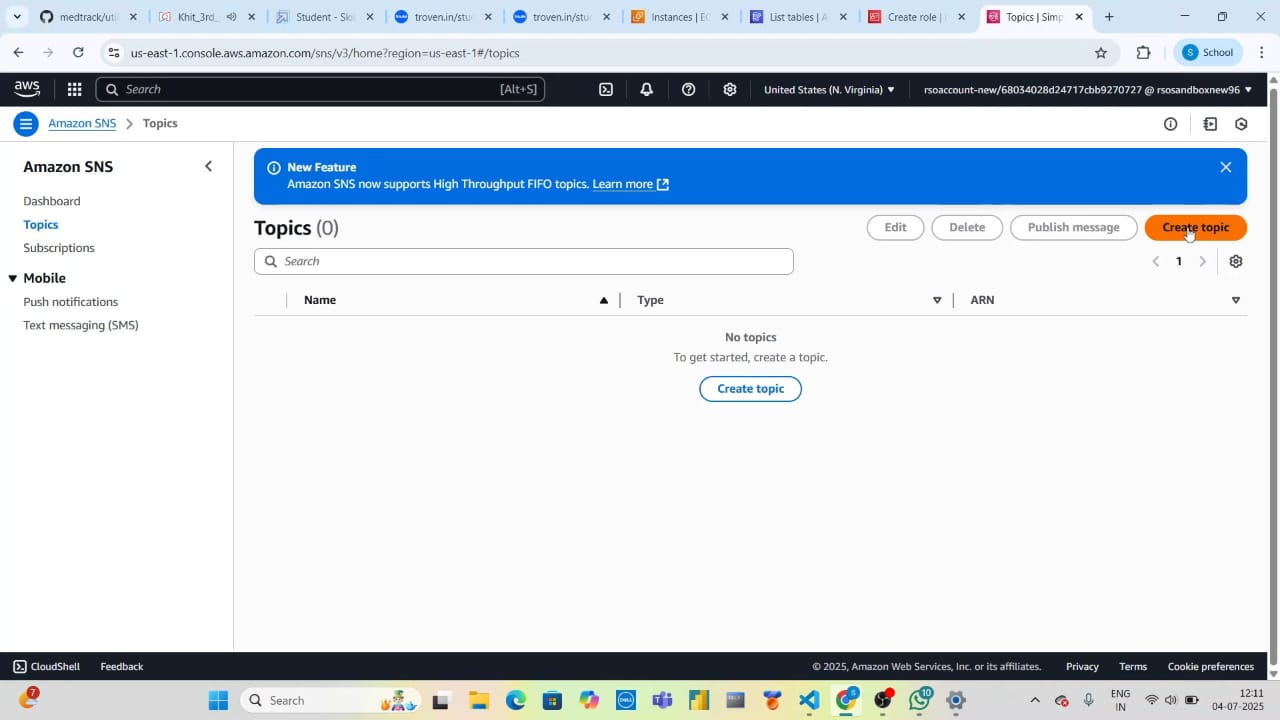


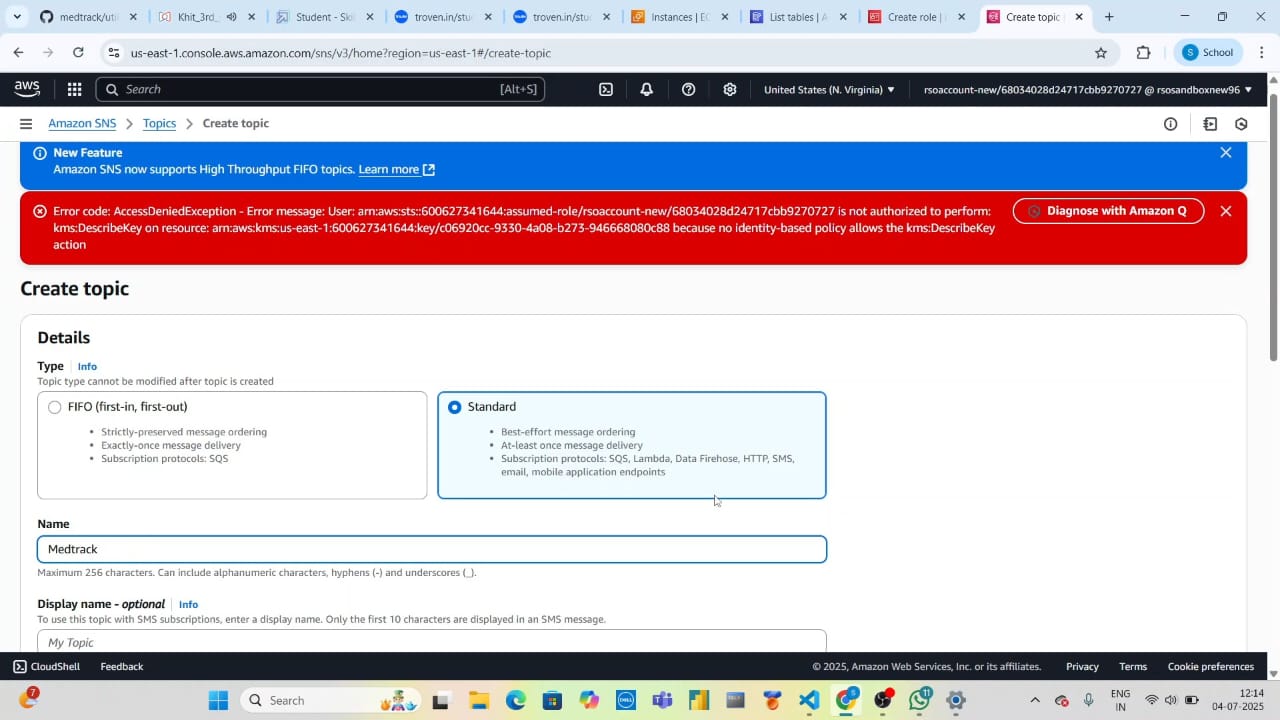


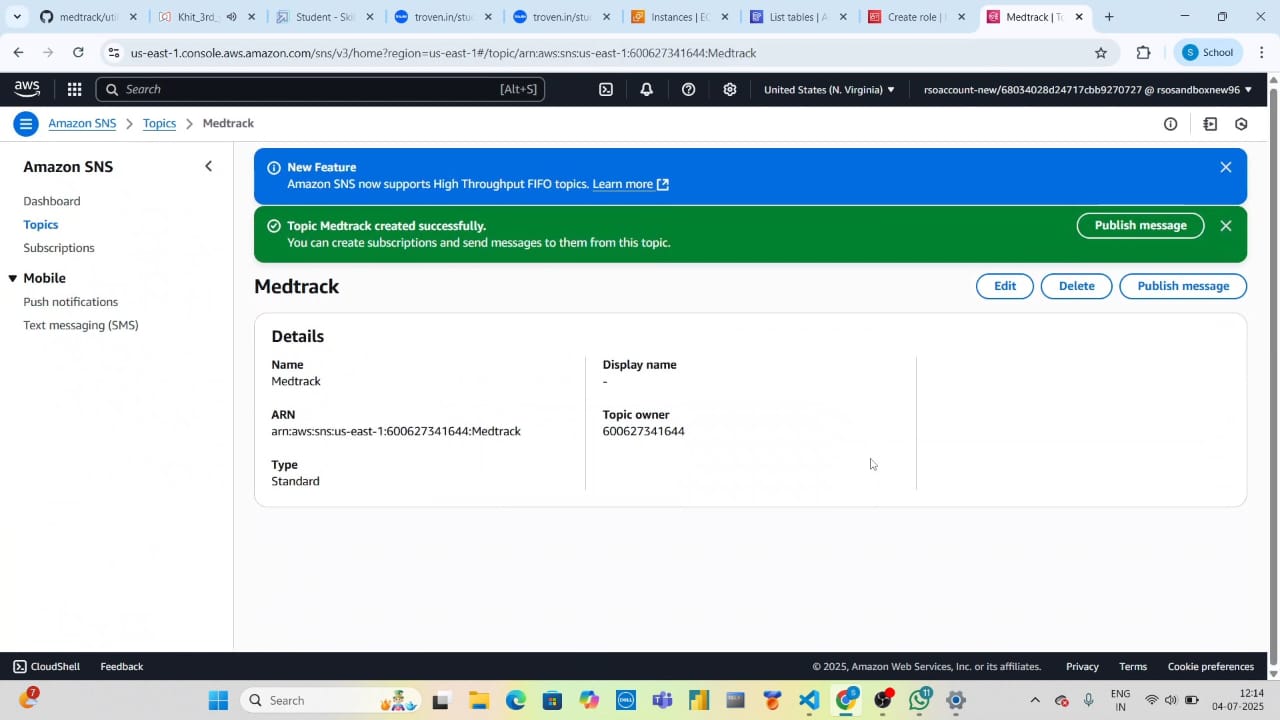


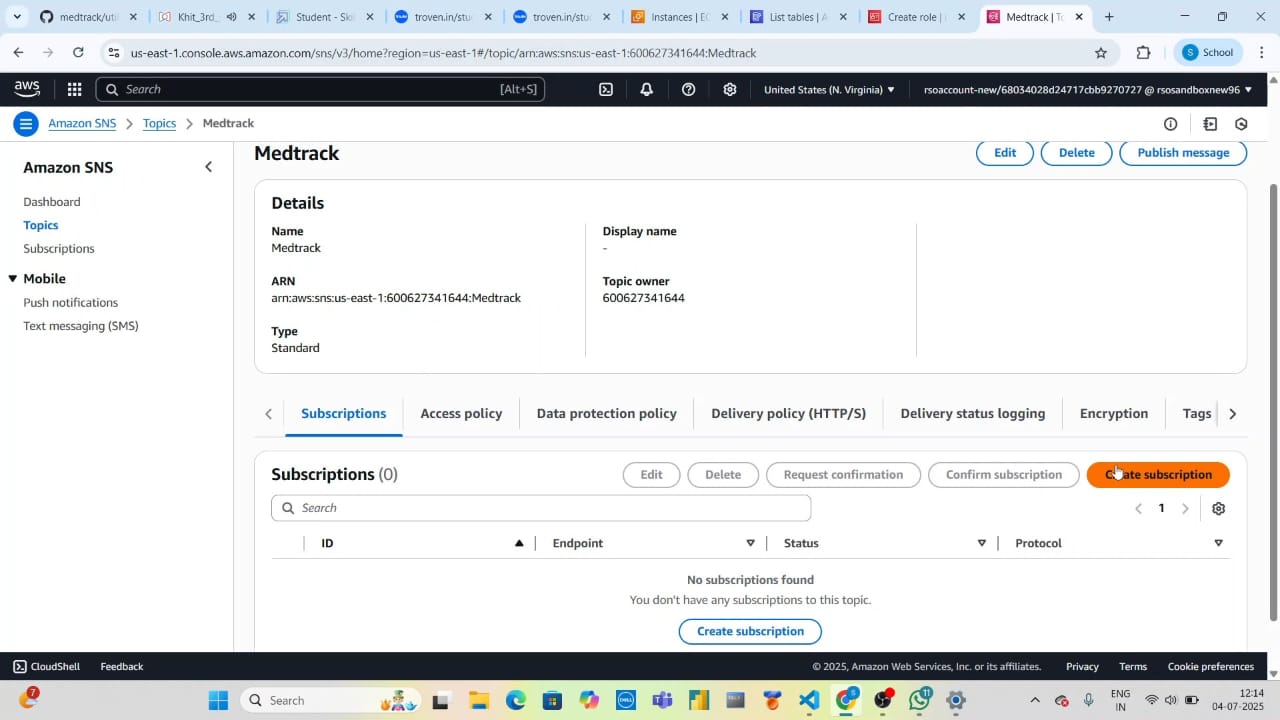
4.Configure SNS

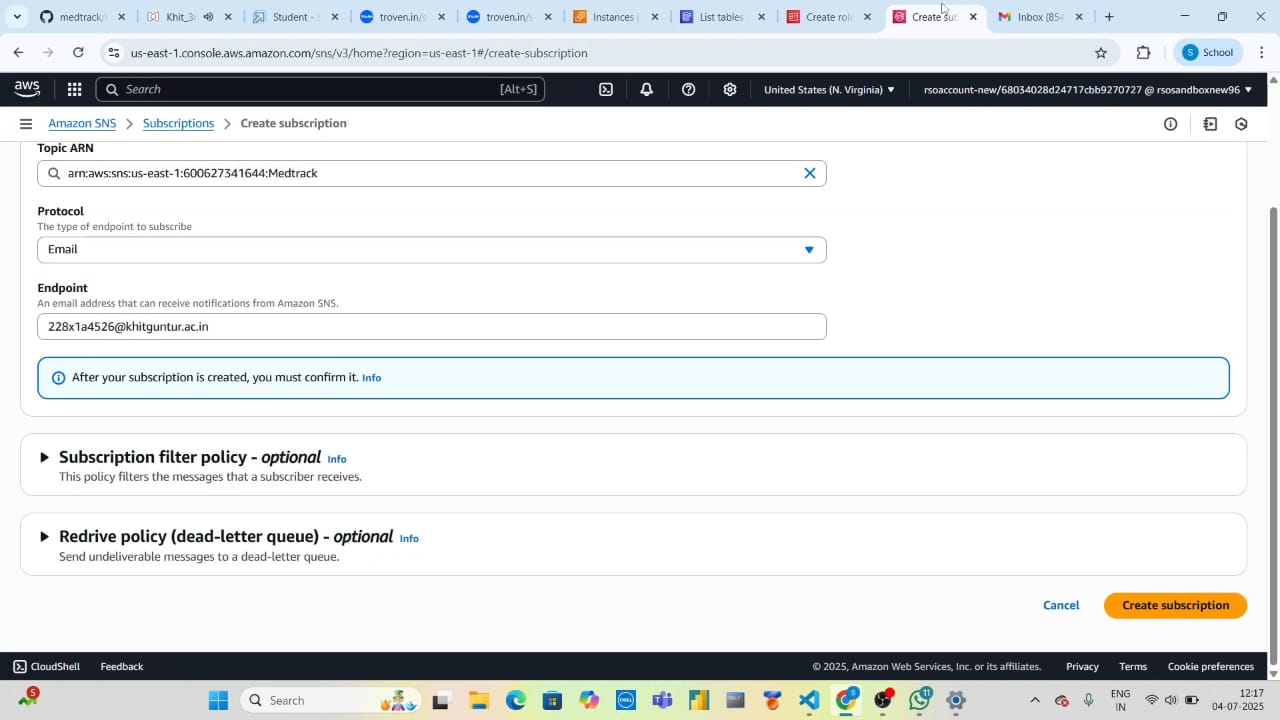


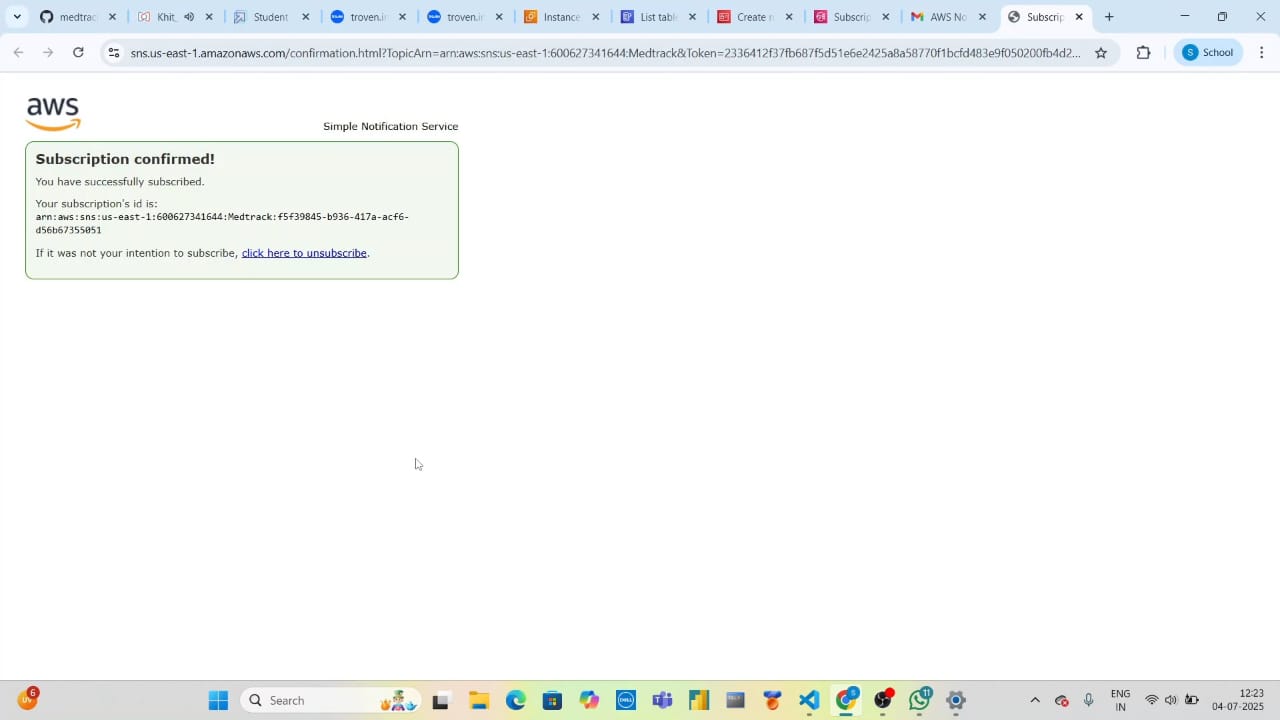


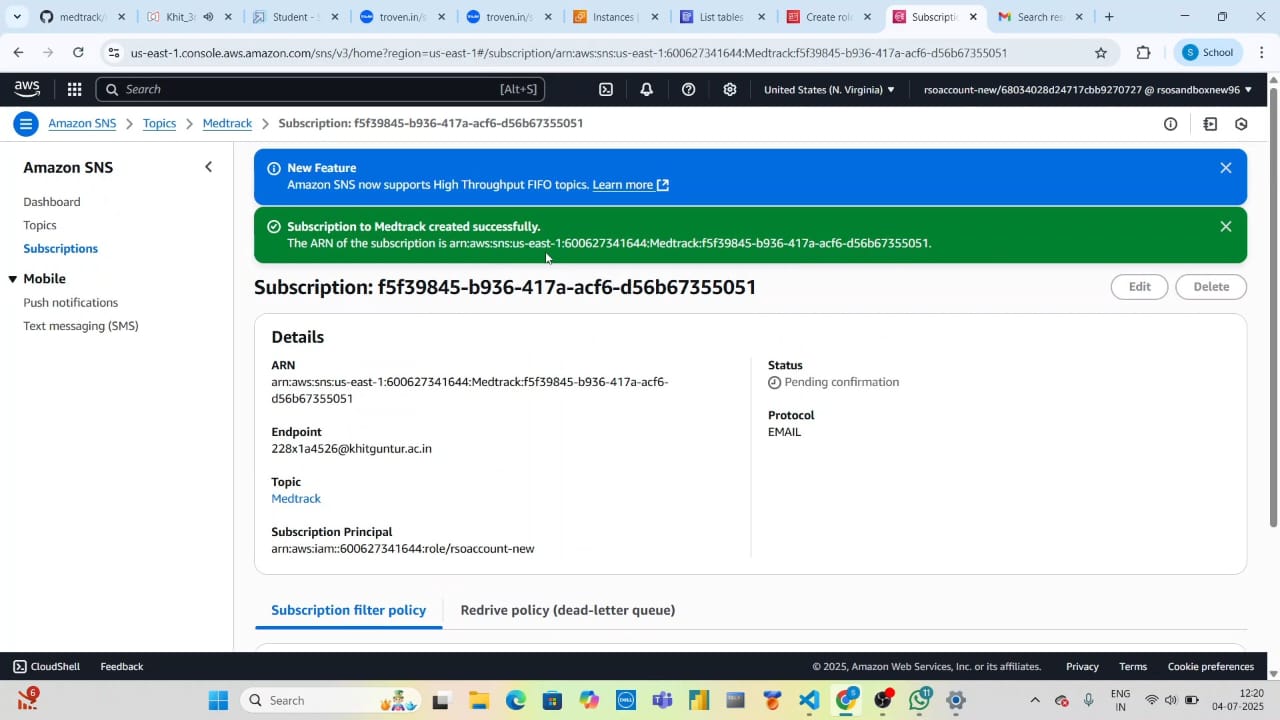


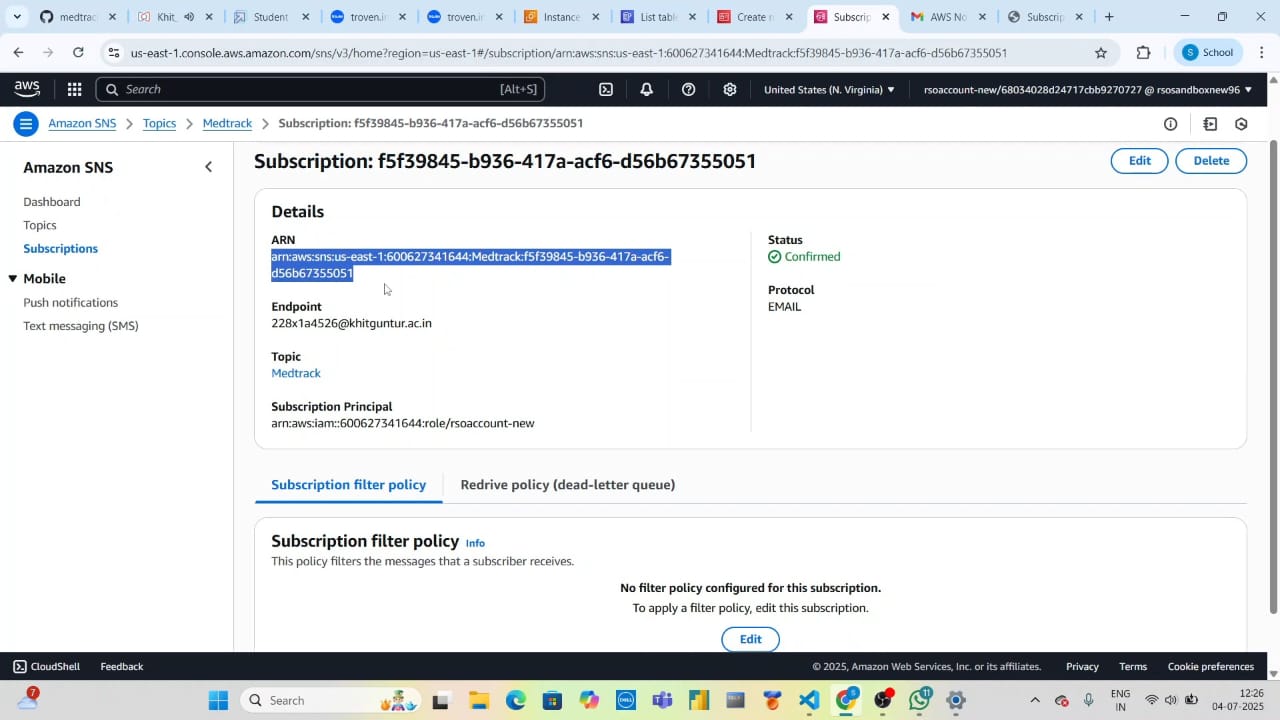






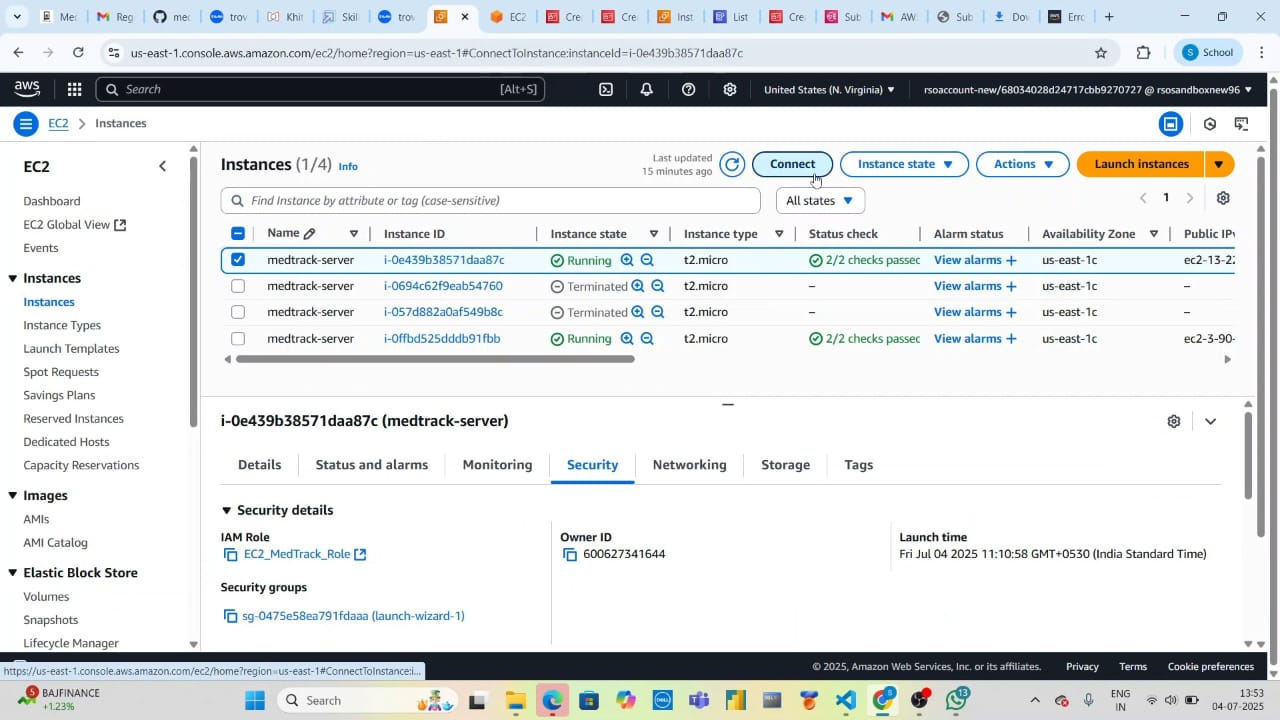


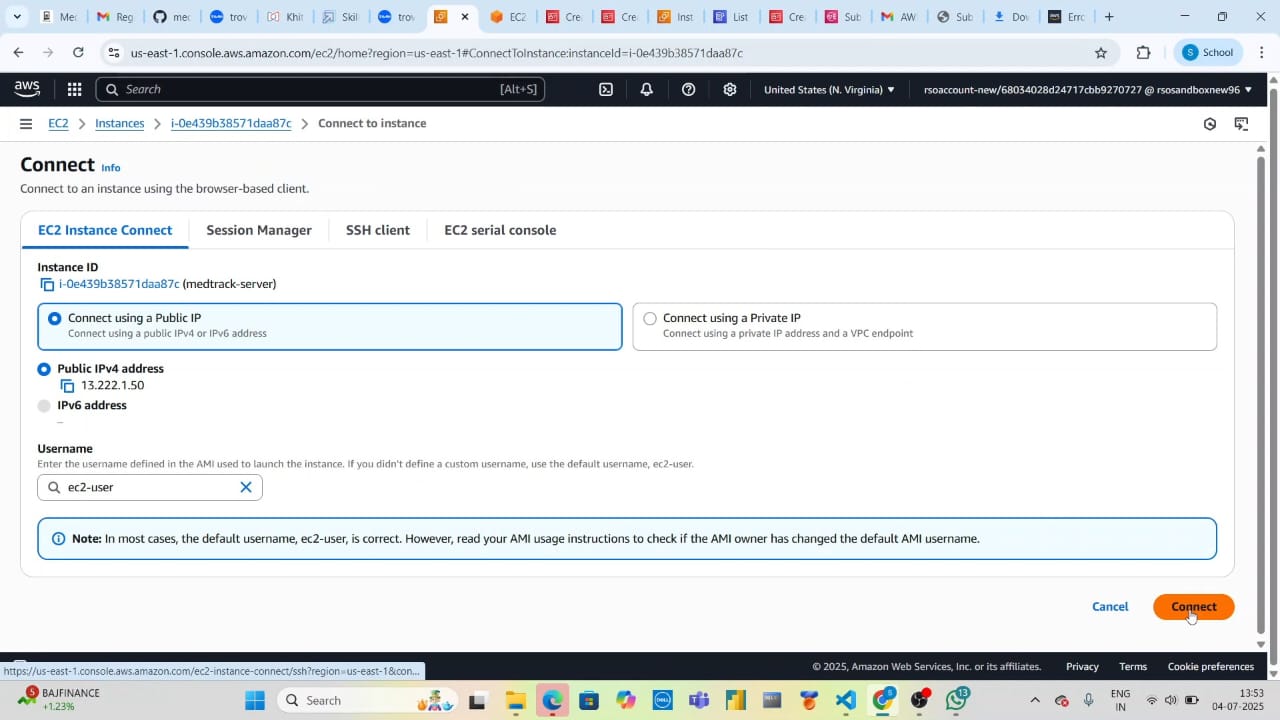


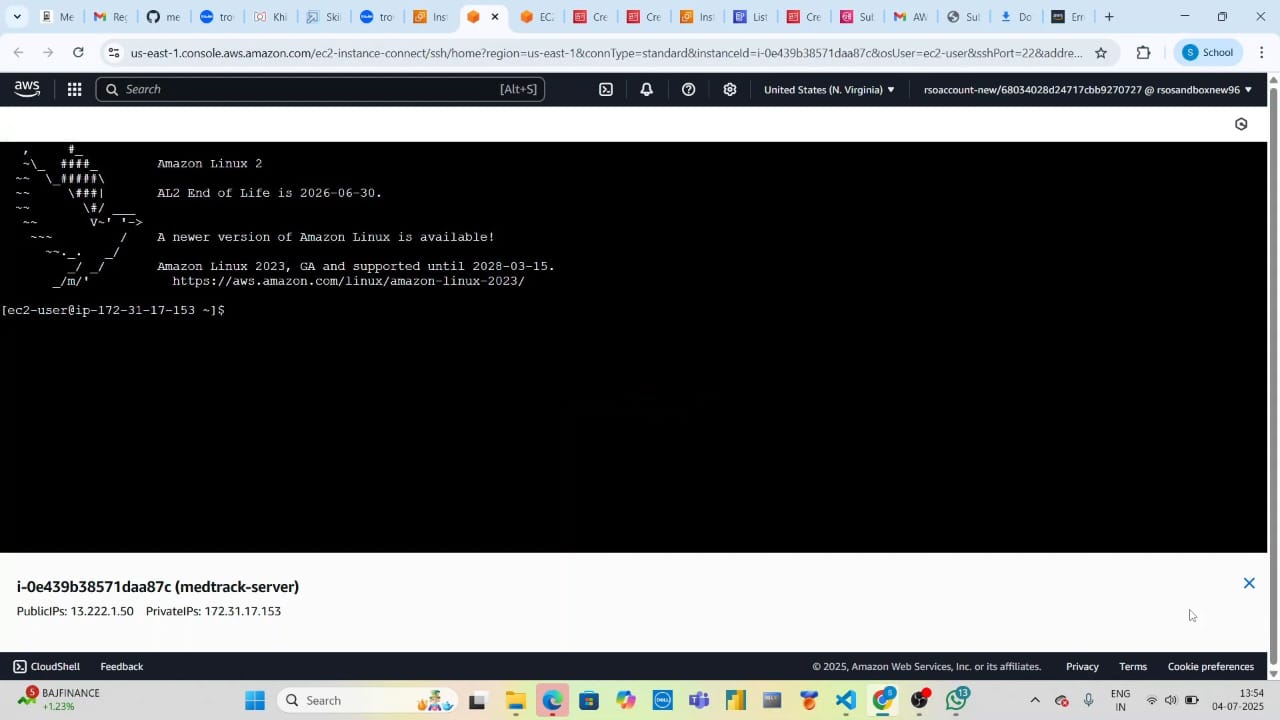


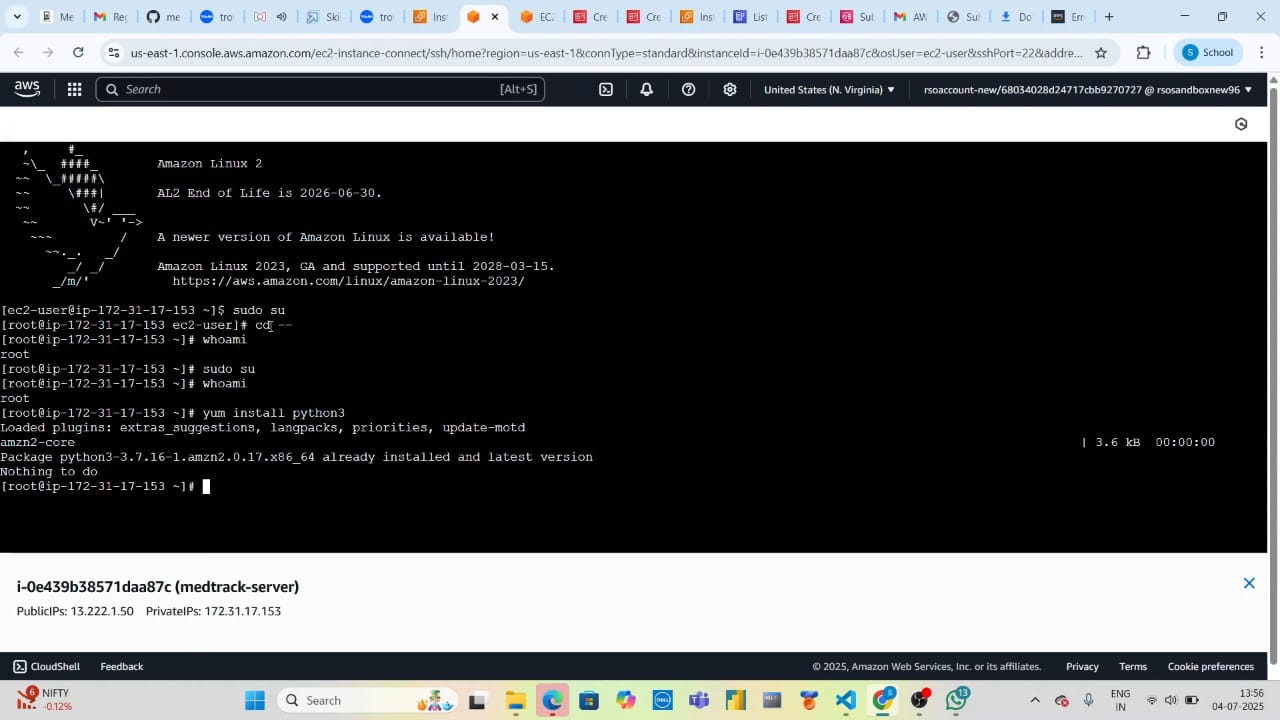
5.Configure and Launch Flask Application

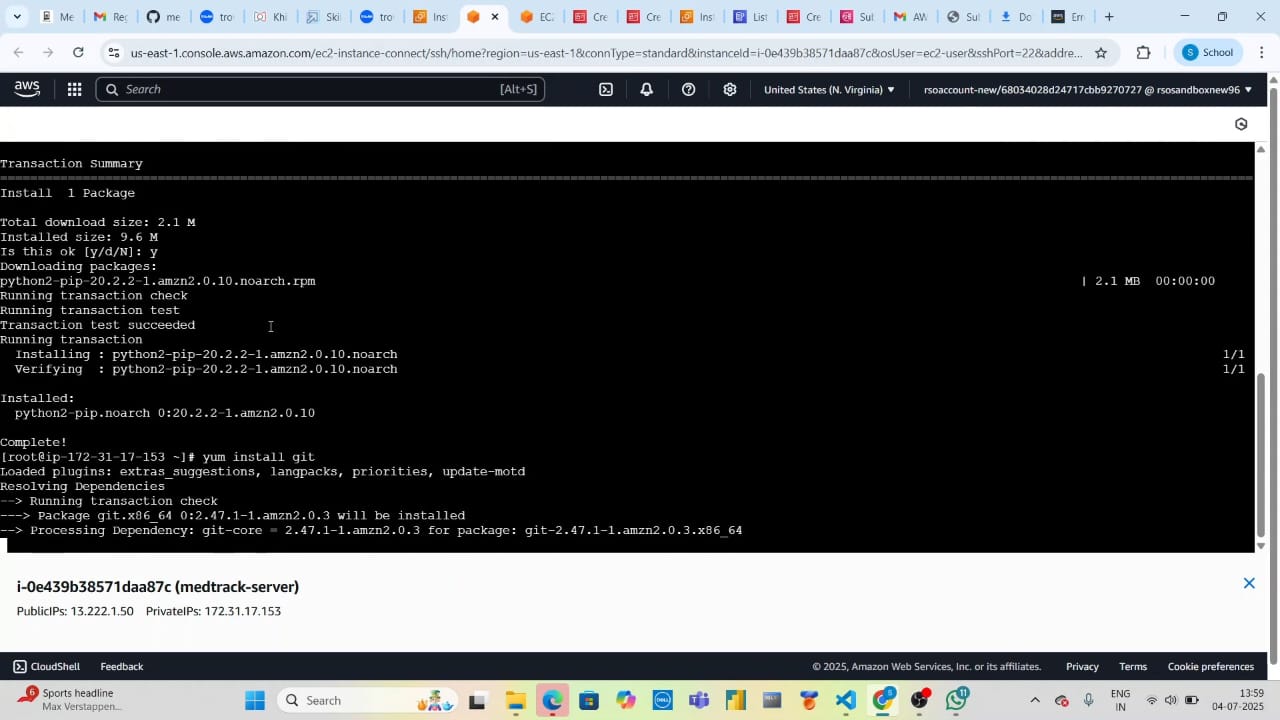
* SSH into the instance
* Install Python, Git, and Pip
* Clone the GitHub repo
* Set up .env file with AWS credentials
* Run the Flask app

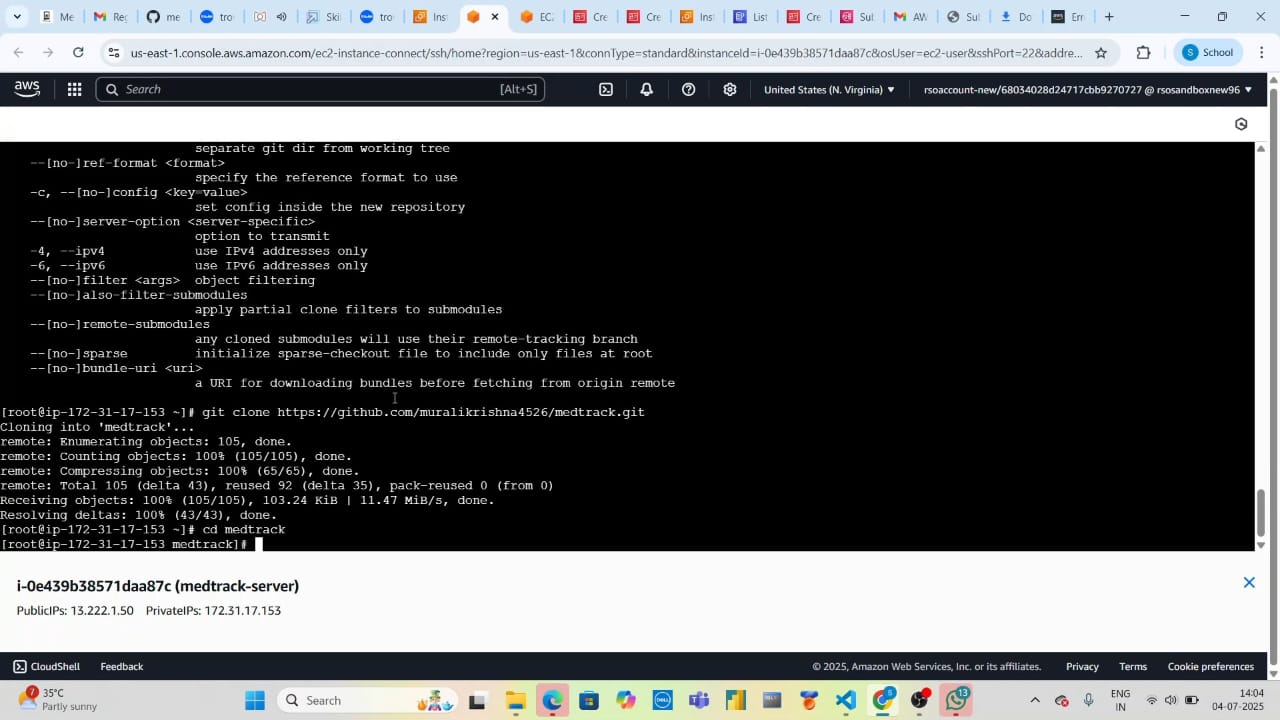


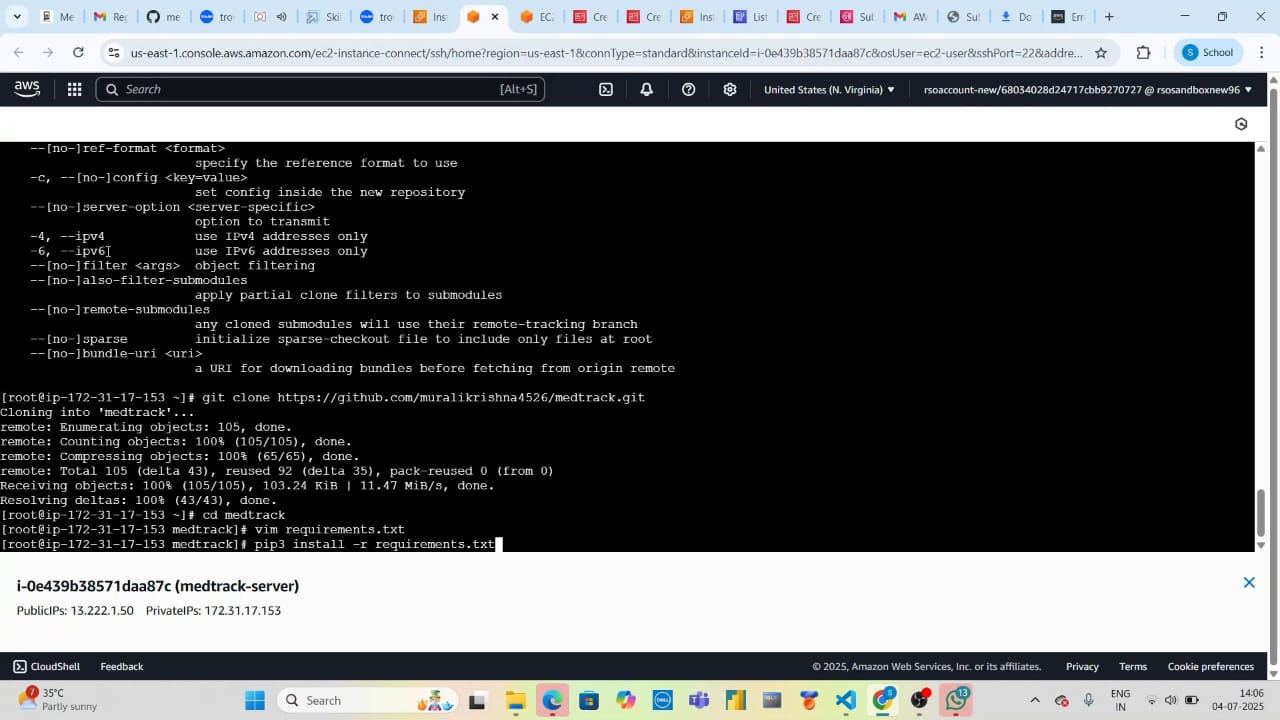


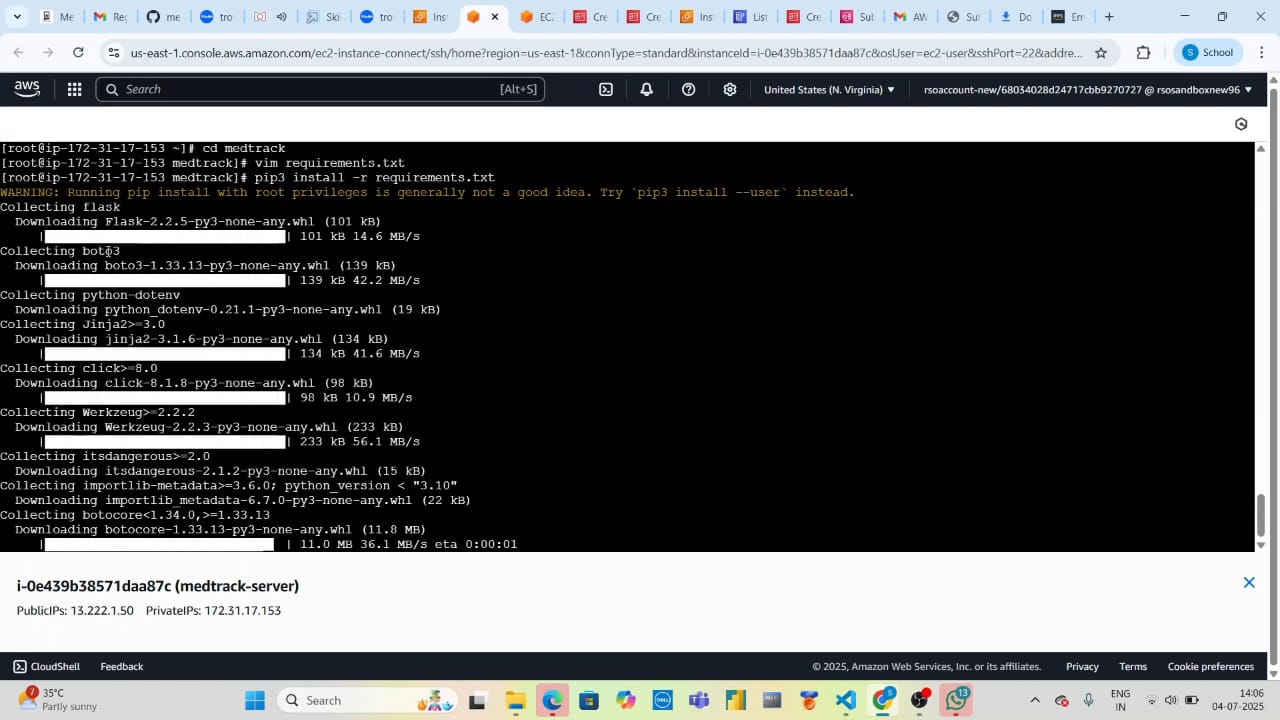


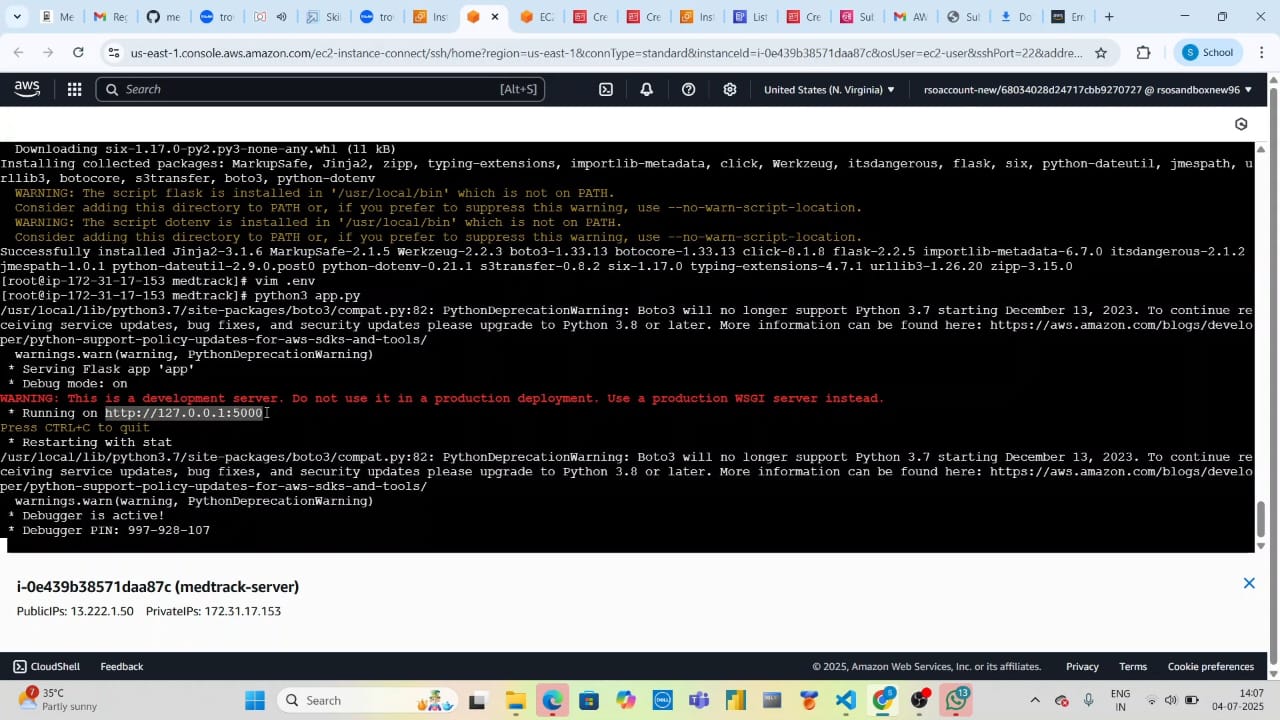






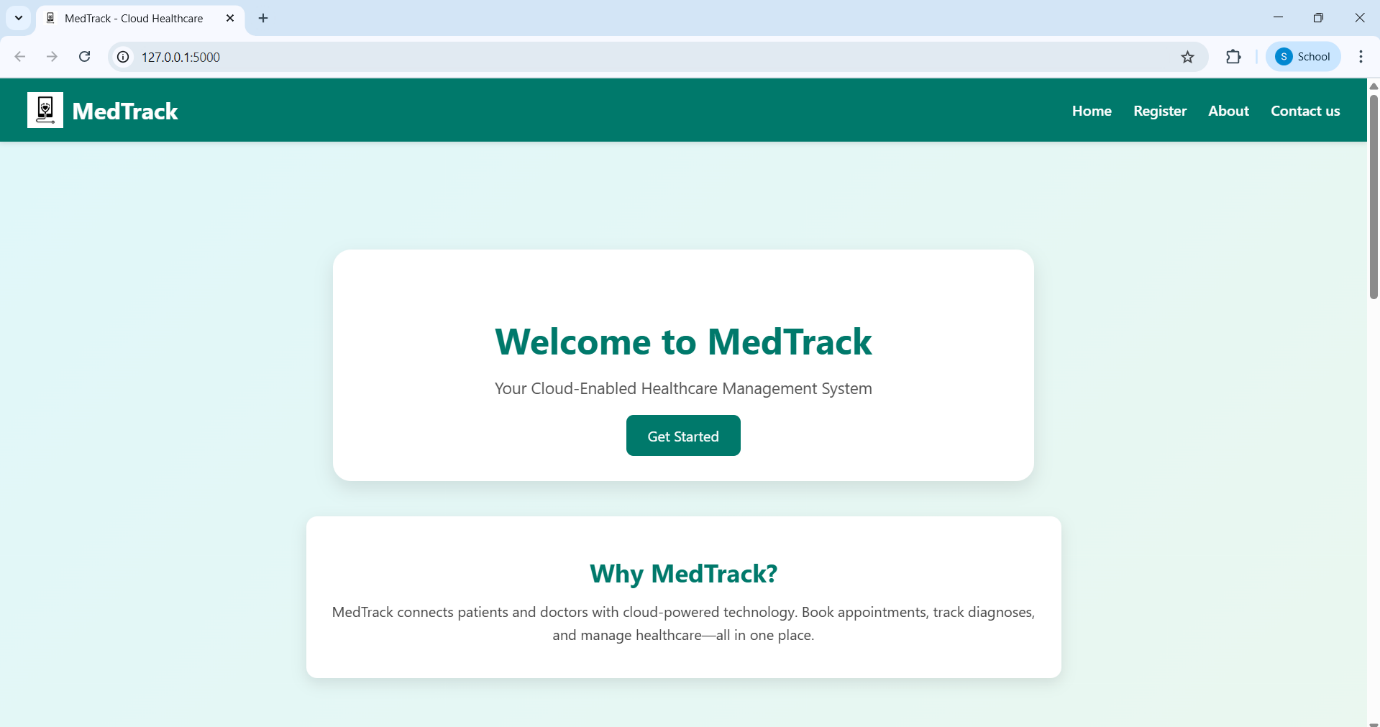


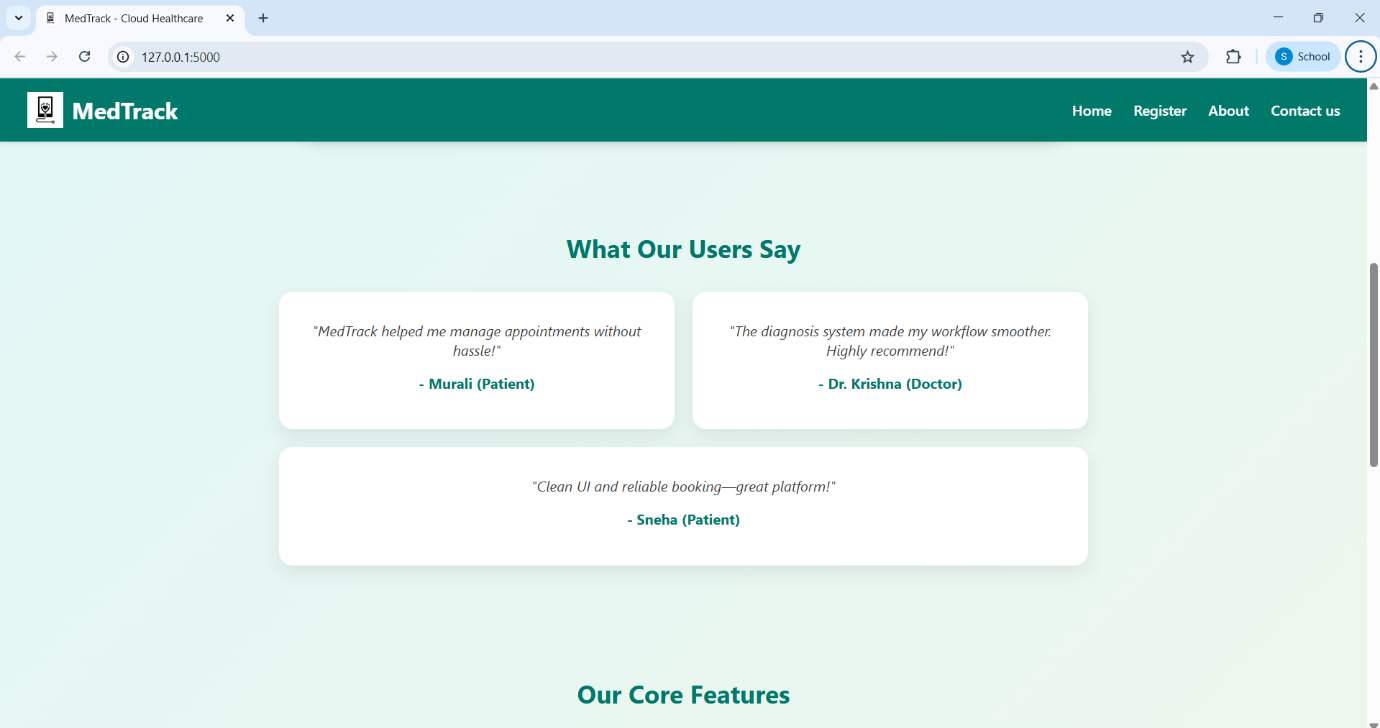


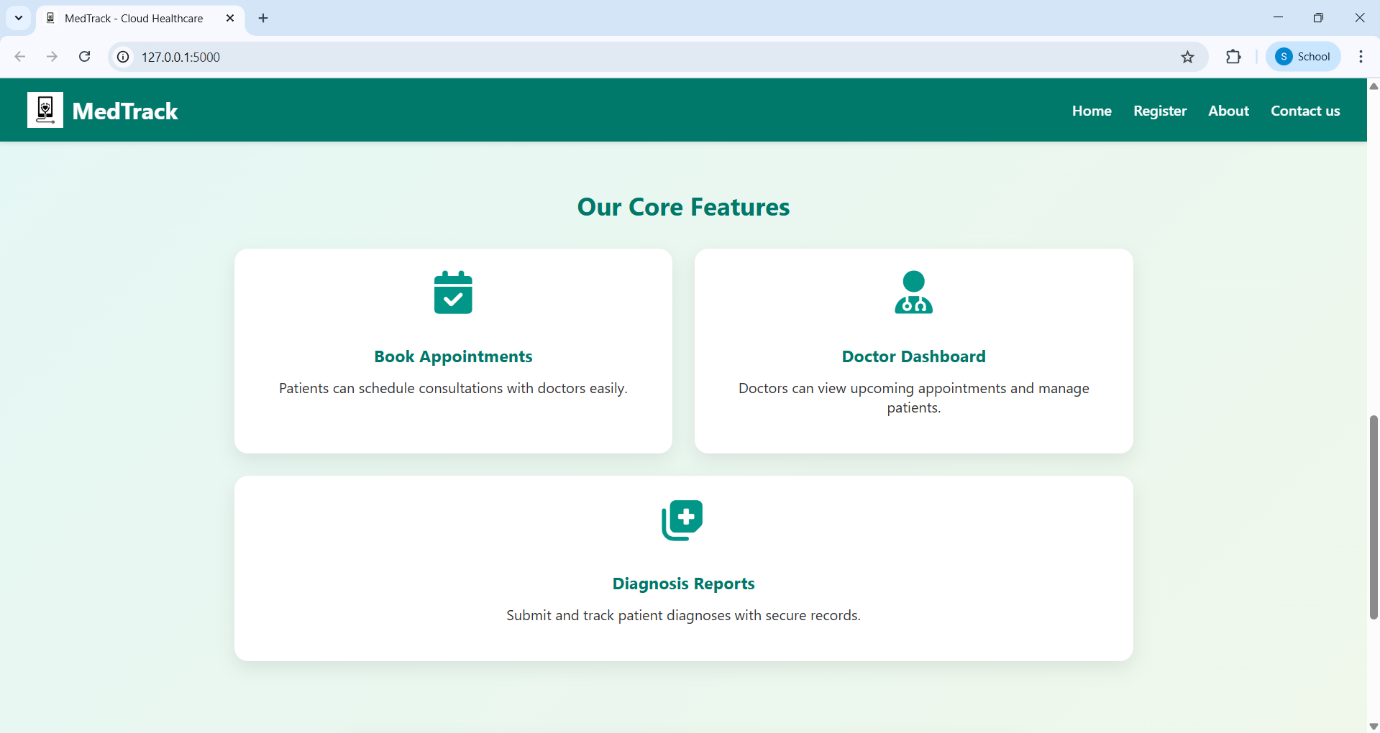


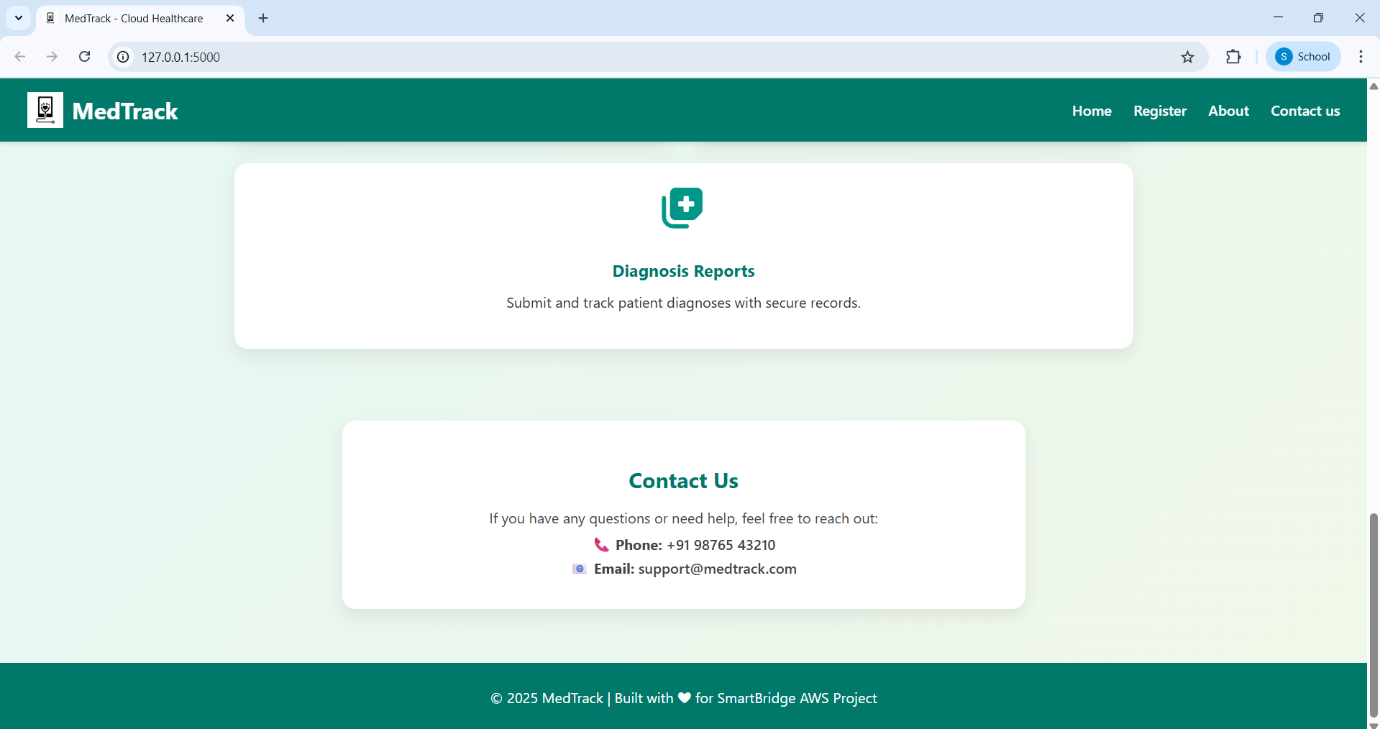
**📷 Screenshots**

1.Homepage

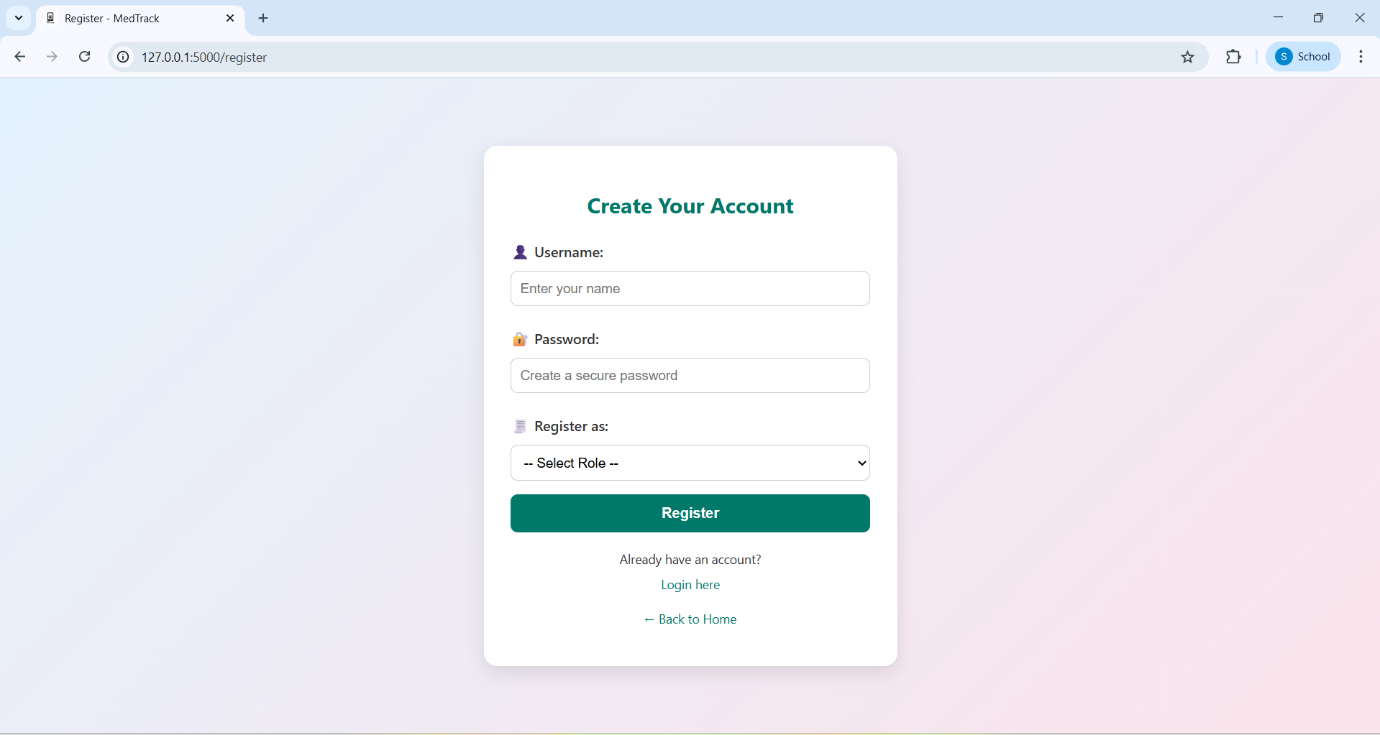


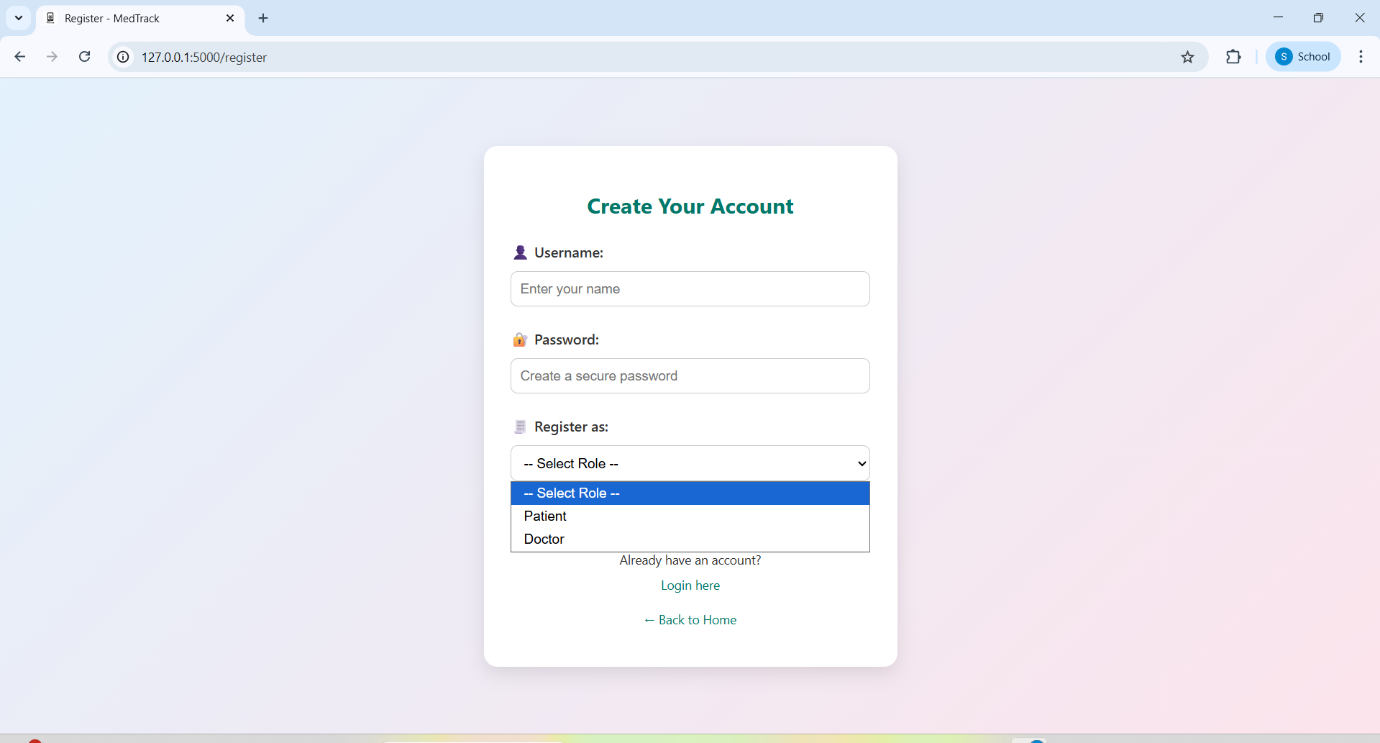




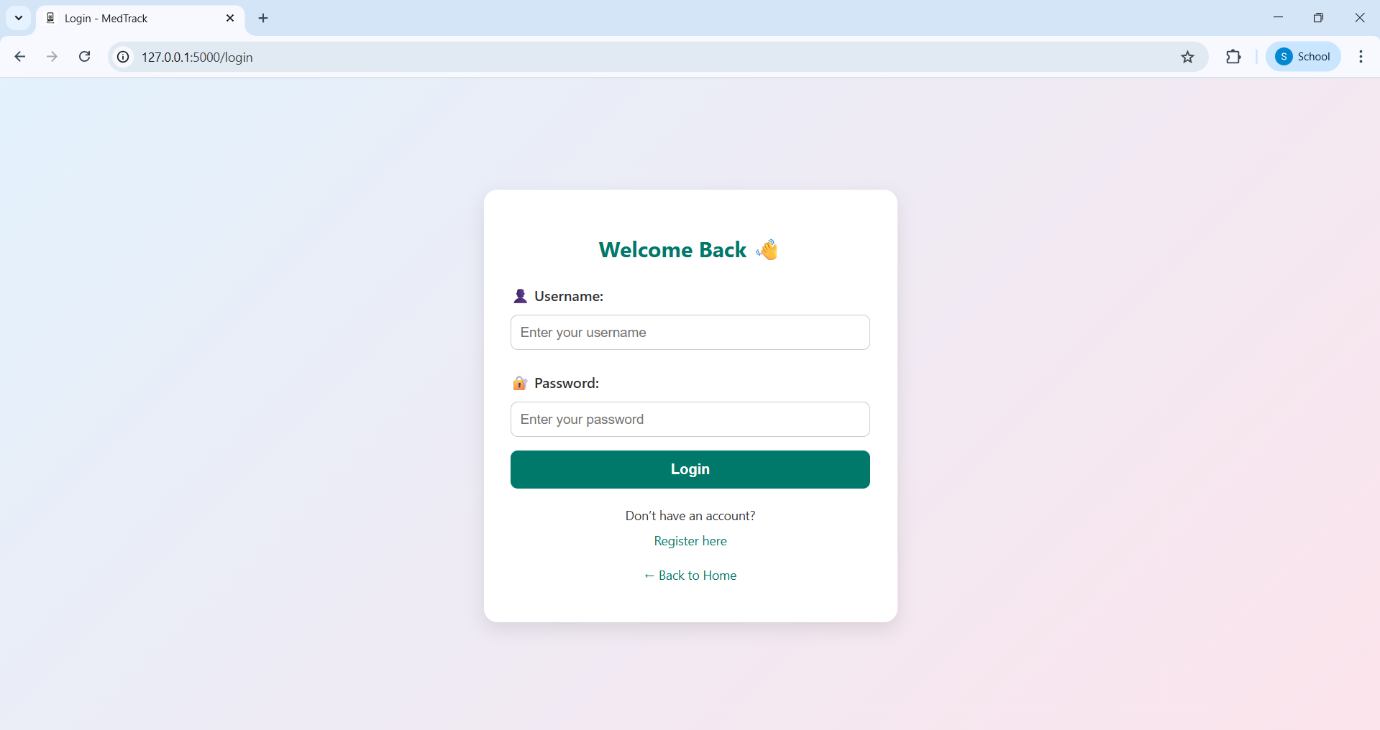


2.Register

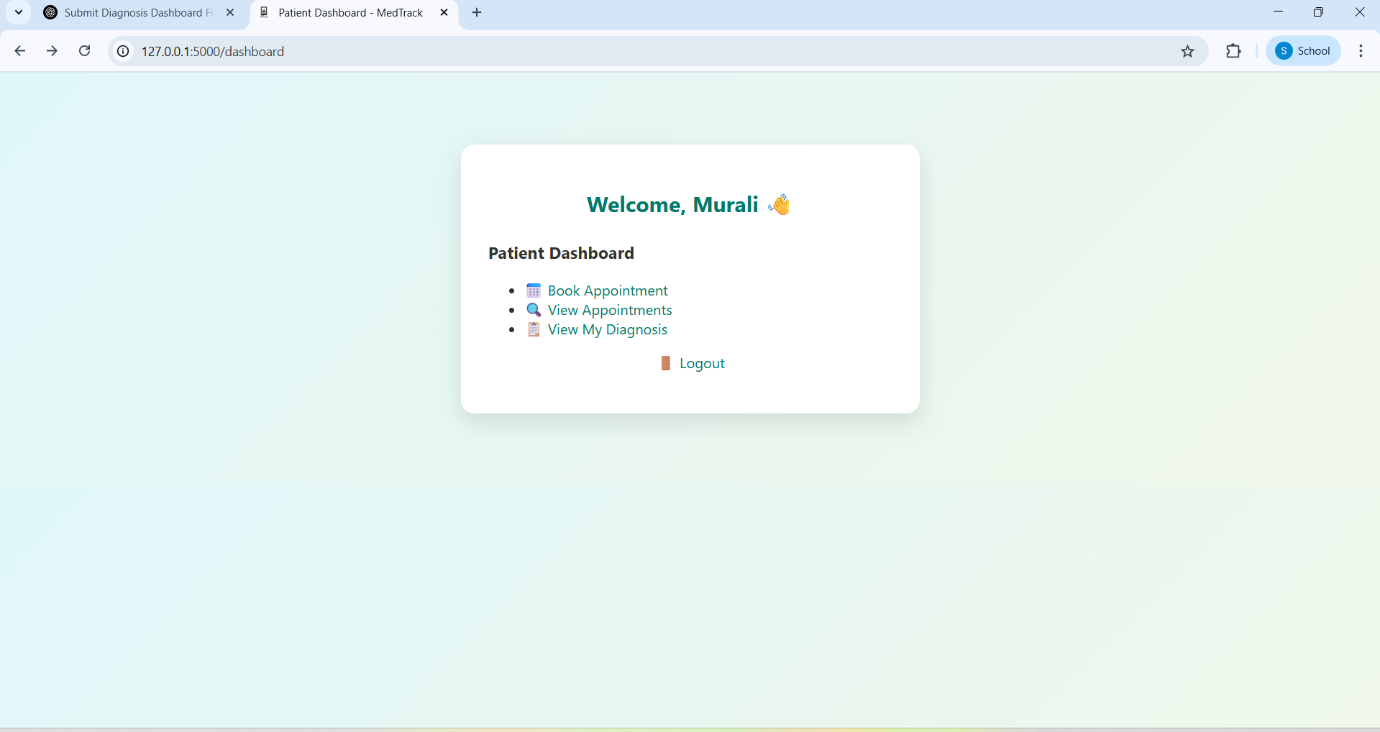




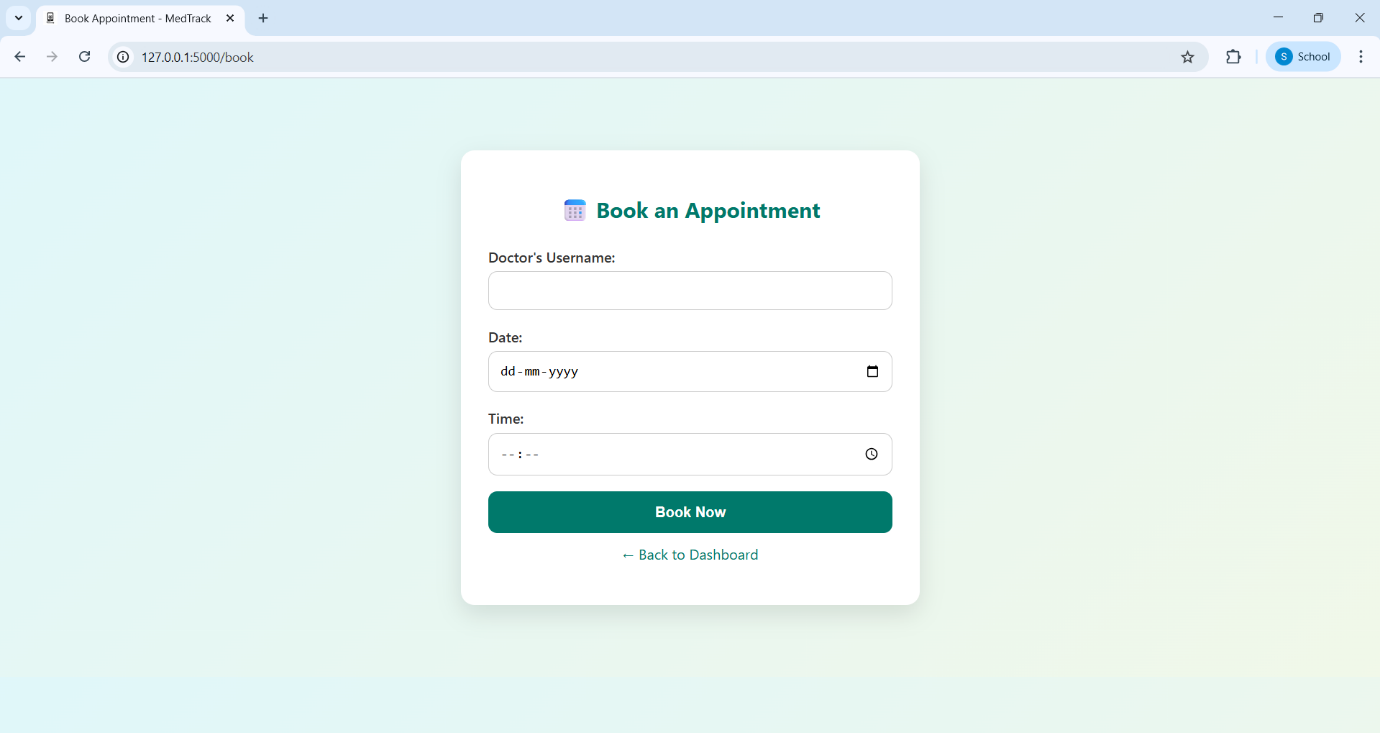
3.Login



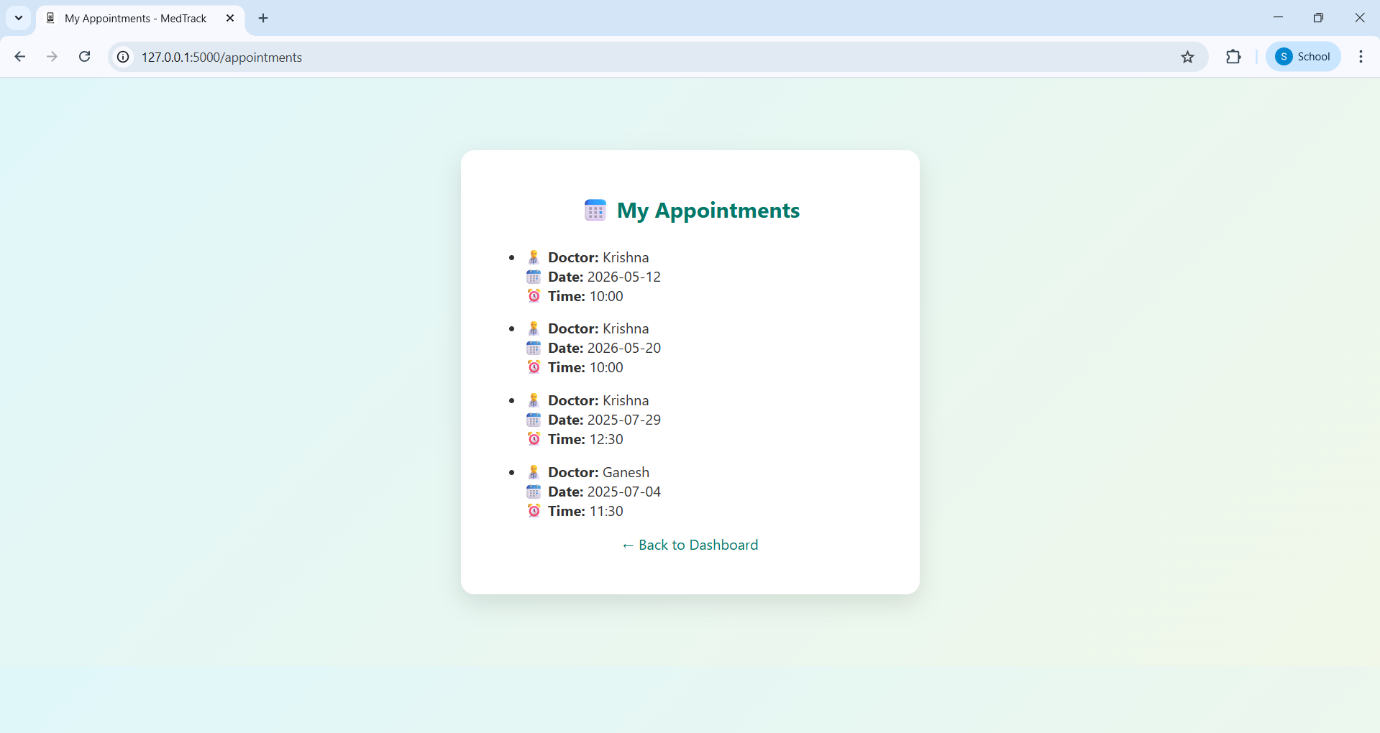
4.Patient Dashboard



5.Appointment Booking Form



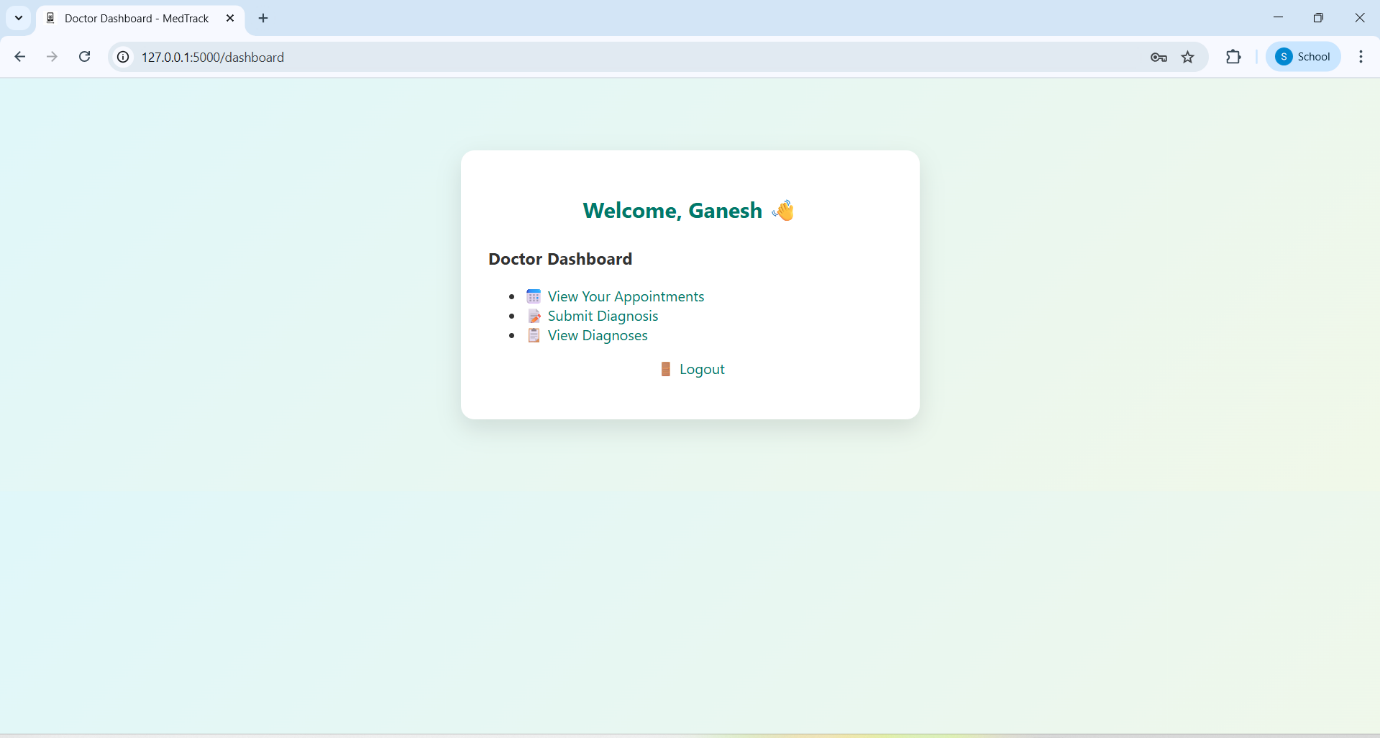
6.View Appointments



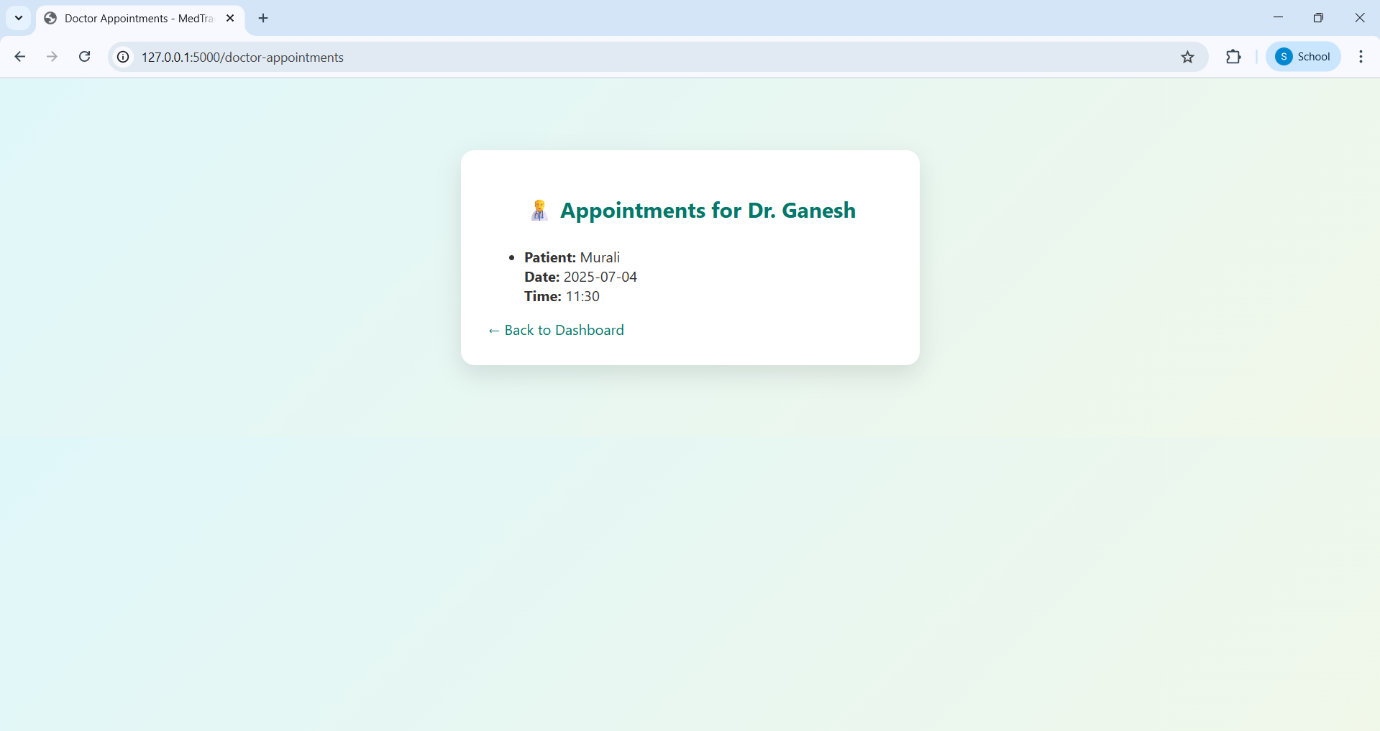
7.View My Diagnosis



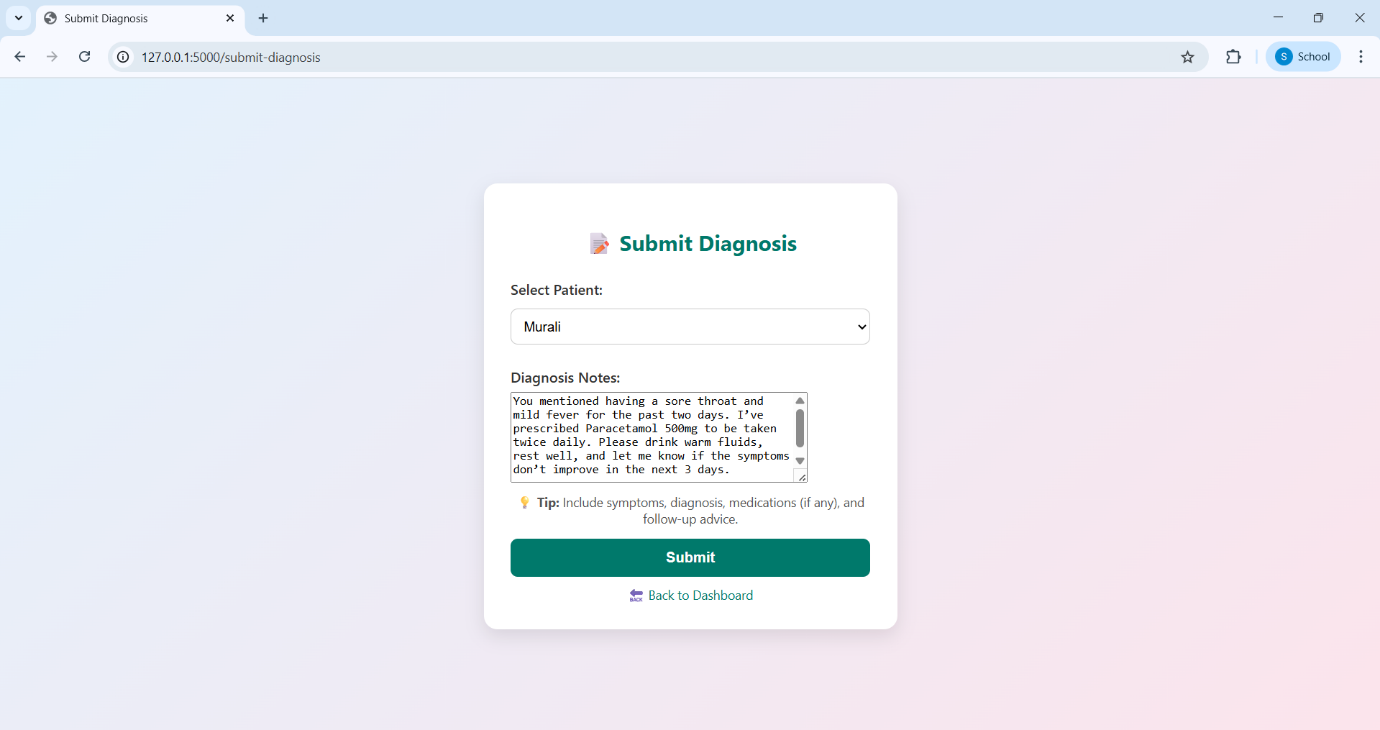
8.Doctor Dashboard

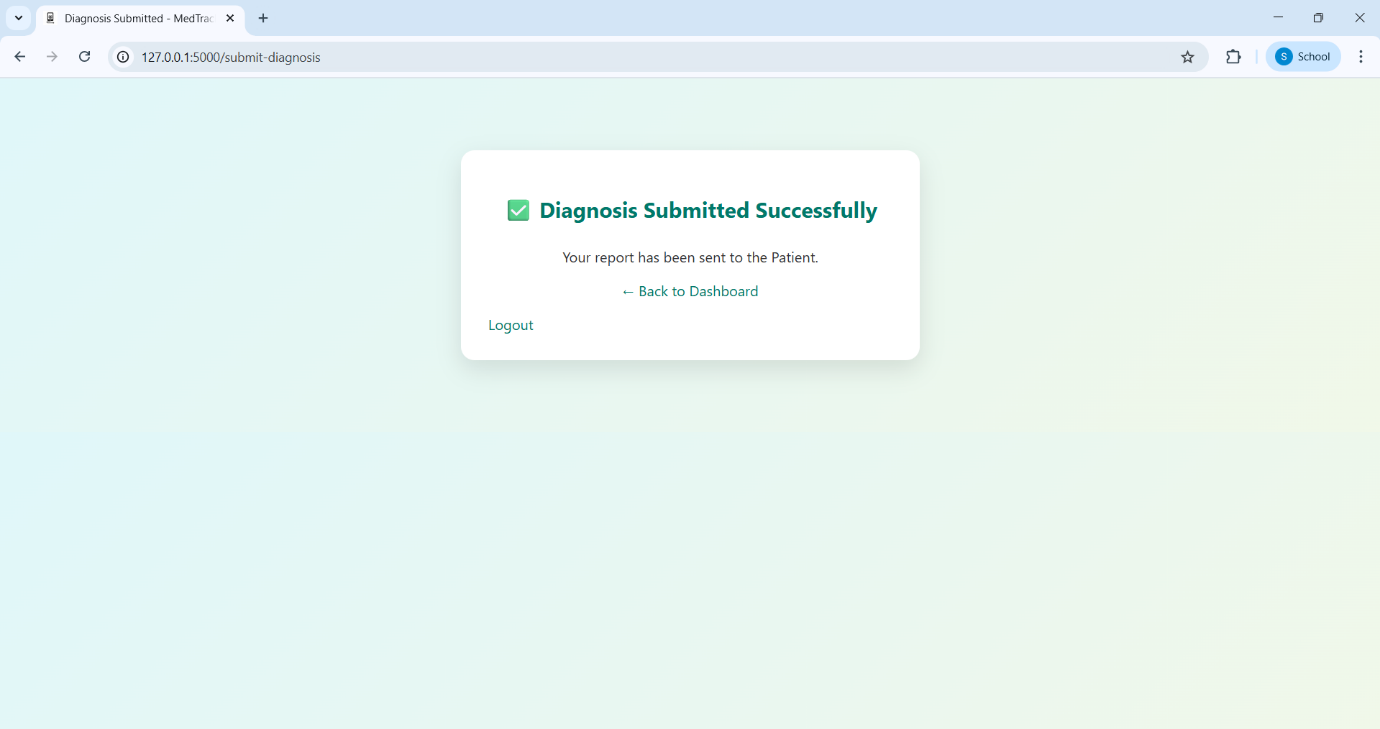


9.View Your Appointments

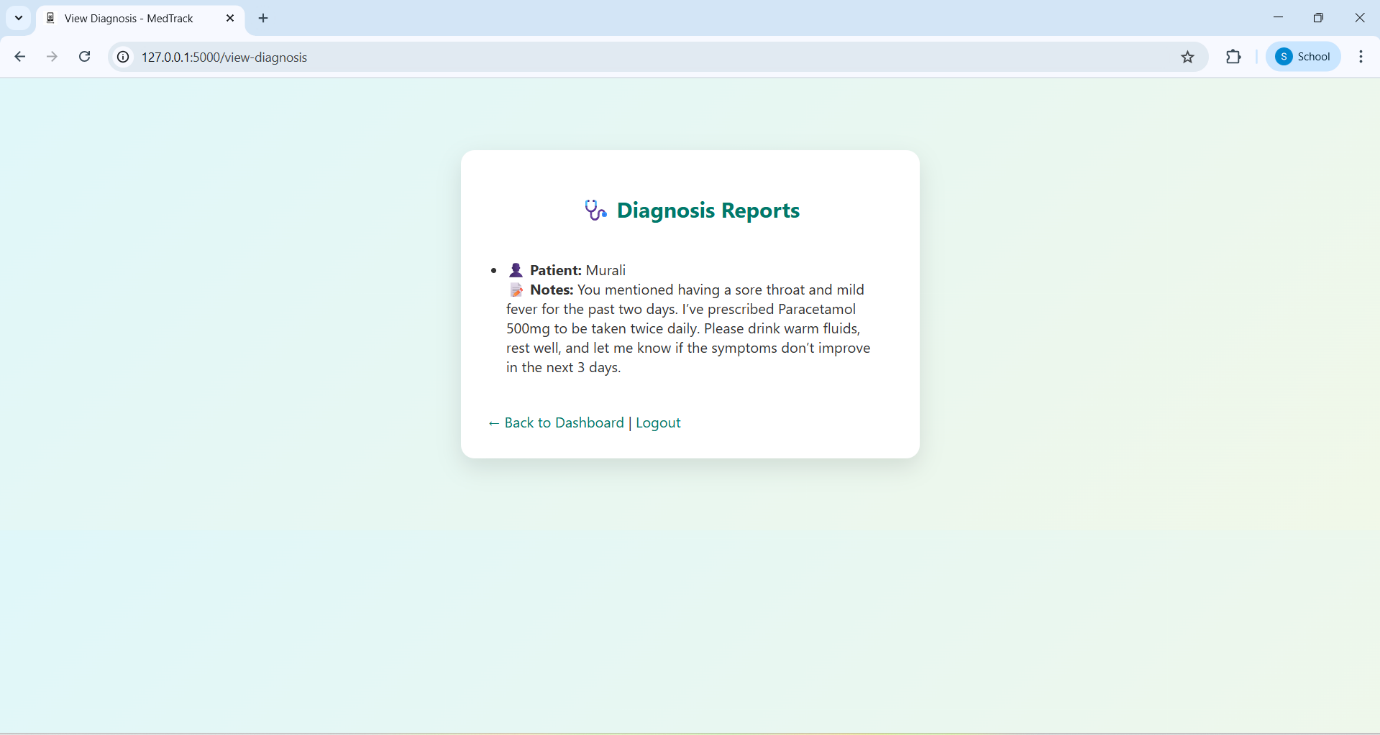


10.Submit Daignosis





11.View Diagnosis



**Demo Video**

📌<https://drive.google.com/file/d/1tyDcfKP3ZWG6uDnCSghopN0xWIw0KHlM/view?usp=drivesdk>

**🔗 GitHub Repository**

<https://github.com/VANGARA-VENKATESH/MedTrack>

**✅ Conclusion**

MedTrack is a secure, cloud-ready solution for managing patient appointments and diagnoses. It demonstrates full-stack development with real AWS services and follows a clean, modular design.

**Prepared by**: VANGARA VENKATESH

**Roll No:** 228X1A42H7

**Branch :** AI&ML

**College Name :** KHIT

**SmartBridge AWS Internship Project - 2025**