



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

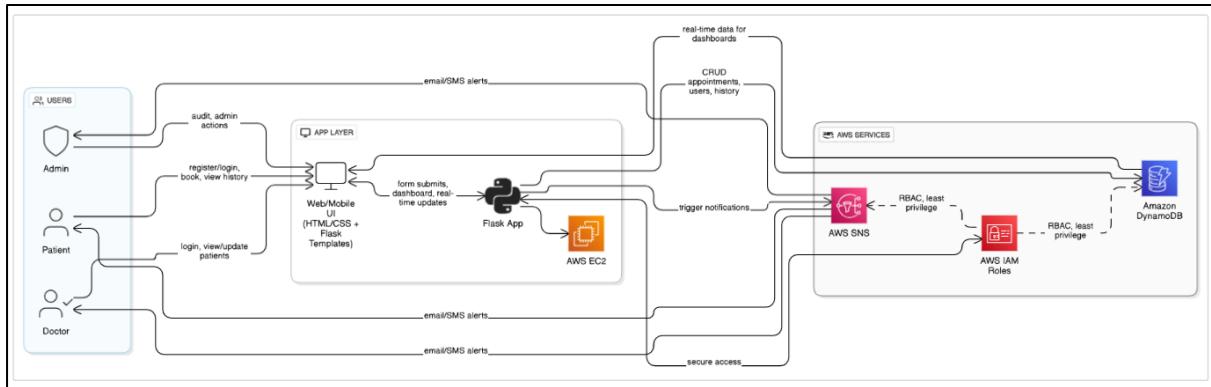
- Secure login & role-based access (Patient / Doctor)
- Book and view appointments
- Submit and view diagnosis reports
- Data stored in AWS DynamoDB
- Notifications via AWS SNS
- Deployed using AWS EC2



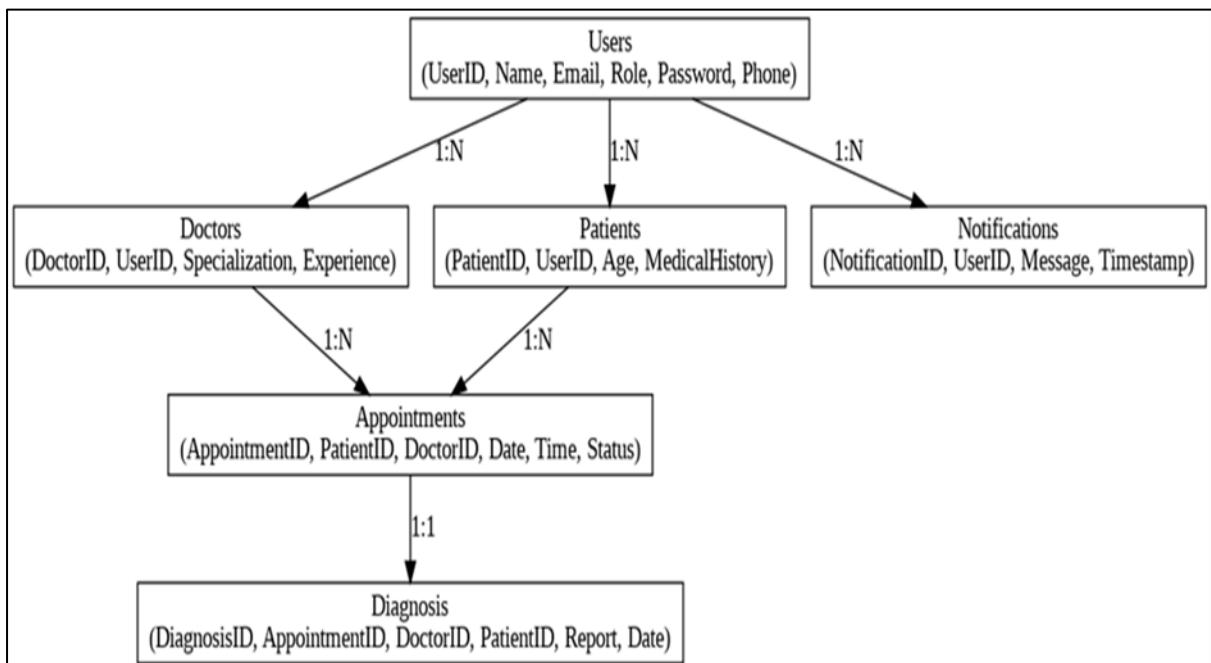
Tech Stack

- 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:

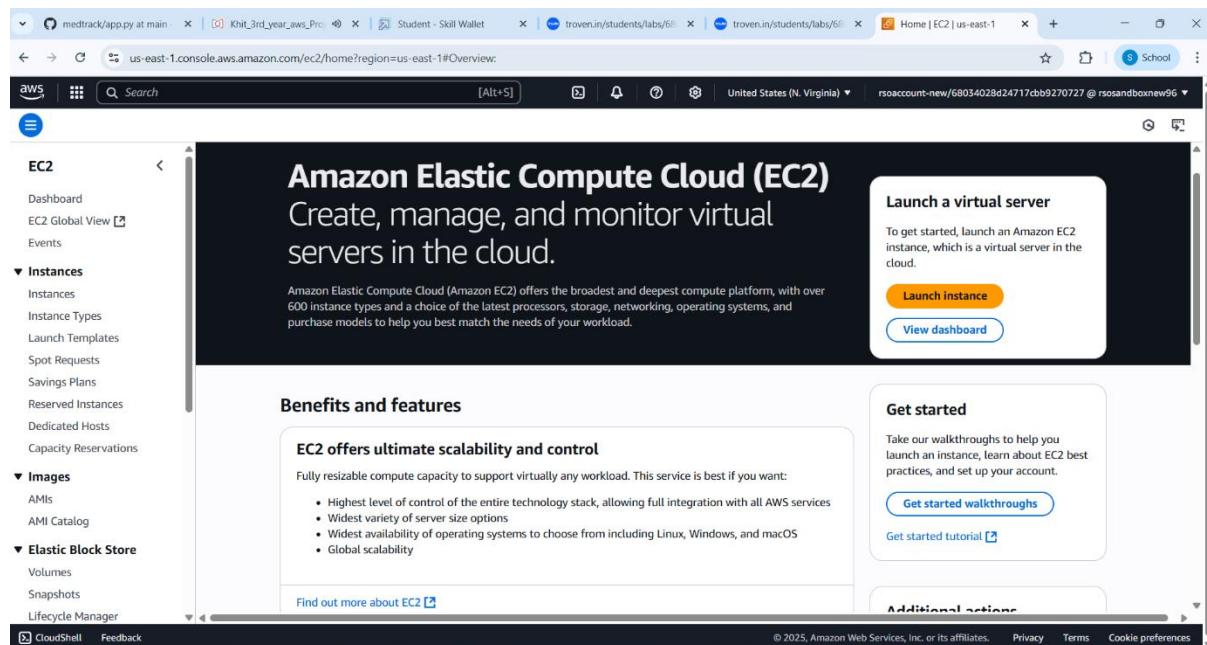
```
pip install flask boto3 python-dotenv
```

2. Run the Flask server:

```
python app.py
```

☁️ Deployment Steps

1. Launch EC2 Instance



The screenshot shows the AWS EC2 home page. On the left, there's a sidebar with navigation links for EC2 (Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store), AMIs, and AMI Catalog. The main content area has a dark header: "Amazon Elastic Compute Cloud (EC2)" and "Create, manage, and monitor virtual servers in the cloud." Below the header, there's a paragraph about EC2's features and a "Launch instance" button. To the right, there's a "Get started" section with a "Get started walkthrough" button. At the bottom, there are copyright and legal links.

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. The first step, 'Set instance details', is displayed. Key fields include:

- Name and tags**: Name is set to 'medtrack-server'.
- Application and OS Images (Amazon Machine Image)**: A search bar is present, and the 'Quick Start' section shows various AMI icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian.
- Summary**: Shows 1 instance selected, AMI 'Amazon Linux 2 Kernel 5.10 AMI...', instance type 't2.micro', and 1 volume(s) - 8 GiB.
- Buttons**: 'Launch instance' (orange button) and 'Preview code'.

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console, step 2: 'Configure instance details'. The configuration includes:

- Amazon Machine Image (AMI)**: Selected AMI is 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' (ami-000ec6c25978d5999).
- Description**: Details about the selected AMI.
- Architecture**: 64-bit (x86).
- AMI ID**: ami-000ec6c25978d5999.
- Publish Date**: 2025-06-20.
- Username**: ec2-user (verified provider).
- Instance type**: t2.micro (selected), Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true.
- Key pair (login)**: A note about selecting a key pair for secure connection.
- Summary**: Shows 1 instance selected, AMI 'Amazon Linux 2 Kernel 5.10 AMI...', instance type 't2.micro', and 1 volume(s) - 8 GiB.
- Buttons**: 'Launch instance' (orange button) and 'Preview code'.

Screenshot of the AWS EC2 'Launch an instance' wizard, step 3: 'Create key pair'.

Create key pair

Key pair name: medtrack-server
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type: RSA (selected) vs ED25519

Private key file format: .pem (selected) vs .ppk

Warning: When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Buttons: Cancel, Create key pair

Summary: Number of instances: 1, Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI..., Virtual server type (instance type): t2.micro, Firewall (security group): New security group, Storage (volumes): 1 volume(s) - 8 GiB

Bottom: Launch instance, Preview code

Screenshot of the AWS EC2 'Launch an instance' wizard, step 4: 'Inbound Security Group Rules'.

Inbound Security Group Rules:

- Security group rule 1 (TCP; 22, 157.50.88.125/32):
 - Type: ssh, Protocol: TCP, Port range: 22
 - Source type: My IP, Name: Add CIDR, prefix list or security group, Description: e.g. SSH for admin desktop, Value: 157.50.88.125/32
- Security group rule 2 (TCP; 80, 0.0.0.0/0):
 - Type: HTTP, Protocol: TCP, Port range: 80
 - Source type: Anywhere, Name: Add CIDR, prefix list or security group, Description: e.g. SSH for admin desktop, Value: 0.0.0.0/0

Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary: Number of instances: 1, Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI..., Virtual server type (instance type): t2.micro, Firewall (security group): New security group, Storage (volumes): 1 volume(s) - 8 GiB

Bottom: Launch instance, Preview code

The screenshot shows the AWS EC2 'Launch an instance' page. At the top, there is a green success message: 'Success Successfully initiated launch of instance (i-0e439b38571daa87c)'. Below this, there is a 'Launch log' button. Under 'Next Steps', there are several options: 'Create billing usage alerts', 'Connect to your instance', 'Connect an RDS database', 'Create EBS snapshot policy', 'Manage detailed monitoring', 'Create Load Balancer', 'Create AWS budget', and 'Manage CloudWatch alarms'. The 'Create billing usage alerts' section includes a 'Create billing alerts' button. The 'Connect to your instance' section includes a 'Connect to instance' button. The 'Connect an RDS database' section includes a 'Connect an RDS database' button. The 'Create EBS snapshot policy' section includes a 'Create EBS snapshot policy' button. The 'Create AWS budget' section includes a 'Create AWS budget' button. The 'Manage CloudWatch alarms' section includes a 'Create CloudWatch alarms' button. The bottom of the screen shows the Windows taskbar with various pinned icons.

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

The screenshot shows the AWS IAM Dashboard. On the left, there is a sidebar with 'Identity and Access Management (IAM)' and sections for 'Access management' (User groups, Users, Roles, Policies), 'Access reports' (Resource analysis, Unused access, Analyzer settings, Credential report), and 'Organizations activities'. The main area displays 'IAM resources' with an 'Access denied' message: 'You don't have permission to iam:GetAccountSummary. To request access, copy the following text and send it to your AWS administrator.' It also shows a 'User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717ccb9270727' and 'Action: iam:GetAccountSummary' with 'Context: no identity-based policy allows the action'. There is a 'Diagnose with Amazon Q' button. To the right, there is an 'AWS Account' section with another 'Access denied' message: 'You don't have permission to iam>ListAccountAliases. To request access, copy the following text and send it to your AWS administrator.' It also shows a 'User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717ccb9270727' and 'Action: iam>ListAccountAliases' with 'Context: no identity-based policy allows it'. There is a 'Diagnose with Amazon Q' button. At the bottom, there is a 'Tools' section with a 'Policy simulator' button. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with navigation links like Dashboard, Access management, and Access reports. The main area displays a table titled "Roles (12)" with columns for Role name, Trusted entities, and Last activity. The table lists various AWS service roles and a custom role named "EC2_MedTrack_Role".

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Service-Linked Role)	140 days ago
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	140 days ago
AWSServiceRoleForECS	AWS Service: ecs (Service-Linked Role)	136 days ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	212 days ago
AWSServiceRoleForSSO	AWS Service: sso (Service-Linked Role)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
EC2_MedTrack_Role	AWS Service: ec2	-
OrganizationAccountAccessRole	Account: 058264256896	1 hour ago
rsoaccount-new	Account: 058264256896	6 minutes ago

The screenshot shows the "Create role" wizard at Step 1: Select trusted entity. It has three steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The current step is Step 3. The "Role details" section allows entering a role name (set to "EC2_MedTrack_Role") and a description (set to "Allows EC2 instances to call AWS services on your behalf"). The "Trust policy" section shows a JSON-based trust policy:

```
1 - {  
2 -     "Version": "2012-10-17",  
3 -     "Statement": [  
4 -         {  
5 -             "Effect": "Allow",  
6 -             "Action": [  
7 -                 "sts:AssumeRole"  
8 -             ]  
9 -         }  
10 -    ]  
11 -}
```

The screenshot shows the 'Create role' wizard in the AWS IAM console. The user is on Step 2: Add permissions. They have attached two AWS managed policies: 'AmazonDynamoDBFullAccess' and 'AmazonSNSFullAccess'. The interface includes a search bar, a toolbar with icons for copy, paste, and refresh, and a bottom navigation bar with 'Cancel', 'Previous', and 'Create role' buttons.

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonSNSFullAccess	AWS managed	Permissions policy

The screenshot shows the 'Modify IAM role' wizard in the AWS EC2 console. The user is attaching an IAM role to an instance. They have selected the 'EC2_MedTrack_Role' from a dropdown menu. The interface includes a search bar, a toolbar with icons for copy, paste, and refresh, and a bottom navigation bar with 'Cancel' and 'Update IAM role' buttons.

The screenshot shows the AWS EC2 Instances page. At the top, a green success message says "Successfully attached EC2_MedTrack_Role to instance i-0ffbd525ddb91fbb". Below it, the "Instances (1/3) Info" section lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
medtrack-server	i-0e439b38571daa87c	Running	t2.micro	us-east-1c	ec2-13-2-
medtrack-server	i-057d882a0af549b8c	Running	t2.micro	us-east-1c	ec2-54-8-
medtrack-server	i-0ffbd525ddb91fbb	Running	t2.micro	us-east-1c	ec2-3-90-

The selected instance, i-0ffbd525ddb91fbb (medtrack-server), is shown in more detail below. It has a Public IPv4 address of 3.90.103.136 and a Private IPv4 address of 172.31.28.184. Its instance state is Running.

3. Setup DynamoDB Tables: Users, Appointments, Diagnoses

The screenshot shows the AWS DynamoDB service homepage. On the left, a sidebar menu includes options like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, Settings, DAX (Clusters, Subnet groups, Parameter groups, Events), CloudShell, and Feedback.

The main content area features a large heading "Amazon DynamoDB" with the subtext "A fast and flexible NoSQL database service for any scale". It includes a "Get started" button and a "Pricing" section. The "How it works" section contains a video thumbnail with the text "What is Amazon DynamoDB? | Amazon Web Ser...".

A blue banner at the top encourages users to "Share your feedback on Amazon DynamoDB".

The screenshot shows the AWS DynamoDB console with the 'Tables' page open. The left sidebar includes links for Dashboard, Tables (selected), Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Under 'DAX', there are links for Clusters, Subnet groups, Parameter groups, and Events. The main area displays a table with columns: Name, Status, Partition key, Sort key, Indexes, Replication Regions, Deletion protection, Favorite, Read capacity mode, and Write capacity mode. A search bar at the top allows filtering by Name, Tag key, and Tag value. A feedback survey banner at the top right encourages users to share their feedback.

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The first step, 'Table name', is completed with the table name 'Users'. The second step, 'Partition key', is completed with the partition key 'username' of type String. The third step, 'Sort key - optional', is partially completed with the sort key 'Enter the sort key name' of type String. The fourth step, 'Table settings', is selected and shows two options: 'Default settings' (selected) and 'Customize settings'. The 'Default settings' section describes the fastest way to create the table. The bottom of the screen shows a Windows taskbar with various pinned icons and system status.

The screenshot shows the 'Create table' step in the AWS DynamoDB console. It displays several configuration options:

Setting	Value
Maximum write capacity units	-
Local secondary indexes	-
Global secondary indexes	-
Encryption key management	AWS owned key
Deletion protection	Off
Resource-based policy	Not active

Below the table, there is a section for 'Tags' with a note that no tags are associated with the resource. A button to 'Add new tag' is present.

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

[Add new tag](#)

You can add 50 more tags.

This table will be created with auto scaling deactivated. You do not have permissions to turn on auto scaling.

[Cancel](#)

[Create table](#)

The screenshot shows the 'Create table' step in the AWS DynamoDB console. A blue banner at the top encourages users to share feedback on Amazon DynamoDB.

Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

[Diagnoses](#)

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

I

String

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

String

1 to 255 characters and case sensitive.

The screenshot shows the 'Create table' step in the AWS DynamoDB console. A blue banner at the top encourages users to share feedback on Amazon DynamoDB.

Screenshot of the AWS DynamoDB Create Table page:

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Appointments

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

appointment_id String

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name String

1 to 255 characters and case sensitive.

CloudShell **Feedback** © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:56 04-07-2025

Screenshot of the AWS DynamoDB Tables page:

DynamoDB

- Dashboard
- Tables**
 - Explore Items
 - PartQL editor
 - Backups
 - Exports to S3
 - Imports from S3
 - Integrations New
 - Reserved capacity
 - Settings
- DAX**
 - Clusters
 - Subnet groups
 - Parameter groups
 - Events

Tables (3) Info

Share your feedback on Amazon DynamoDB. Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Notifications 0 0 1 1 0 0

<input type="checkbox"/>	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read c
<input type="checkbox"/>	Appointments	Active	appointment_id (\$)	-	0 0	Off	<input checked="" type="checkbox"/>	<input type="checkbox"/>	On-der
<input type="checkbox"/>	Diagnoses	Active	diagnosis_id (\$)	-	0 0	Off	<input checked="" type="checkbox"/>	<input type="checkbox"/>	On-der
<input type="checkbox"/>	Users	Active	username (\$)	-	0 0	Off	<input checked="" type="checkbox"/>	<input type="checkbox"/>	On-der

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:56 04-07-2025

4. Configure SNS

The screenshot shows the AWS Simple Notification Service (SNS) homepage. At the top, there is a blue banner with the text "Amazon SNS now supports High Throughput FIFO topics. Learn more". Below the banner, the title "Amazon Simple Notification Service" is displayed, followed by the subtitle "Pub/sub messaging for microservices and serverless applications". A paragraph of text describes Amazon SNS as a highly available, durable, secure, fully managed pub/sub messaging service. To the right, a "Create topic" dialog box is open, asking for a "Topic name" (with "MyTopic" entered). There are "Next step" and "Start with an overview" buttons. Below the dialog, a "Pricing" section is visible, stating that Amazon SNS has no upfront costs and is pay-based on message volume. The bottom of the page includes a navigation bar with links like CloudShell, Feedback, and a search bar.

The screenshot shows the "Topics" page within the AWS SNS service. On the left, a sidebar menu lists "Amazon SNS" (selected), "Dashboard", "Topics" (selected), and "Subscriptions". Under "Mobile", it shows "Push notifications" and "Text messaging (SMS)". The main content area is titled "Topics (0)" and features a search bar and a table header with columns "Name" and "Type". Below the table, a message says "No topics" and "To get started, create a topic." A prominent orange "Create topic" button is located at the bottom of this section. The bottom of the page includes a navigation bar with links like CloudShell, Feedback, and a search bar.

Screenshot of the AWS CloudShell interface showing the creation of an SNS topic named "Medtrack".

The browser address bar shows: us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/create-topic

The page title is "Create topic" under "Amazon SNS > Topics".

A blue banner at the top indicates "New Feature: Amazon SNS now supports High Throughput FIFO topics. Learn more".

An error message in a red banner states: "Error code: AccessDeniedException - Error message: User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717cbb9270727 is not authorized to perform: kms:DescribeKey on resource: arn:aws:kms:us-east-1:600627341644:key/c06920cc-9330-4a08-b273-946668080c88 because no identity-based policy allows the kms:DescribeKey action". A "Diagnose with Amazon Q" button is present.

The "Details" section shows two options:

- FIFO (first-in, first-out)**
 - Strictly-preserved message ordering
 - Exactly-once message delivery
 - Subscription protocols: SQS
- Standard**
 - Best-effort message ordering
 - At-least once message delivery
 - Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

The "Name" field is filled with "Medtrack".

The "Display name - optional" field contains "My Topic".

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 04-07-2025.

Screenshot of the AWS CloudShell interface showing the successful creation of the SNS topic "Medtrack".

The browser address bar shows: us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/topic/arn:aws:sns:us-east-1:600627341644:Medtrack

The page title is "Medtrack" under "Amazon SNS > Topics".

A green banner at the top indicates "Topic Medtrack created successfully. You can create subscriptions and send messages to them from this topic." A "Publish message" button is present.

The left sidebar shows navigation links: Dashboard, Topics, Subscriptions, Mobile (Push notifications, Text messaging (SMS)), and a CloudShell/Feedback link.

The "Medtrack" topic details are displayed:

Details	Value
Name	Medtrack
ARN	arn:aws:sns:us-east-1:600627341644:Medtrack
Type	Standard
Display name	-
Topic owner	600627341644

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 04-07-2025.

Screenshot of the AWS SNS console showing the 'Medtrack' topic details.

Details

Name	Medtrack
ARN	arn:aws:sns:us-east-1:600627341644:Medtrack
Type	Standard

Subscriptions (0)

ID	Endpoint	Status	Protocol
No subscriptions found You don't have any subscriptions to this topic.			

Actions: Edit, Delete, Publish message

Screenshot of the AWS SNS console showing the 'Create subscription' wizard.

Topic ARN: arn:aws:sns:us-east-1:600627341644:Medtrack

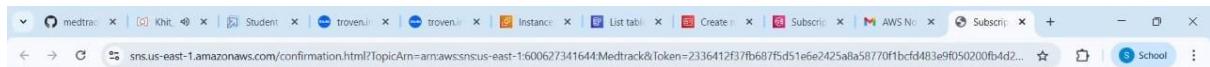
Protocol: Email

Endpoint: 228x1a4526@khitguntur.ac.in

Subscription filter policy - optional: This policy filters the messages that a subscriber receives.

Redrive policy (dead-letter queue) - optional: Send undeliverable messages to a dead-letter queue.

Actions: Cancel, Create subscription



aws
Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's ID is:
arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051

If it was not your intention to subscribe, [click here to unsubscribe](#).

The screenshot shows the AWS SNS console with a subscription details page. The subscription ARN is `arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051`. The status is Confirmed. The endpoint is `228x1a4526@khitguntur.ac.in`. The topic is `Medtrack`. The subscription principal is `arn:aws:iam::600627341644:role/rsoaccount-new`. The protocol is EMAIL. The subscription filter policy is empty, and the redrive policy (dead-letter queue) is also empty.

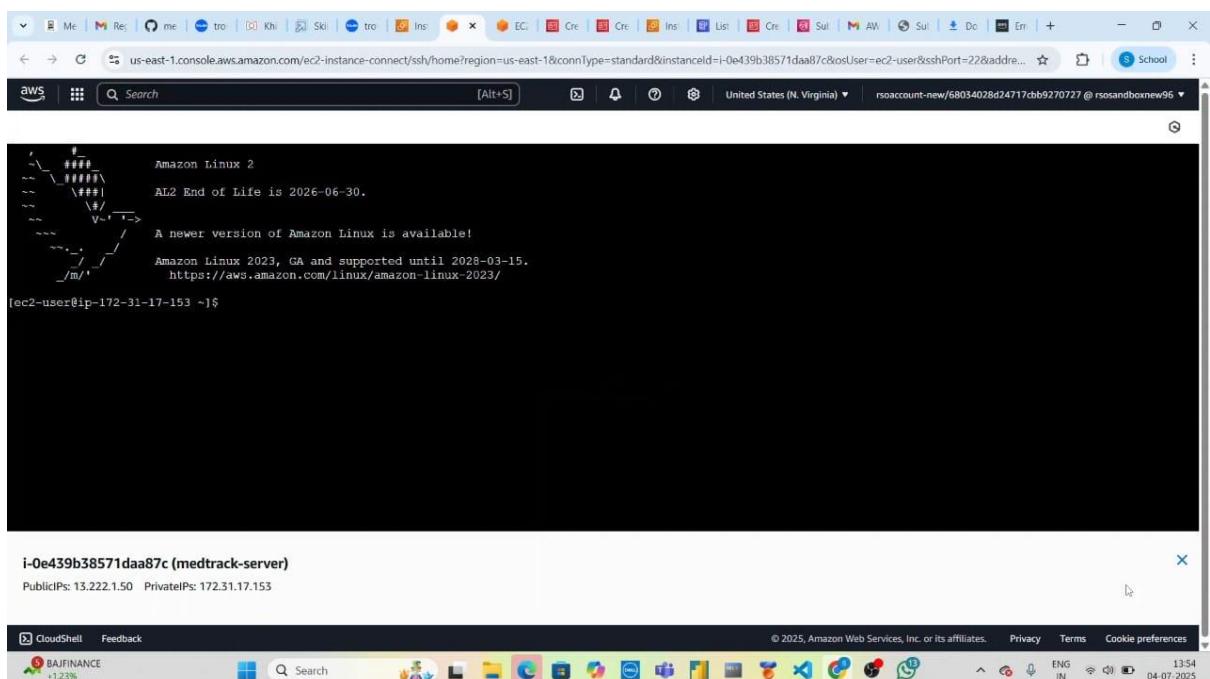
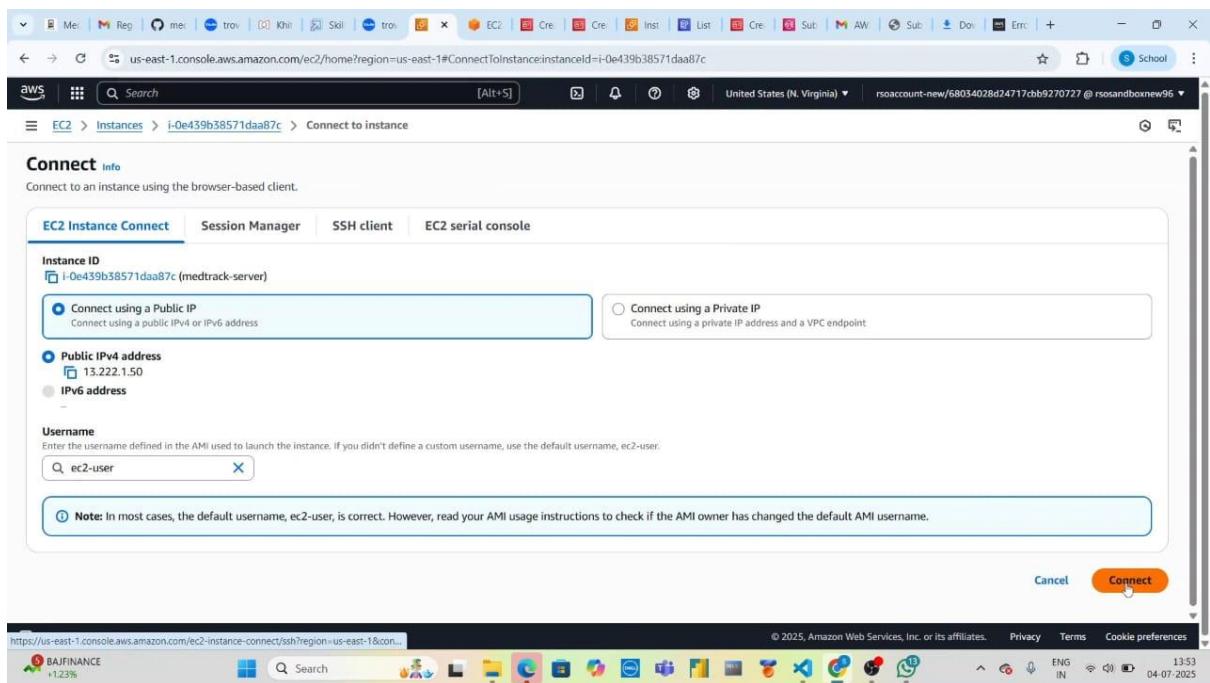
5. Configure and Launch Flask Application

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

The screenshot shows the AWS EC2 Instances page. There are four instances listed:

- medtrack-server**: Instance ID `i-0e439b38571daa87c`, State **Running**, Type **t2.micro**, Status **2/2 checks passed**, View alarms +, Availability Zone **us-east-1c**, Public IP **ec2-13-2-**.
- medtrack-server**: Instance ID `i-0694c62f9eb54760`, State **Terminated**, Type **t2.micro**, Status **-**, View alarms +, Availability Zone **us-east-1c**, Public IP **-**.
- medtrack-server**: Instance ID `i-057d882a0af549b8c`, State **Terminated**, Type **t2.micro**, Status **-**, View alarms +, Availability Zone **us-east-1c**, Public IP **-**.
- medtrack-server**: Instance ID `i-0ffbd525ddd91fb`, State **Running**, Type **t2.micro**, Status **2/2 checks passed**, View alarms +, Availability Zone **us-east-1c**, Public IP **ec2-3-90-**.

 The instance `i-0e439b38571daa87c` is selected. The Security tab is active, showing the IAM Role `EC2_MedTrack_Role`, Owner ID `600627341644`, and Launch time `Fri Jul 04 2025 11:10:58 GMT+0530 (India Standard Time)`.



```
Amazon Linux 2
AL2 End of Life is 2026-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-17-153 ~]$ sudo su
[root@ip-172-31-17-153 ~]# whoami
root
[root@ip-172-31-17-153 ~]# sudo su
[root@ip-172-31-17-153 ~]# whoami
root
[root@ip-172-31-17-153 ~]# yum install python3
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package python3-3.7.16-1.amzn2.0.17.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-17-153 ~]#
```

i-0e439b38571daa87c (medtrack-server)
PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:56 04-07-2025

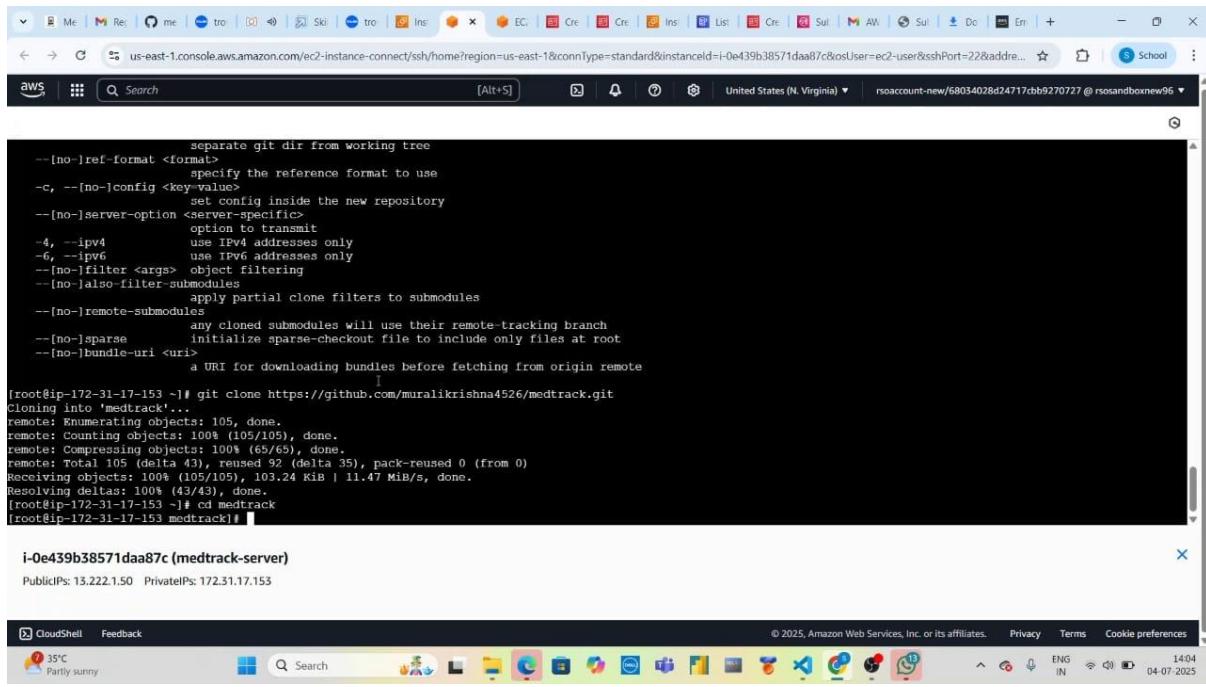
```
Transaction Summary
Install 1 Package

total download size: 2.1 M
Installed size: 9.6 M
Is this ok [y/d/N]: y
Downloading packages:
python2-pip-20.2.2-1.amzn2.0.10.noarch.rpm | 2.1 MB 00:00:00
Running transaction check
Running transaction test
transaction test succeeded
Running transaction
  Installing : python2-pip-20.2.2-1.amzn2.0.10.noarch
  Verifying   : python2-pip-20.2.2-1.amzn2.0.10.noarch
Installed:
  python2-pip.noarch 0:20.2.2-1.amzn2.0.10

Complete!
[root@ip-172-31-17-153 ~]# yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.47.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.47.1-1.amzn2.0.3 for package: git-2.47.1-1.amzn2.0.3.x86_64
```

i-0e439b38571daa87c (medtrack-server)
PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:59 04-07-2025

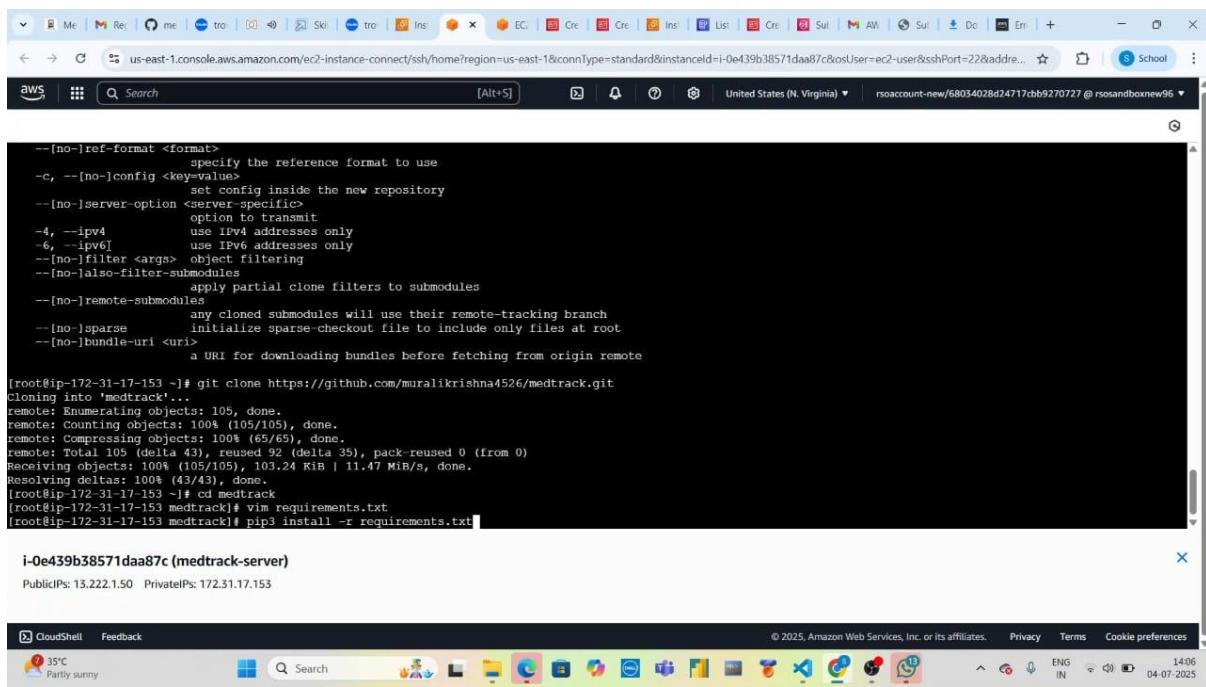


```
--[no-]ref-format <format>
    specify the reference format to use
--c, --[no-]config <key-value>
    set config inside the new repository
--[no-]server-option <server-specific>
    option to transmit
-4, --ipv4
    use IPv4 addresses only
-6, --ipv6
    use IPv6 addresses only
--[no-]filter <args>
    object filtering
--[no-]also-filter-submodules
    apply partial clone filters to submodules
--[no-]remote-submodules
    any cloned submodules will use their remote-tracking branch
--[no-]sparse
    initialize sparse-checkout file to include only files at root
--[no-]bundle-uri <uri>
    a URI for downloading bundles before fetching from origin remote

[root@ip-172-31-17-153 ~]# git clone https://github.com/muralikrishna4526/medtrack.git
Cloning into 'medtrack'...
remote: Enumerating objects: 105, done.
remote: Counting objects: 100% (105/105), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 105 (delta 43), reused 92 (delta 35), pack-reused 0 (from 0)
Receiving objects: 100% (105/105), 103.24 KiB | 11.47 MiB/s, done.
Resolving deltas: 100% (43/43), done.
[root@ip-172-31-17-153 ~]# cd medtrack
[root@ip-172-31-17-153 medtrack]#
```

i-0e439b38571daa87c (medtrack-server)
PublicIPs: 13.222.1.50 PrivatelPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 35°C Partly sunny Search ENG IN 14:06 04-07-2025



```
--[no-]ref-format <format>
    specify the reference format to use
--c, --[no-]config <key-value>
    set config inside the new repository
--[no-]server-option <server-specific>
    option to transmit
-4, --ipv4
    use IPv4 addresses only
-6, --ipv6
    use IPv6 addresses only
--[no-]filter <args>
    object filtering
--[no-]also-filter-submodules
    apply partial clone filters to submodules
--[no-]remote-submodules
    any cloned submodules will use their remote-tracking branch
--[no-]sparse
    initialize sparse-checkout file to include only files at root
--[no-]bundle-uri <uri>
    a URI for downloading bundles before fetching from origin remote

[root@ip-172-31-17-153 ~]# git clone https://github.com/muralikrishna4526/medtrack.git
Cloning into 'medtrack'...
remote: Enumerating objects: 105, done.
remote: Counting objects: 100% (105/105), done.
remote: Compressing objects: 100% (65/65), done.
remote: Total 105 (delta 43), reused 92 (delta 35), pack-reused 0 (from 0)
Receiving objects: 100% (105/105), 103.24 KiB | 11.47 MiB/s, done.
Resolving deltas: 100% (43/43), done.
[root@ip-172-31-17-153 ~]# cd medtrack
[root@ip-172-31-17-153 medtrack]# vim requirements.txt
[root@ip-172-31-17-153 medtrack]# pip3 install -r requirements.txt
```

i-0e439b38571daa87c (medtrack-server)
PublicIPs: 13.222.1.50 PrivatelPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 35°C Partly sunny Search ENG IN 14:06 04-07-2025

```
[root@ip-172-31-17-153 ~]# cd medtrack
[root@ip-172-31-17-153 medtrack]# vim requirements.txt
[root@ip-172-31-17-153 medtrack]# pip3 install -r requirements.txt
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting Flask
  Downloading Flask-2.2.5-py3-none-any.whl (101 kB)
    |████████| 101 kB 14.6 MB/s
Collecting botocore
  Downloading botocore-1.33.13-py3-none-any.whl (139 kB)
    |████████| 139 kB 42.2 MB/s
Collecting python-dotenv
  Downloading python_dotenv-0.21.1-py3-none-any.whl (19 kB)
Collecting Jinja2>=3.0
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
    |████████| 134 kB 41.6 MB/s
Collecting click>=8.0
  Downloading click-8.1.8-py3-none-any.whl (98 kB)
    |████████| 98 kB 10.9 MB/s
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.3-py3-none-any.whl (233 kB)
    |████████| 233 kB 56.1 MB/s
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting importlib-metadata>=3.6.0; python_version < "3.10"
  Downloading importlib_metadata-6.7.0-py3-none-any.whl (22 kB)
Collecting botocore<1.34.0,>=1.33.13
  Downloading botocore-1.33.13-py3-none-any.whl (11.8 MB)
    |████████| 11.8 MB 36.1 MB/s eta 0:00:01
```

i-0e439b38571daa87c (medtrack-server)

PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 35°C Partly sunny Search ENG IN 14:06 04-07-2025

```
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: MarkupSafe, zipp, typing-extensions, importlib-metadata, click, Werkzeug, itsdangerous, flask, six, python-dateutil, jmespath, u
rlib3, botocore, s3transfer, boto3, python-dotenv
  WARNING: The script Flask is installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script dotenv is installed in '/usr/local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-2.1.5 Werkzeug-2.2.3 botocore-1.33.13 click-8.1.8 flask-2.2.5 importlib-metadata-6.7.0 itsdangerous-2.1.2
jmespath-1.0.1 python-dateutil-2.9.0.post0 python-dotenv-0.21.1 s3transfer-0.8.2 six-1.17.0 typing-extensions-4.7.1 urllib3-1.26.20 zipp-3.15.0
[root@ip-172-31-17-153 medtrack]# vim .env
[root@ip-172-31-17-153 medtrack]# python3 app.py
/usr/local/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer support Python 3.7 starting December 13, 2023. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.8 or later. More information can be found here: https://aws.amazon.com/blogs/develop
er/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
  * Serving Flask app 'app'
  * Debug mode: on
  * Environment: development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000!
Press CTRL+C to quit
  * Restarting with stat
/usr/local/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer support Python 3.7 starting December 13, 2023. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.8 or later. More information can be found here: https://aws.amazon.com/blogs/develop
er/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
  * Debugger PIN: 997-928-107
```

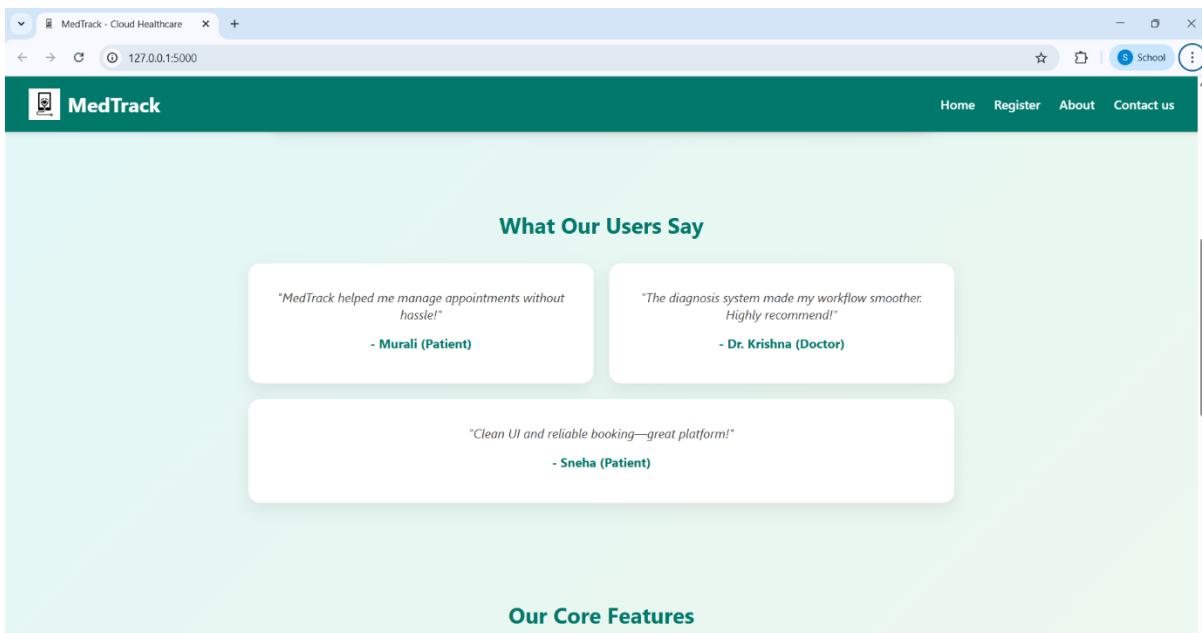
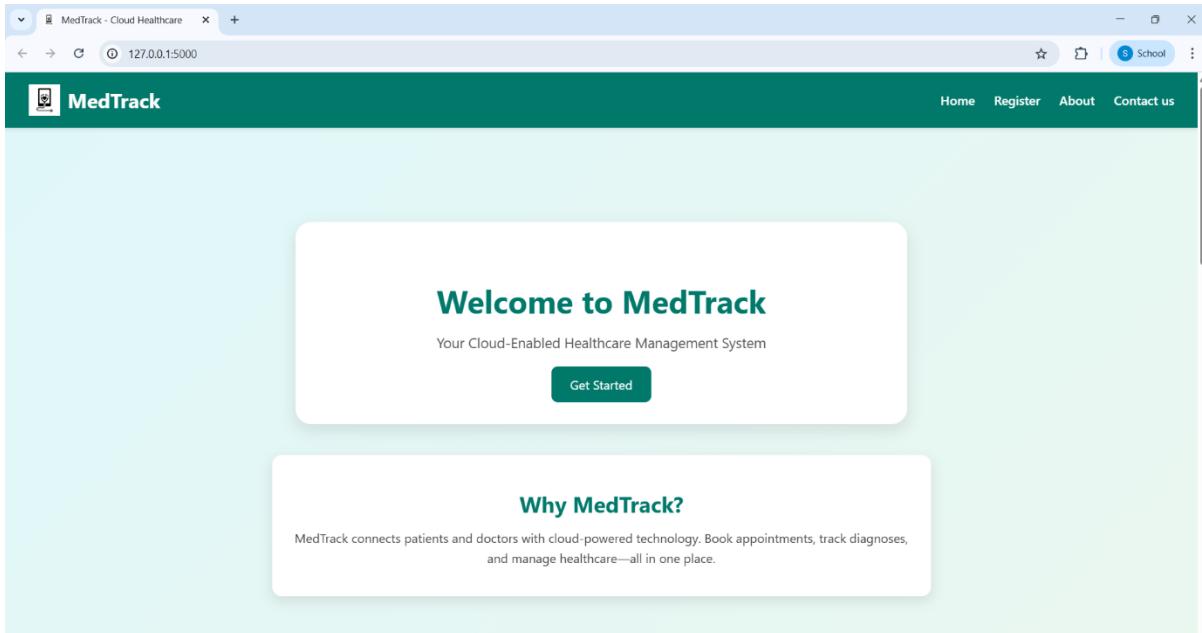
i-0e439b38571daa87c (medtrack-server)

PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 35°C Partly sunny Search ENG IN 14:07 04-07-2025

Screenshots

1. Homepage



MedTrack - Cloud Healthcare

127.0.0.1:5000

Our Core Features

 **Book Appointments**
Patients can schedule consultations with doctors easily.

 **Doctor Dashboard**
Doctors can view upcoming appointments and manage patients.

 **Diagnosis Reports**
Submit and track patient diagnoses with secure records.

MedTrack - Cloud Healthcare

127.0.0.1:5000

Diagnosis Reports

Submit and track patient diagnoses with secure records.

Contact Us

If you have any questions or need help, feel free to reach out:

📞 Phone: +91 98765 43210
✉️ Email: support@medtrack.com

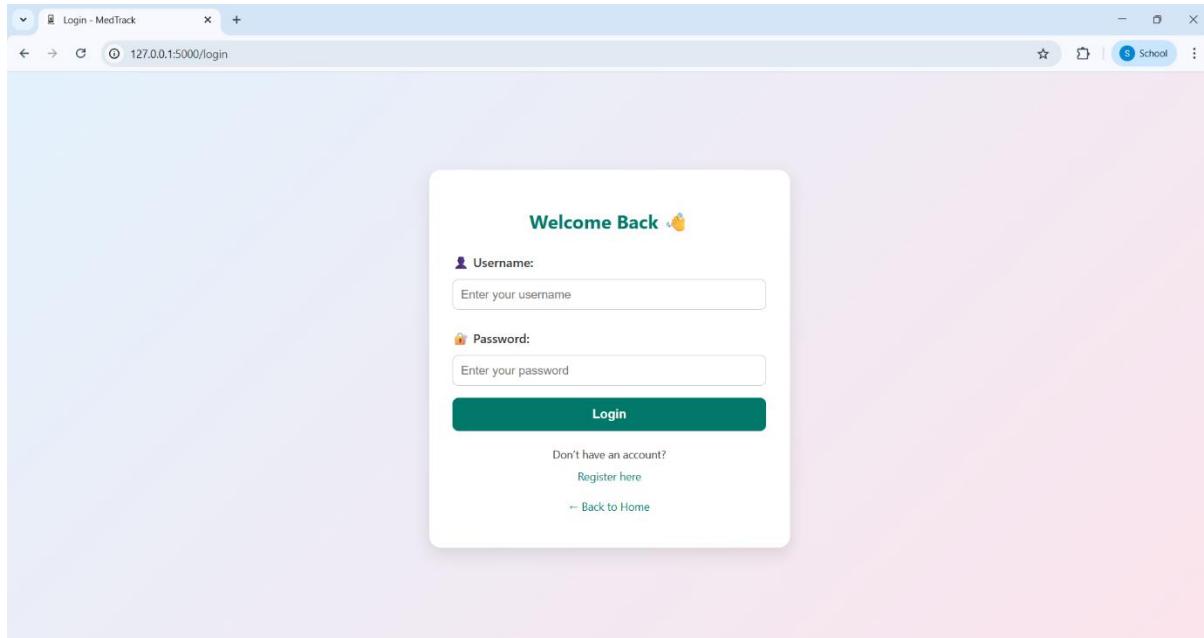
© 2025 MedTrack | Built with ❤️ for SmartBridge AWS Project

2.Register

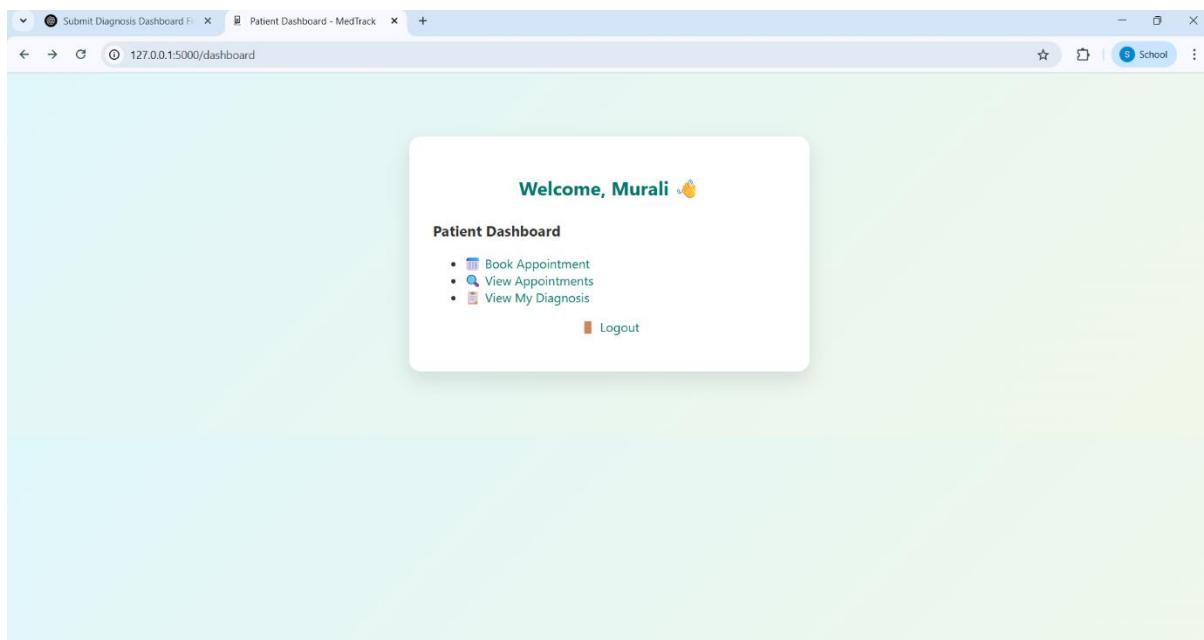
The screenshot shows a web browser window titled "Register - MedTrack" with the URL "127.0.0.1:5000/register". The main content is a white rectangular form titled "Create Your Account". It contains three input fields: "Username" (placeholder "Enter your name"), "Password" (placeholder "Create a secure password"), and "Register as:" (dropdown menu showing "-- Select Role --"). Below the form is a teal button labeled "Register". At the bottom, there are links for "Already have an account? Login here" and "← Back to Home".

This screenshot is identical to the one above, but the "Register as:" dropdown menu has been interacted with. The option "Patient" is now highlighted with a blue selection bar, while the other options "Select Role" and "Doctor" are visible below it.

3.Login



4.Patient Dashboard



5. Appointment Booking Form

A screenshot of a web browser window titled "Book Appointment - MedTrack". The URL in the address bar is "127.0.0.1:5000/book". The page displays a "Book an Appointment" form. It includes fields for "Doctor's Username" (with a placeholder input field), "Date" (a date picker with placeholder "dd-mm-yyyy"), and "Time" (a time picker with placeholder "-- : --"). A large green "Book Now" button is centered below these fields. At the bottom left of the form is a link "← Back to Dashboard".

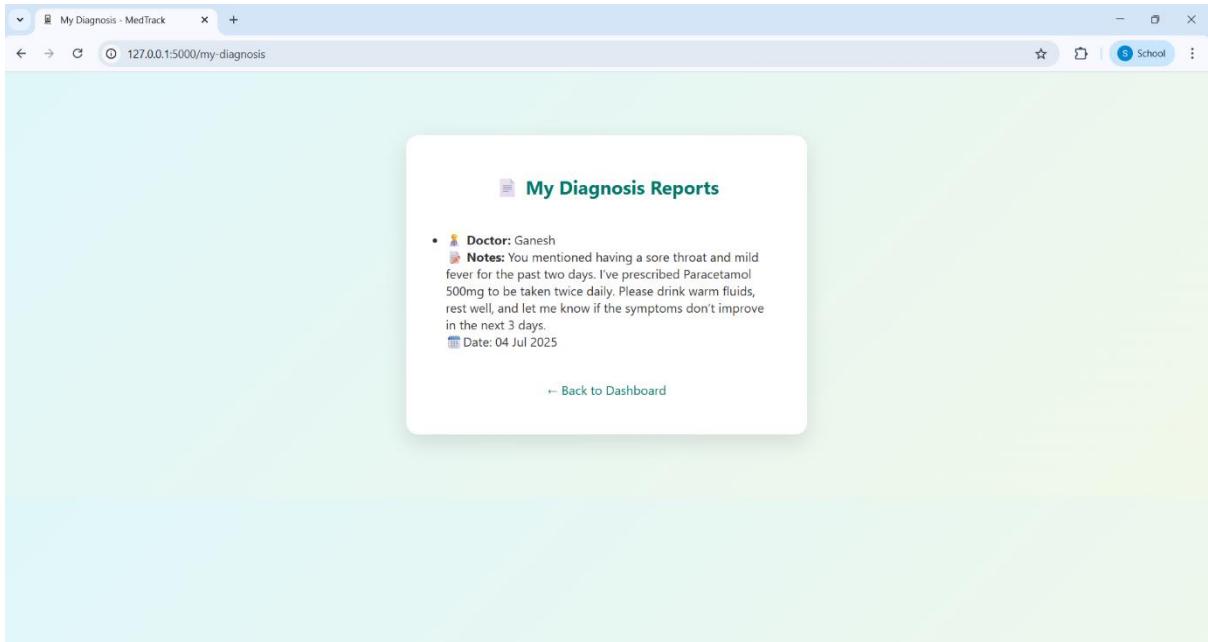
6. View Appointments

A screenshot of a web browser window titled "My Appointments - MedTrack". The URL in the address bar is "127.0.0.1:5000/appointments". The page displays a "My Appointments" section. It lists four appointment entries, each with a doctor icon, name, date, and time. The entries are:

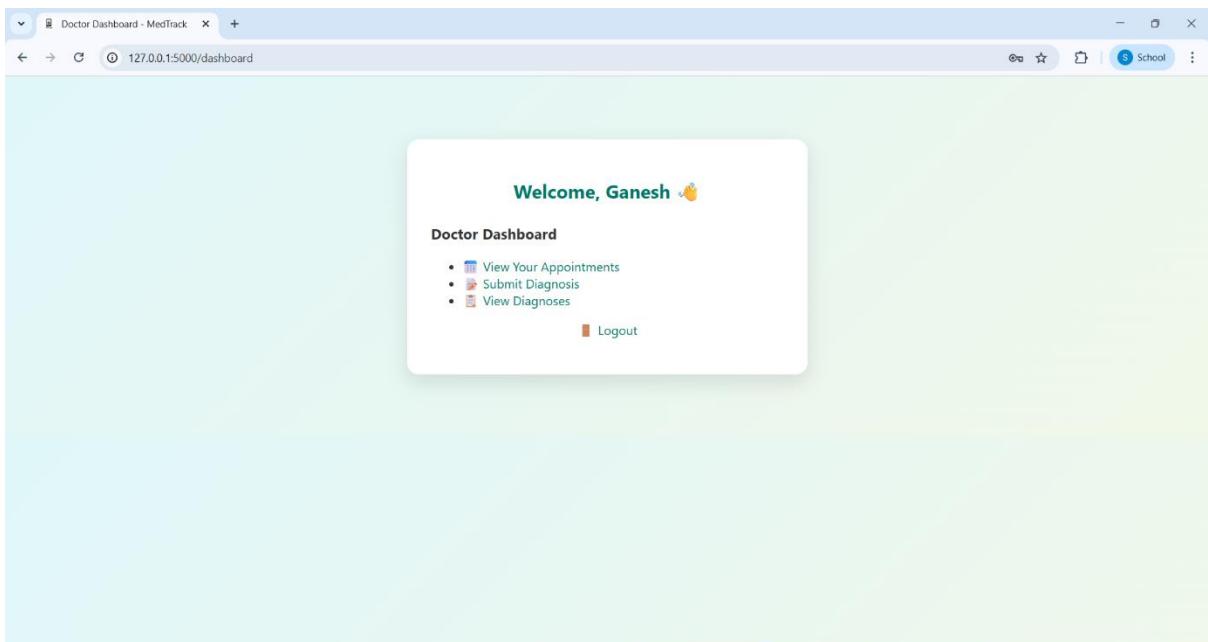
- Doctor: Krishna
📅 Date: 2026-05-12
🕒 Time: 10:00
- Doctor: Krishna
📅 Date: 2026-05-20
🕒 Time: 10:00
- Doctor: Krishna
📅 Date: 2025-07-29
🕒 Time: 12:30
- Doctor: Ganesh
📅 Date: 2025-07-04
🕒 Time: 11:30

At the bottom left of the list is a link "← Back to Dashboard".

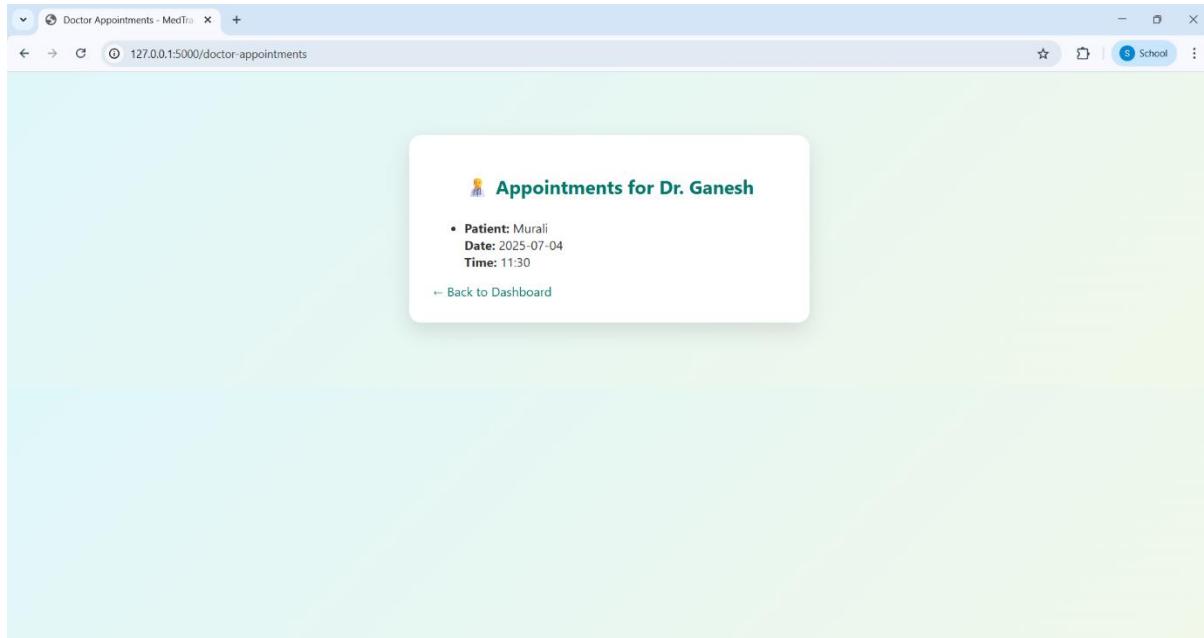
7. View My Diagnosis



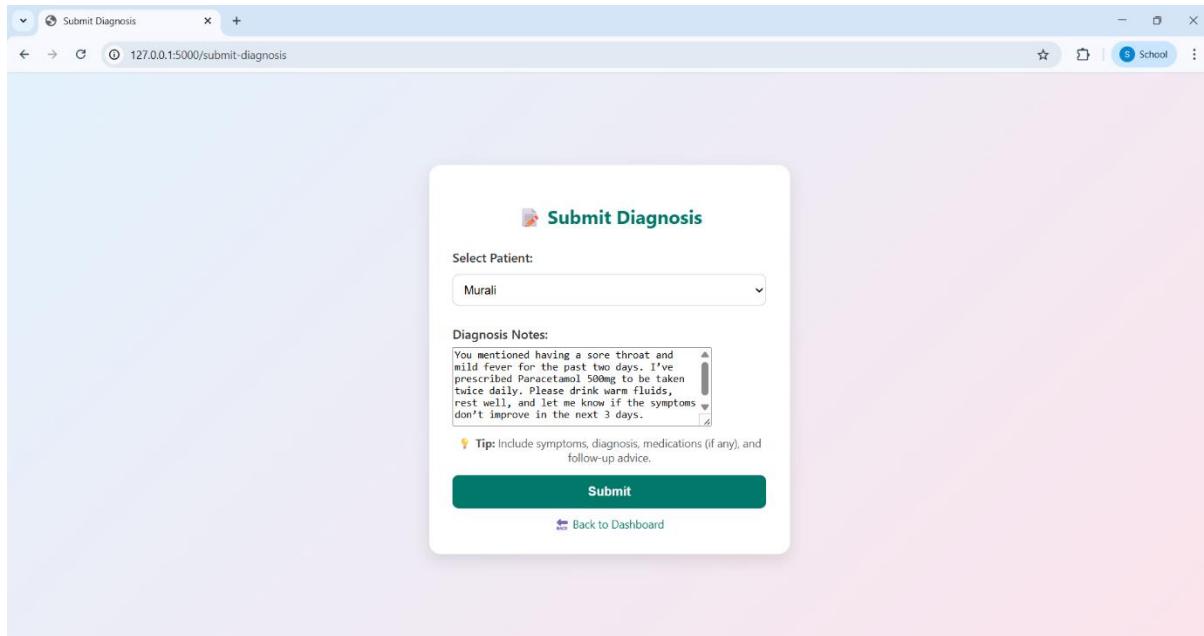
8. Doctor Dashboard

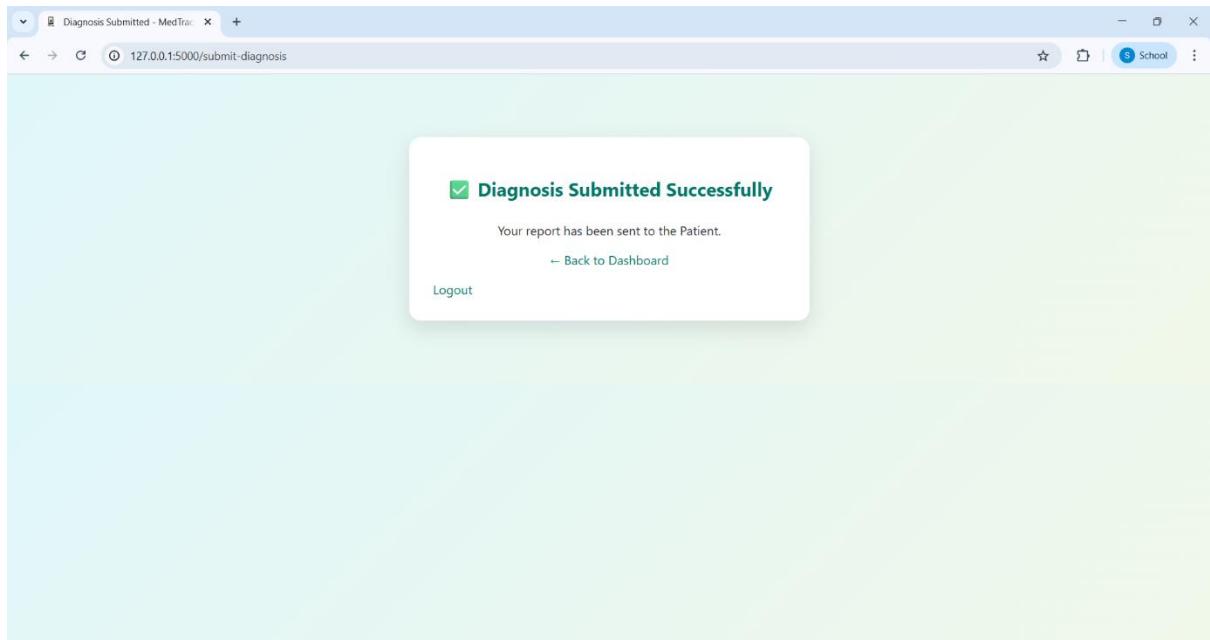


9. View Your Appointments

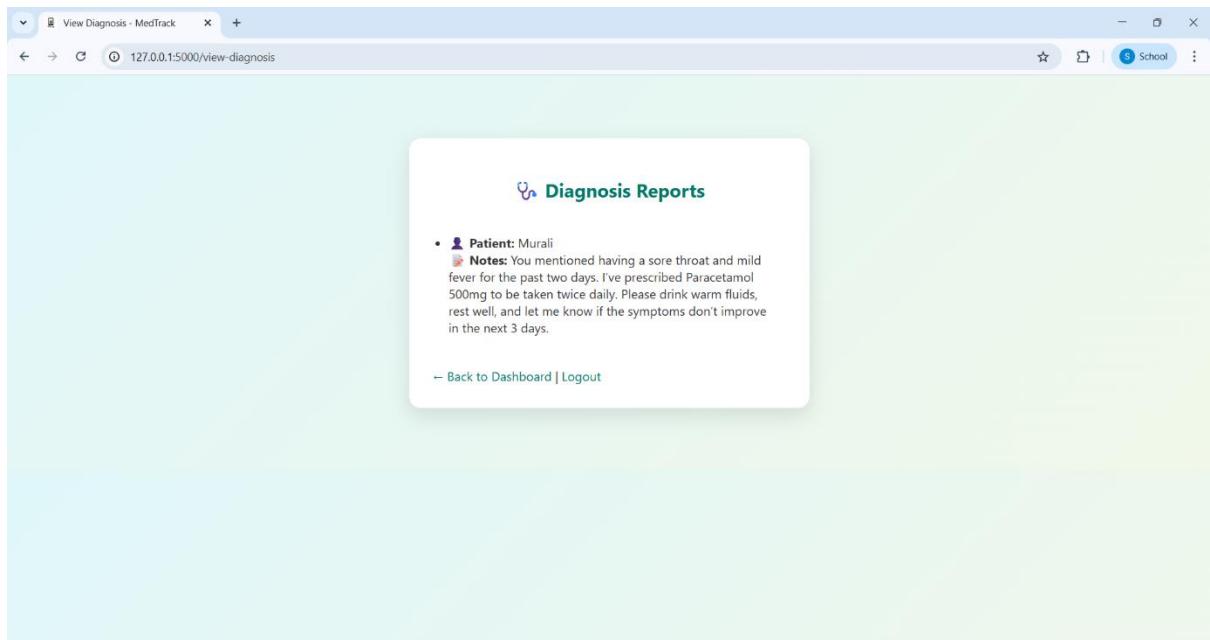


10. Submit Diagnosis





11. View Diagnosis



Demo Video

👉 <https://drive.google.com/file/d/1tyDcfKP3ZWG6uDnCSghopN0xWIw0KHIM/view?usp=drivesdk>

🔗 GitHub Repository

<https://github.com/Gopi284/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: MULA GOPI

Roll No: 228X1A4284

Branch : AI&ML

College Name : KHIT

SmartBridge AWS Internship Project - 2025