



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.

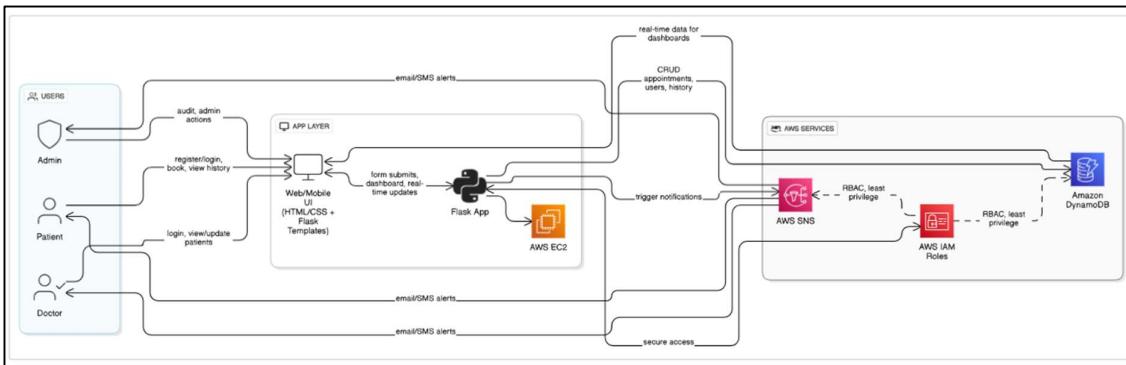
✨ Key Features

- 🔒 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

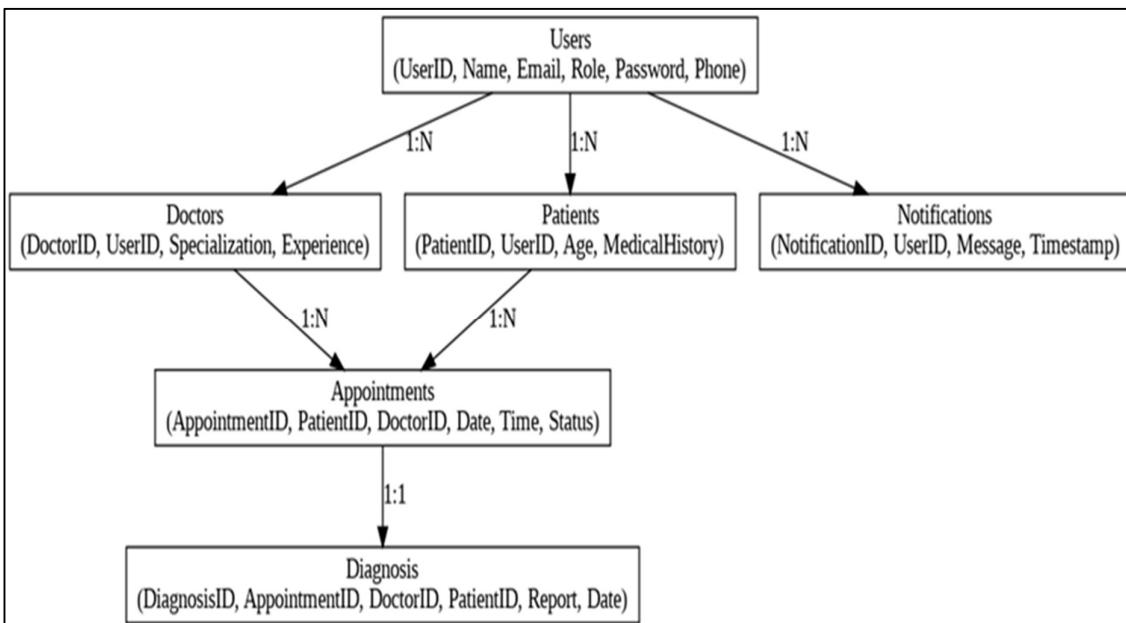
⚙️ Tech Stack

- 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:

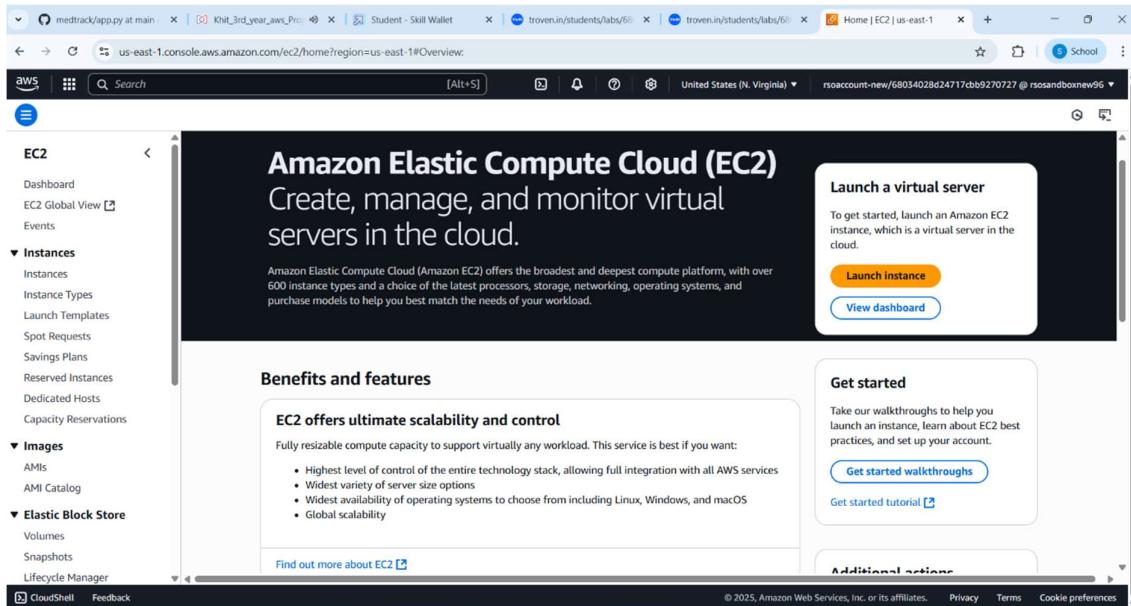
```
pip install flask boto3 python-dotenv
```

2. Run the Flask server:

```
python app.py
```

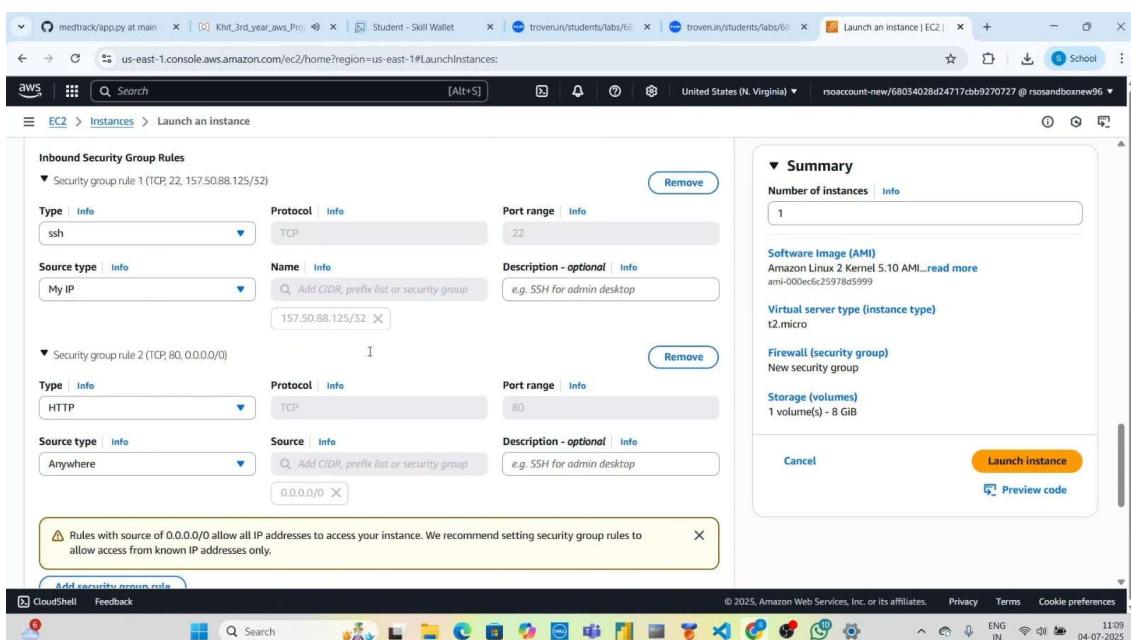
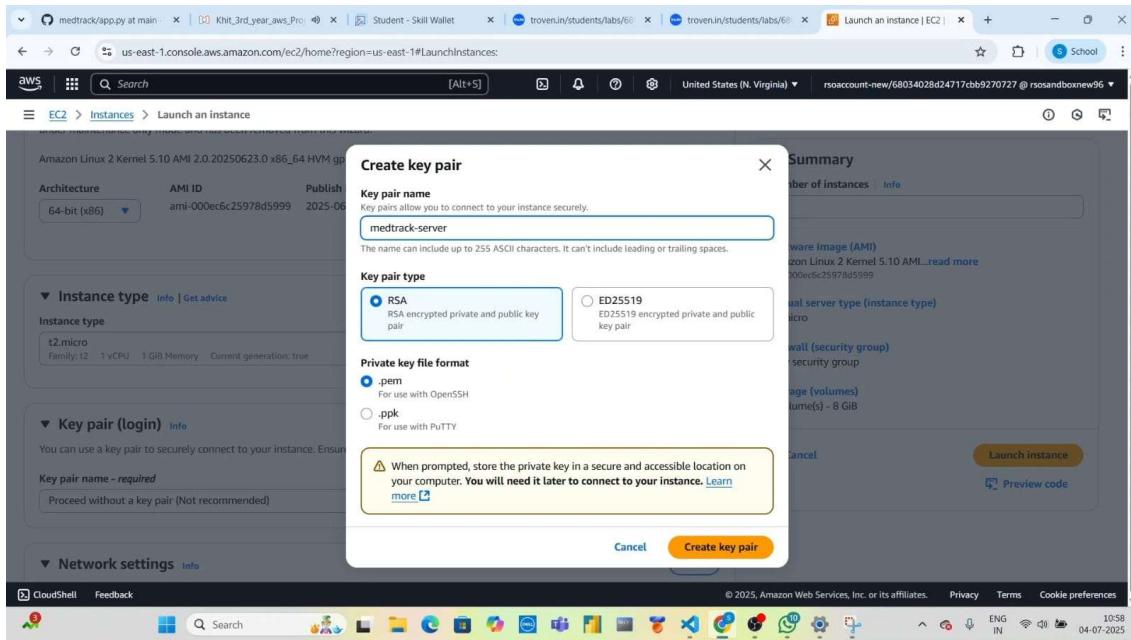
Deployment Steps

1. Launch EC2 Instance



The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. The first step, 'Name and tags', is completed with the name 'medtrack-server'. The second step, 'Application and OS Images (Amazon Machine Image)', is currently selected. It displays a search bar and a grid of AMI icons for various operating systems like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. The right panel shows a summary of the configuration: 1 instance, Amazon Linux 2 Kernel 5.10 AMI, t2.micro instance type, New security group, and 1 volume(s) - 8 GiB storage. A large orange 'Launch instance' button is prominent.

The screenshot shows the 'Amazon machine image (AMI)' step of the wizard. It details the selected AMI: 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' (ami-000ec6c25978d5999). The description notes it's tuned for optimal performance on Amazon EC2. The configuration includes: Architecture (64-bit x86), AMI ID (ami-000ec6c25978d5999), Publish Date (2025-06-20), Username (ec2-user), and a 'Verified provider' badge. The 'Instance type' section shows a selection of 't2.micro' (Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true). The 'Key pair (login)' section is partially visible at the bottom. The right panel remains the same as the previous screenshot, showing the summary and launch button.



The screenshot shows the AWS EC2 Instances Launch an instance page. A green success banner at the top states "Successfully initiated launch of instance (i-0e439b38571daa87c)". Below the banner, there's a "Launch log" section. Under "Next Steps", there are several cards: "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 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, a sidebar lists "Identity and Access Management (IAM)" and "Access management" (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management). The main area has a "New access analyzers available" banner. The "IAM resources" section shows an "Access denied" error for the user "arnaws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717cb9270727". The "AWS Account" section shows another "Access denied" error for the user "arnaws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717cb9270727". The bottom of the screen shows the Windows taskbar.

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and sections for 'Access management' (User groups, Users, Roles), 'Access reports' (Access Analyzer, Resource analysis, Unused access, Analyzer settings), and 'Credential report'. The main area displays a table of roles:

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

At the bottom right, there are links for 'Create role', 'CloudShell', 'Feedback', and the AWS logo.

The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. On the left, a sidebar lists steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main area has a title 'Name, review, and create' and a 'Role details' section:

Role name: EC2_MedTrack_Role

Description: Allows EC2 instances to call AWS services on your behalf.

Below this is the 'Step 1: Select trusted entities' section, which contains a 'Trust policy' editor with the following JSON code:

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

At the bottom right of the editor is an 'Edit' button.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The current step is 'Step 2: Add permissions'. It displays a 'Permissions policy summary' table:

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

Below the table, there's a section titled 'Step 3: Add tags' with a 'Add tags - optional' button. At the bottom right are 'Cancel', 'Previous', and 'Create role' buttons.

The screenshot shows the 'Modify IAM role' dialog in the AWS EC2 console. The instance ID is listed as i-0ffbd525dddb91fb (medtrack-server). The 'IAM role' dropdown contains the option EC2_MedTrack_Role. At the bottom right are 'Cancel' and 'Update IAM role' buttons.

The screenshot shows the AWS EC2 Instances page. A green success message at the top states "Successfully attached EC2_MedTrack_Role to instance i-Offbd525dddb91fbb". Below this, the "Instances (1/3) info" section displays 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-057d8882aa549b8c	Running	t2.micro	us-east-1c	ec2-54-8-
medtrack-server	i-Offbd525dddb91fbb	Running	t2.micro	us-east-1c	ec2-3-90-

The instance details for "i-Offbd525dddb91fbb (medtrack-server)" are shown in the "Details" tab. Key information includes:

- Instance ID: i-Offbd525dddb91fbb
- Public IPv4 address: 3.90.103.136 (open address)
- Private IPv4 addresses: 172.31.28.184
- Public DNS: ec2-3-90-103-136.compute-1.amazonaws.com (open address)
- Instance state: Running

At the bottom, there are links for CloudShell and Feedback, along with the standard Windows taskbar.

3. Setup DynamoDB Tables: Users, Appointments, Diagnoses

The screenshot shows the AWS DynamoDB homepage. A blue banner at the top encourages users to "Share your feedback on Amazon DynamoDB" and provides a "Share feedback" button.

The main content area features the following sections:

- Amazon DynamoDB**: Described as "A fast and flexible NoSQL database service for any scale".

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.
- Get started**: A call-to-action button to "Create table".
- How it works**: A section with a video thumbnail titled "What is Amazon DynamoDB? | Amazon Web Ser...".
- Pricing**: Information about charges for reading, writing, and storing data in DynamoDB tables.

On the left sidebar, there are navigation links for DynamoDB (Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, Settings) and DAX (Clusters, Subnet groups, Parameter groups, Events). At the bottom, there are CloudShell and Feedback links, along with the standard Windows taskbar.

The screenshot shows the AWS DynamoDB console in the 'Tables' section. On the left, there's a sidebar with 'DynamoDB' navigation items like 'Dashboard', 'Tables', 'Explore items', 'PartQL editor', 'Backups', 'Exports to S3', 'Imports from S3', 'Integrations', 'Reserved capacity', and 'Settings'. Below that is a 'DAX' section with 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main area has a header with 'Share your feedback on Amazon DynamoDB' and a 'Create table' button. It includes filters for 'Name', 'Status', 'Partition key', 'Sort key', 'Indexes', 'Replication Regions', 'Deletion protection', 'Favorite', 'Read capacity mode', and 'Write capacity mode'. A search bar says 'Find tables' and dropdowns for 'Any tag key' and 'Any tag value'. At the bottom, it says 'Loading tables'.

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The first step, 'Table name', has a field for 'Users' with placeholder text 'Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).'. The second step, 'Partition key', has a field for 'username' with type 'String'. The third step, 'Sort key - optional', has a field for 'Enter the sort key name' with type 'String'. The fourth step, 'Table settings', contains two sections: 'Default settings' (selected) with a note about modifying settings after creation, and 'Customize settings' with a note about using advanced features. At the bottom, there's a note about default table settings and a footer with standard AWS links and system status.

Screenshot of the AWS DynamoDB 'Create table' wizard. The configuration settings are as follows:

Maximum write capacity units	-	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	AWS owned key	Yes
Deletion protection	Off	Yes
Resource-based policy	Not active	Yes

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](#)

Screenshot of the AWS CloudShell interface showing the AWS logo and various icons.

Screenshot of the AWS DynamoDB 'Create table' wizard. A blue banner at the top encourages sharing feedback: "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." It includes a "Share feedback" button and a notifications counter (0 notifications).

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.

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.

1

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.

Screenshot of the AWS CloudShell interface showing the AWS logo and various icons.

Screenshot of the AWS DynamoDB 'Create table' wizard:

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.

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.

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.

Share feedback

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

Create table

Screenshot of the AWS DynamoDB 'Tables' page:

Dashboard

Tables

- Explore items
- PartiQL editor
- Backups
- Exports to S3
- Imports from S3
- Integrations
- 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.

Actions Delete Create table

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read c
Appointments	Active	appointment_id (\$)	-	0 0	Off	☆	On-demand	
Diagnoses	Active	diagnosis_id (\$)	-	0 0	Off	☆	On-demand	
Users	Active	username (\$)	-	0 0	Off	☆	On-demand	

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 Amazon Simple Notification Service (SNS) homepage. At the top, there is a banner announcing the support for High Throughput FIFO topics. Below the banner, the title "Amazon Simple Notification Service" is displayed, followed by a subtitle "Pub/sub messaging for microservices and serverless applications." A brief description of Amazon SNS is provided, mentioning its availability, durability, security, and managed nature. To the right, there is a "Create topic" form with a "Topic name" input field containing "MyTopic" and a "Next step" button. Below this, a "Pricing" section is shown, stating that there are no upfront costs and providing information about pay-as-you-go pricing based on message volume. At the bottom left, a "Benefits and features" section is visible. The browser's address bar shows the URL "us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/homepage". The status bar at the bottom indicates the date as "04-07-2025" and the time as "12:11".

The screenshot shows the "Topics" page within the Amazon SNS service. On the left, a sidebar menu includes "Dashboard", "Topics", "Subscriptions", and "Mobile" sections, with "Push notifications" and "Text messaging (SMS)" listed under "Mobile". The main content area is titled "Topics (0)". It features a search bar and a table with columns for "Name", "Type", and "ARN". A message at the bottom encourages users to "To get started, create a topic." and provides a "Create topic" button. The browser's address bar shows the URL "us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/topics". The status bar at the bottom indicates the date as "04-07-2025" and the time as "12:11".

Screenshot of the AWS SNS 'Create topic' page.

New Feature: Amazon SNS now supports High Throughput FIFO topics. Learn more [\[Info\]](#)

Error code: AccessDeniedException - Error message: User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d24717ccb9270727 is not authorized to perform kms:DescribeKey on resource: arn:aws:kms:us-east-1:600627341644:key/c06920cc-9330-4a08-b273-94666808c88 because no identity-based policy allows the kms:DescribeKey action

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Standard

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Name

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional [Info](#)
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:14 04-07-2025

Screenshot of the AWS SNS 'Topics' page showing the newly created 'Medtrack' topic.

New Feature: Amazon SNS now supports High Throughput FIFO topics. Learn more [\[Info\]](#)

Topic Medtrack created successfully.
You can create subscriptions and send messages to them from this topic.

Medtrack

Details

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

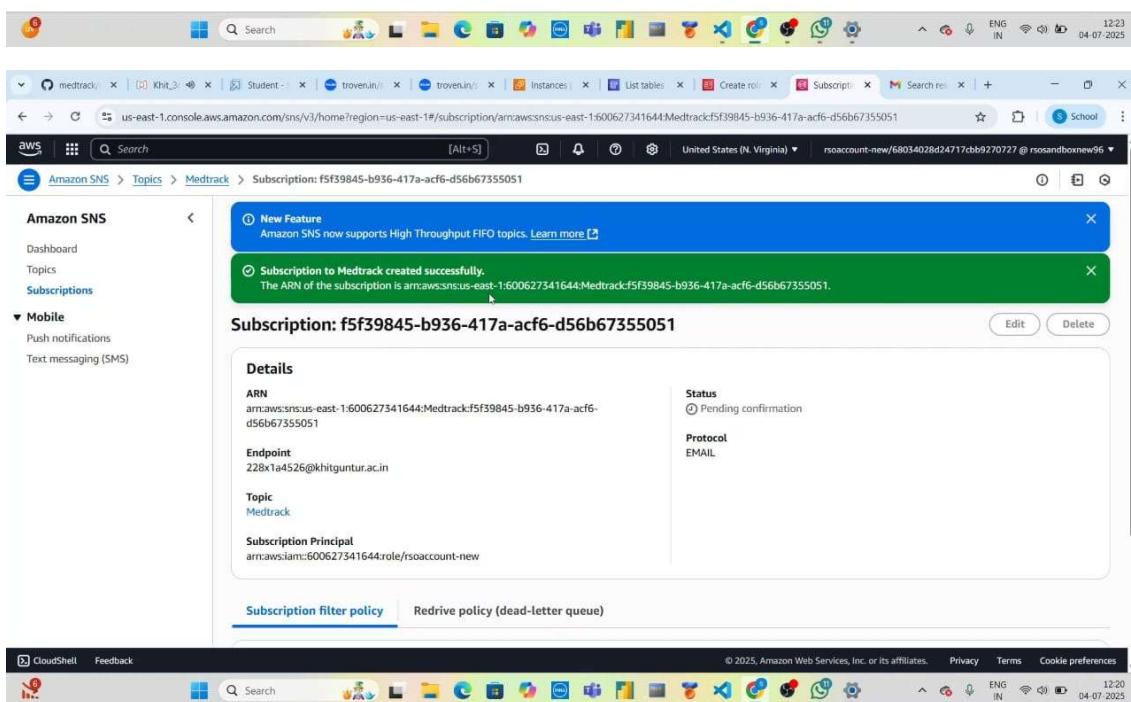
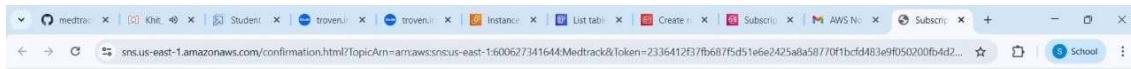
[Edit](#) [Delete](#) [Publish message](#)

[CloudShell](#) [Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 12:14 04-07-2025

The screenshot shows the AWS SNS console with the 'Medtrack' topic selected. The left sidebar shows navigation options like 'Amazon SNS', 'Dashboard', 'Topics', 'Subscriptions', and 'Mobile'. The main panel displays the 'Medtrack' topic details, including its name, ARN, and type. A 'Subscriptions' tab is active, showing a table with no subscriptions found. Buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and 'Create subscription' are available.

The screenshot shows the 'Create subscription' dialog for the 'Medtrack' topic. It requires entering the 'Topic ARN' (arn:aws:sns:us-east-1:600627341644:Medtrack), choosing the 'Protocol' (Email), and providing the 'Endpoint' (an email address: 228x1a4526@khitguntur.ac.in). A note indicates that confirmation is required after creation. The 'Create subscription' button is highlighted in orange.



The screenshot shows the AWS SNS console with the URL <https://us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/subscription/arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051>. The page displays a subscription for the topic 'Medtrack' with the ARN `arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051`. The subscription status is confirmed via email to the endpoint `2281a4526@khitguntur.ac.in`.

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 with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance-instanceId=i-0e439b38571daa87c>. The page lists four instances under the 'Instances' section, with the first one, 'medtrack-server' (ID: i-0e439b38571daa87c), being running. The instance details show it's a t2.micro type, running, with 2/2 checks passed, and assigned a public IP of ec2-13-23-11-199. The security tab indicates it has an IAM role named 'EC2_MedTrack_Role' and is part of a security group 'sg-0475e8a791fdaaa'.

The screenshot shows the AWS EC2 Connect interface. At the top, there's a navigation bar with tabs like 'EC2', 'Instances', and 'Connect to instance'. Below it, a sub-menu for 'Connect info' is open. It displays two main options: 'Connect using a Public IP' (selected) and 'Connect using a Private IP'. Under 'Public IP', the IP address '13.222.1.50' is listed. A 'Username' field contains 'ec2-user'. A note at the bottom states: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' On the right side of the interface are 'Cancel' and 'Connect' buttons.

The screenshot shows a terminal session in the AWS CloudShell. The session title is 'i-0e439b38571daa87c (medtrack-server)'. The terminal output shows the following text:

```
'`#` 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 ~]$'
```

Below the terminal, the AWS CloudShell interface includes a 'CloudShell' tab, a feedback link, and standard browser navigation controls. The status bar at the bottom indicates the session is running on 'BAJFINANCE +1.23%' and shows the date and time as '04-07-2025'.

```
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 ec2-user]# cd --
[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)

Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



```
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
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)

Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



```
separate git dir from working tree
--[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-]parse
    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)

Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



```
separate git dir from working tree
--[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-]parse
    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)

Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



```
[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 boto3-1.33.13-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>=2.1.0
  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 kB)
    ██████████ | 11.8 kB 36.1 MB/s eta 0:00:01
```

i-0e439b38571daa87c (medtrack-server)

PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153



```
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: MarkupSafe, Jinja2, zipp, typing-extensions, importlib-metadata, click, Werkzeug, itsdangerous, flask, six, python-dateutil, jmespath, urllib3, botocore, s3transfer, toolz, python-dotenv
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.
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 Jinja2-3.1.6 MarkupSafe-2.1.5 Werkzeug-2.2.3 boto3-1.33.13 botocore-1.33.13 click-8.1.8 flask-2.2.5 importlib-metadata-6.7.0 itsdangerous-2.1.2 jmespath-1.0.0 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/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
  warnings.warn(warning, PythonDeprecationWarning)
  * Serving static app 'app'
  * Debugging mode on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:9000
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/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
  * Debugger is active!
  * Debugger PIN: 997-928-107
```

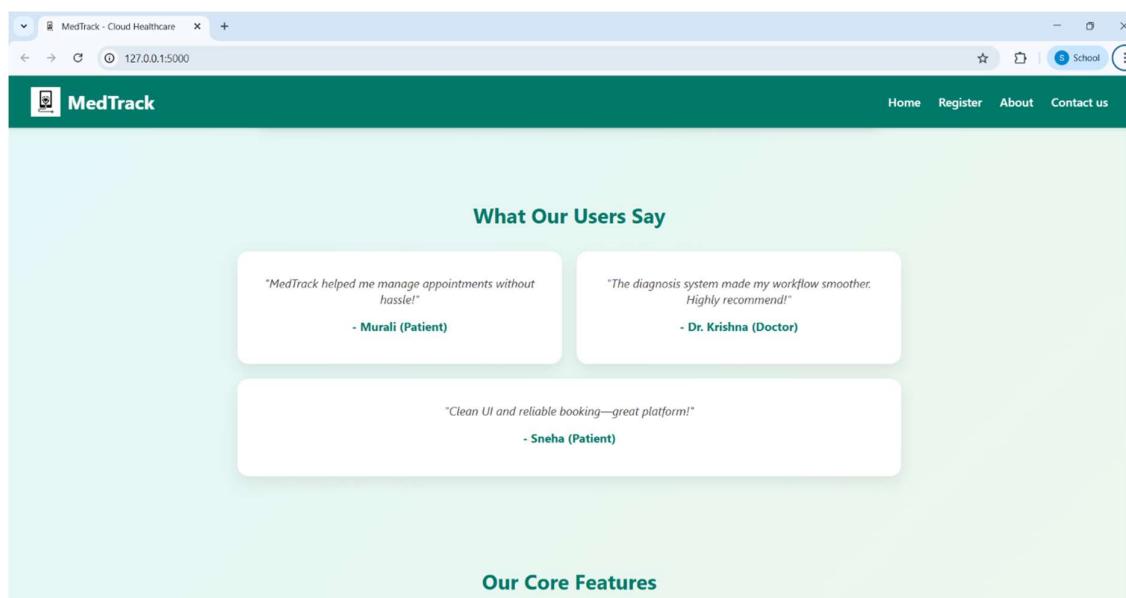
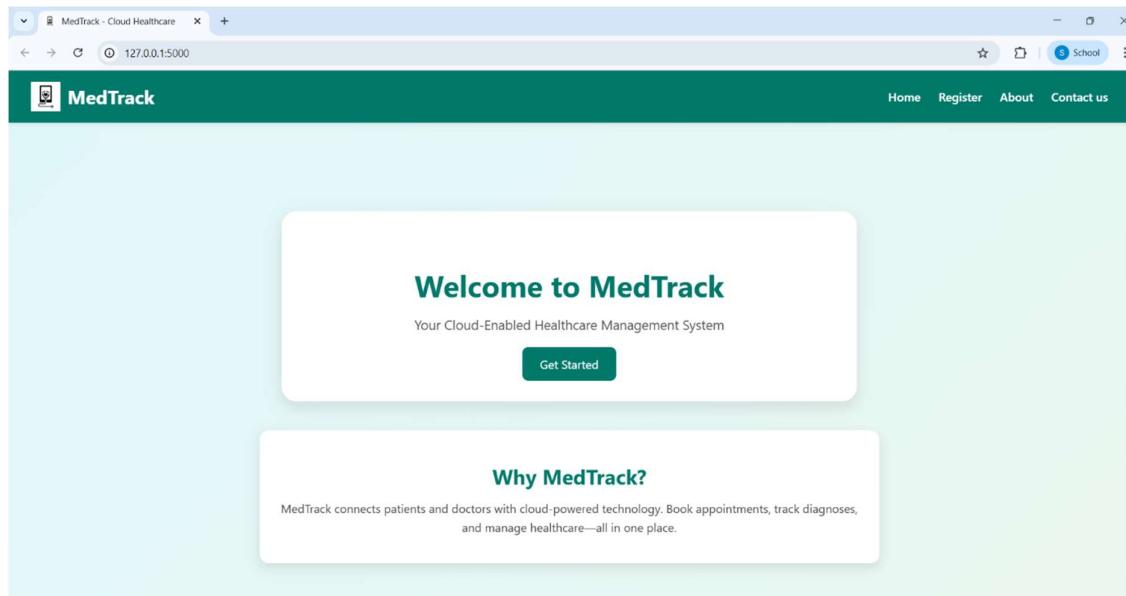
i-0e439b38571daa87c (medtrack-server)

PublicIPs: 13.222.1.50 PrivateIPs: 172.31.17.153



Screenshots

1. Homepage



MedTrack - Cloud Healthcare

127.0.0.1:5000

Home Register About Contact us

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

Home Register About Contact us

 **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

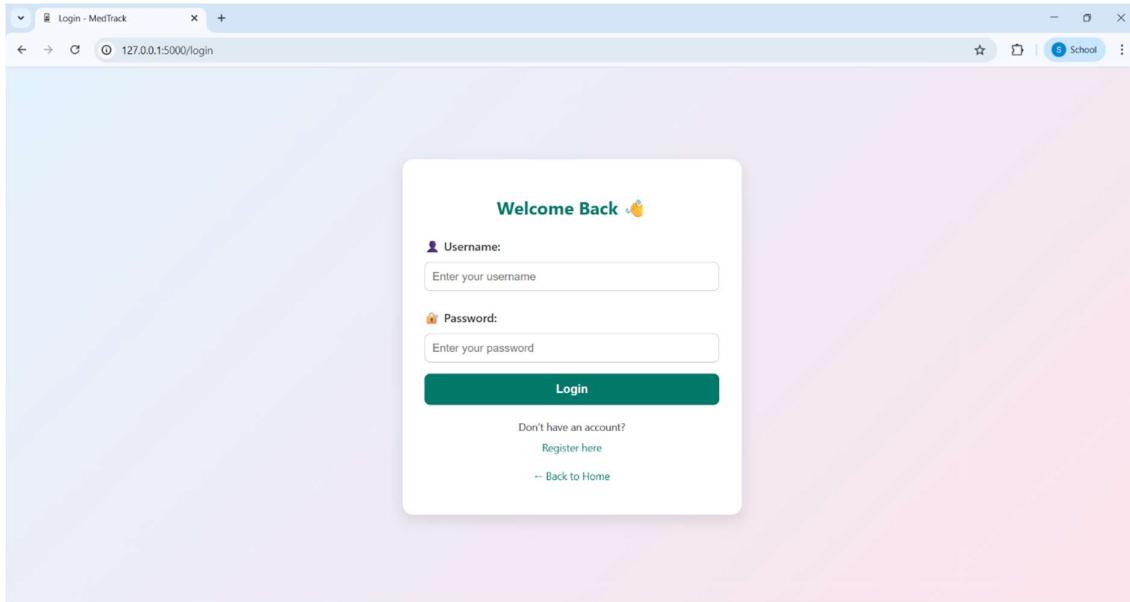
© 2025 MedTrack | Built with ❤ for SmartBridge AWS Project

2. Register

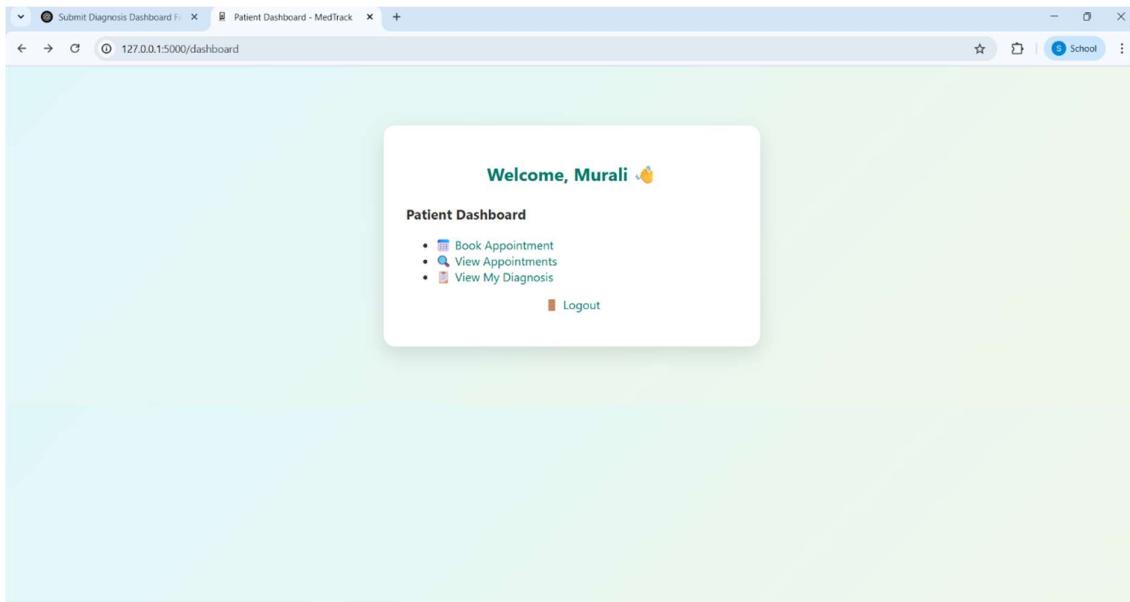
The screenshot shows a web browser window titled "Register - MedTrack" with the URL "127.0.0.1:5000/register". The main content is a "Create Your Account" form. It includes fields for "Username" (placeholder "Enter your name") and "Password" (placeholder "Create a secure password"). A dropdown menu for "Register as" is open, showing "Select Role" and two options: "Patient" and "Doctor". A large green "Register" button is at the bottom. Below the form, links for "Already have an account? Login here" and "Back to Home" are visible.

This screenshot is identical to the one above, but the "Patient" option in the "Register as" dropdown is now highlighted with a blue selection bar, indicating it has been chosen.

3.Login



4.Patient Dashboard



5. Appointment Booking Form

The screenshot shows a web browser window titled "Book Appointment - MedTrack". The URL in the address bar is "127.0.0.1:5000/book". The main content area has a light green background. A central white card contains the title "Book an Appointment" with a calendar icon. It includes three input fields: "Doctor's Username" (with a placeholder box), "Date" (with a date picker showing "dd-mm-yyyy" and a clear button), and "Time" (with a time picker showing "HH:MM"). Below these is a large green "Book Now" button. At the bottom of the card is a link "← Back to Dashboard".

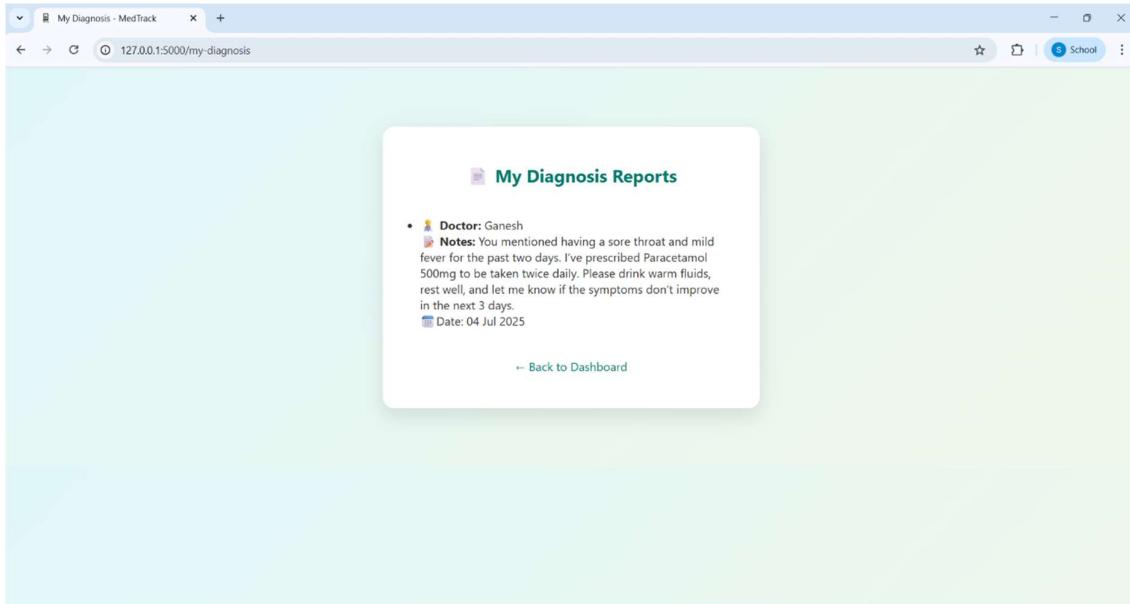
6. View Appointments

The screenshot shows a web browser window titled "My Appointments - MedTrack". The URL in the address bar is "127.0.0.1:5000/appointments". The main content area has a light green background. A central white card contains the title "My Appointments" with a calendar icon. It lists four past appointments in a bulleted format, each with a person icon, doctor name, date, and time. The entries are:

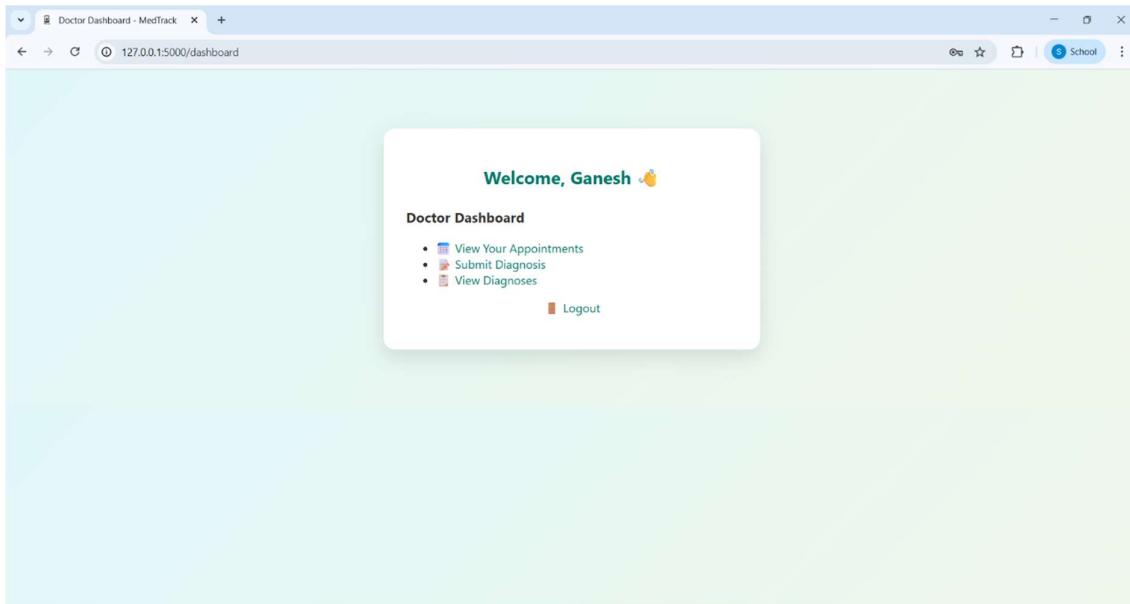
- Doctor: Krishna
Date: 2026-05-12
Time: 10:00
- Doctor: Krishna
Date: 2026-05-20
Time: 10:00
- Doctor: Krishna
Date: 2025-07-29
Time: 12:30
- Doctor: Ganesh
Date: 2025-07-04
Time: 11:30

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

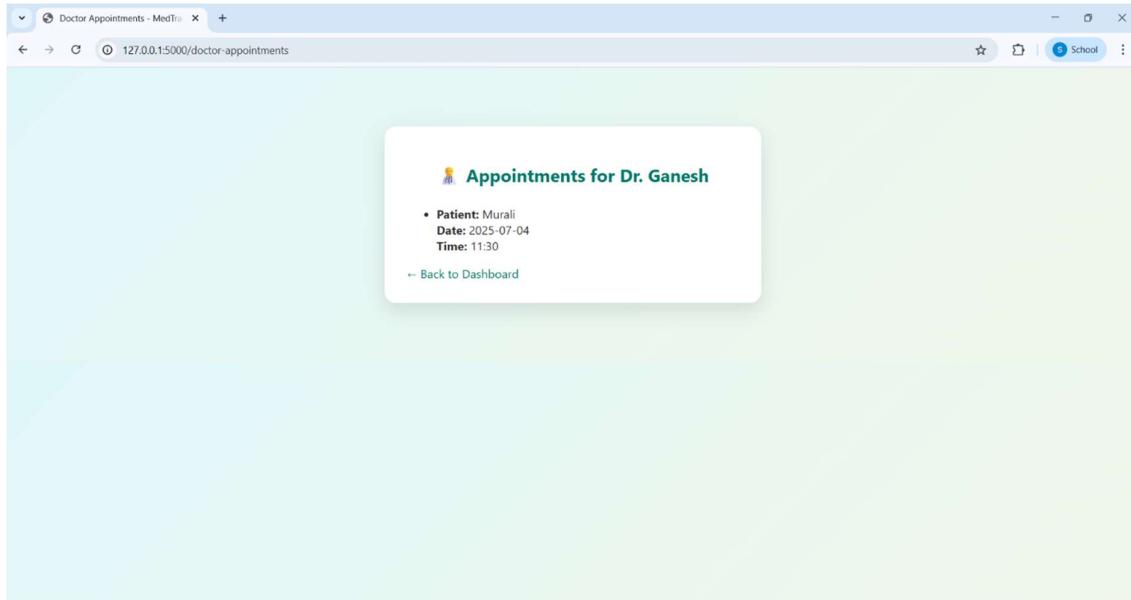
7. View My Diagnosis



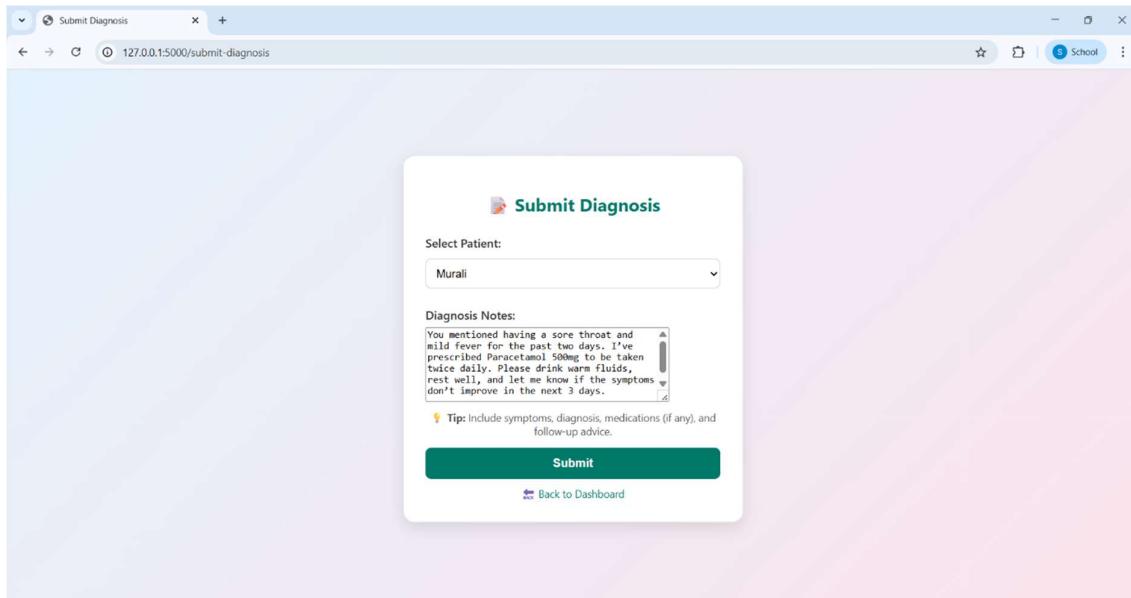
8. Doctor Dashboard

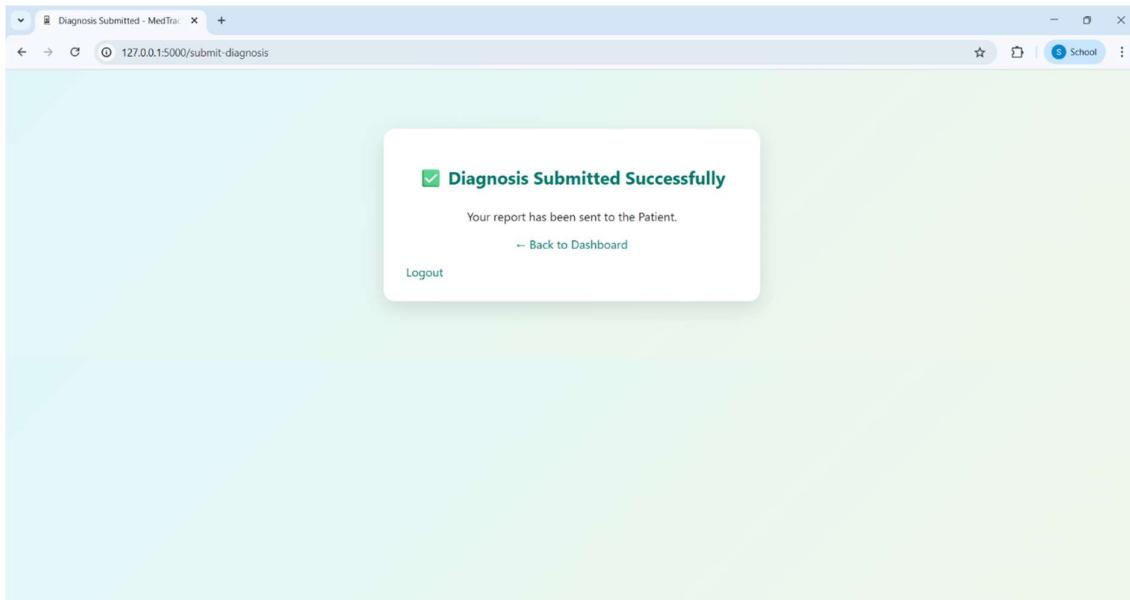


9. View Your Appointments

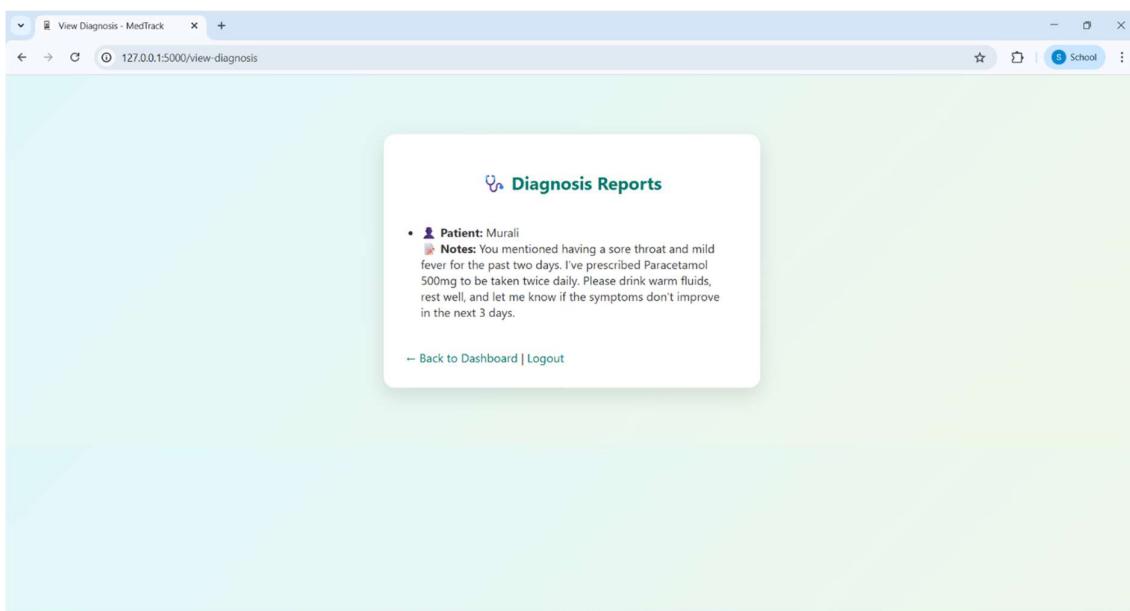


10. Submit Diagnosis





11. View Diagnosis



🔗 GitHub Repository

<https://github.com/vamsi911/medtrack-.git>

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: Kommanaboina Vamsi Krishna
SmartBridge AWS Internship Project - 2025