



# 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

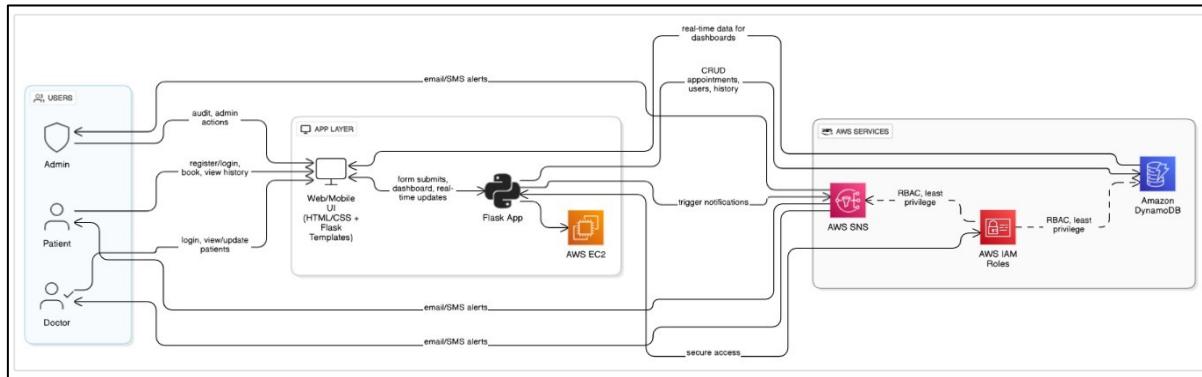


## Tech Stack

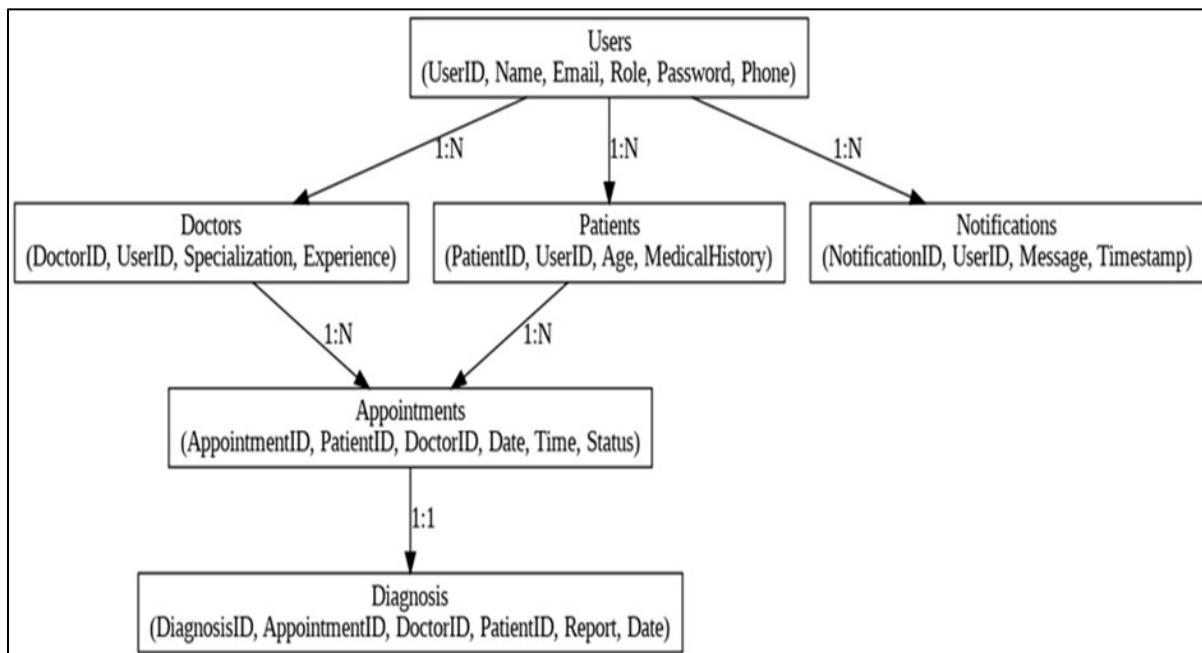
- Frontend: HTML, CSS
- Backend: Python (Flask)

- 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, Global View, Events), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area has a dark header "Amazon Elastic Compute Cloud (EC2)" with the subtext "Create, manage, and monitor virtual servers in the cloud." Below this, a paragraph describes EC2's offerings and benefits. To the right, a large call-to-action button says "Launch a virtual server" with "Launch instance" and "View dashboard" options. A "Get started" section with "Get started walkthroughs" and "Get started tutorial" links is also present.

The screenshot shows the "Launch an instance" wizard. Step 1: Name and tags. It asks for a name ("medtrack-server") and provides an "Add additional tags" option. Step 2: Application and OS Images (Amazon Machine Image). It lists various AMI options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. Step 3: Quick Start, showing a grid of AMI icons. Step 4: Summary, where the user can review the instance details (1 instance, AMI selected, t2.micro instance type, New security group, 1 volume(8 GiB)). Buttons for "Cancel", "Launch instance", and "Preview code" are at the bottom.

Screenshot of the AWS CloudShell interface showing the launch of an EC2 instance.

**Description**  
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

**Architecture**: 64-bit (x86)    **AMI ID**: ami-000ec6c25978d5999    **Publish Date**: 2025-06-20    **Username**: ec2-user    **Verified provider**

**Instance type**: t2.micro (1 vCPU, 1 GiB Memory, Current generation: true)

**Key pair (login)**: medtrack-server

**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

**Launch instance** | **Preview code**

Screenshot of the AWS CloudShell interface showing the creation of a key pair for the EC2 instance.

**Create key pair**

**Key pair name**: medtrack-server

**Key pair type**: RSA (selected)

**Private key file format**: .pem (selected)

**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](#)

**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

**Launch instance** | **Preview code**

The screenshot shows the AWS EC2 'Launch an instance' wizard. On the left, under 'Inbound Security Group Rules', two rules are listed:

- Security group rule 1 (TCP, 22, 157.50.88.125/32)**: Type: ssh, Protocol: TCP, Port range: 22, Source type: My IP, Name: (empty), Description: e.g. SSH for admin desktop, Source: 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: (empty), Description: e.g. SSH for admin desktop, Source: 0.0.0.0/0.

A warning message at the bottom left states: "⚠️ 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." On the right, the 'Summary' section shows:

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

Buttons include 'Cancel', 'Launch instance' (highlighted in yellow), and 'Preview code'.

The screenshot shows the AWS EC2 instance launch success page. A green success banner at the top says: "Success Successfully initiated launch of instance (i-0e439b38571daa87c)". Below it, a 'Next Steps' section lists several options:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Button: Create billing alerts.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Buttons: Connect to instance, Learn more.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Buttons: Connect an RDS database, Create a new RDS database, Learn more.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Button: Create EBS snapshot policy.

At the bottom, there are links for 'Manage detailed monitoring', 'Create Load Balancer', 'Create AWS budget', and 'Manage CloudWatch alarms'.

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

Screenshot of the AWS IAM Dashboard showing access denied errors and account details.

**IAM Dashboard**

**IAM resources**

**Access denied**

You don't have permission to `iam:GetAccountSummary`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d2  
Action: iam:GetAccountSummary  
Context: no identity-based policy allows the action

[Diagnose with Amazon Q](#)

**AWS Account**

**Access denied**

You don't have permission to `iam>ListAccountAliases`. To request access, copy the following text and send it to your AWS administrator. [Learn more about troubleshooting access denied errors.](#)

User: arn:aws:sts::600627341644:assumed-role/rsoaccount-new/68034028d2  
Action: iam>ListAccountAliases  
Context: no identity-based policy allows the action

[Diagnose with Amazon Q](#)

**What's new**

Updates for features in IAM

**Tools**

Policy simulator

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS Roles page showing a list of existing roles.

**Roles (12)**

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

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

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM 'Create role' wizard - Step 1: Name, review, and create.

**Name, review, and create**

Step 1  
Select trusted entity  
Step 2  
Add permissions  
Step 3  
**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**EC2\_MedTrack\_Role**

Maximum 64 characters. Use alphanumeric and '+\_-.@~.' characters.

**Description**  
Add a short explanation for this role.  
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters [A-Z and a-z], numbers [0-9], tabs, new lines, or any of the following characters: +\_-.@~!#\$%^&\*\_=

**Step 1: Select trusted entities**

**Trust policy**

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

```

CloudShell Feedback Search © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:34 04-07-2025

Screenshot of the AWS IAM 'Create role' wizard - Step 2: Add permissions.

**Step 2: Add permissions**

**Permissions policy summary**

Policy name	Type	Attached as
<a href="#">AmazonDynamoDBFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonSNSFullAccess</a>	AWS managed	Permissions policy

**Step 3: Add tags**

**Add tags - optional** Info  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

**Add new tag**  
You can add up to 50 more tags.

Cancel Previous **Create role**

CloudShell Feedback Search © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 11:35 04-07-2025

The screenshot shows the 'Modify IAM role' page in the AWS IAM console. The instance ID 'i-0ffbd525dddb91fbb (medtrack-server)' is selected. In the 'IAM role' section, the dropdown menu shows 'EC2\_MedTrack\_Role'. At the bottom right, there are 'Cancel' and 'Update IAM role' buttons.

The screenshot shows the 'Instances' page in the AWS EC2 console. It displays three instances: 'medtrack-server' (running, t2.micro, us-east-1c, ec2-13-2), 'medtrack-server' (running, t2.micro, us-east-1c, ec2-54-8), and 'medtrack-server' (selected, running, t2.micro, us-east-1c, ec2-3-90). A success message at the top states 'Successfully attached EC2\_MedTrack\_Role to instance i-0ffbd525dddb91fbb'. The instance details for 'medtrack-server' (selected) are shown, including its Public IPv4 address (3.90.103.136), Private IPv4 address (172.31.28.184), and Public DNS (ec2-3-90-103-136.compute-1.amazonaws.com).

### 3. Setup DynamoDB Tables: Users, Appointments, Diagnoses

The screenshot shows the Amazon DynamoDB home page. On the left, there's a navigation sidebar with sections like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Below this is a section for DAX with Clusters, Subnet groups, Parameter groups, and Events. At the bottom of the sidebar are CloudShell and Feedback links. The main content area features a "Share your feedback on Amazon DynamoDB" banner, followed by 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. A "How it works" section with a video thumbnail is also present. The footer contains copyright information and links for Privacy, Terms, and Cookie preferences.

The screenshot shows the Amazon DynamoDB Tables page. The left sidebar is identical to the home page. The main area displays a "Tables" list with columns for 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 of the table allows filtering by Name, Tag key, and Tag value. The status bar at the bottom indicates "Loading tables". The footer is identical to the home page.

Screenshot of the AWS DynamoDB 'Create table' wizard.

The page shows the following steps:

- Table name:** 'Users'. A note says: "DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table."
- Partition key:** 'username' (String type). A note says: "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."
- Sort key - optional:** 'Enter the sort key name' (String type). A note says: "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."
- Table settings:**
  - Default settings:** Selected. A note says: "The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'."
  - Customize settings:** Unselected. A note says: "Use these advanced features to make DynamoDB work better for your needs."
- Default table settings:** Note: "These are the default settings for your new table. You can change some of these settings after creating the table."

At the bottom, there are links for CloudShell, Feedback, and various AWS services like Lambda, S3, and CloudWatch. The status bar shows the date (04-07-2025), time (11:55), and language (ENG IN).

This screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The first step, 'Set table settings', is displayed. Key configuration parameters include:

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

### 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 CloudShell interface is shown at the top of the page. A feedback link is visible in the header.

A blue banner at the top of the page encourages users to share feedback on Amazon DynamoDB. It includes a 'Share feedback' button and a survey link: [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.](#)

### 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.

diagnosis\_id

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

1 to 255 characters and case sensitive.

The CloudShell interface is shown at the top of the page. A feedback link is visible in the header.

Screenshot of the Amazon DynamoDB console showing the 'Create table' wizard.

**Table details**

**Table name:** Appointments

**Partition key:** appointment\_id (String)

**Sort key - optional:** Enter the sort key name (String)

**DynamoDB Tables**

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

## 4. Configure SNS

The screenshot shows the Amazon 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 page title is "Amazon Simple Notification Service" with the subtitle "Pub/sub messaging for microservices and serverless applications". A callout box on the right side is titled "Create topic" and contains a "Topic name" input field with "MyTopic" typed in, a "Next step" button, and a link "Start with an overview". Another callout box titled "Pricing" provides information about pay-as-you-go pricing. At the bottom left, there is a section titled "Benefits and features". The browser status bar at the bottom indicates the URL "us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/homepage", the region "United States (N. Virginia)", and the AWS account number "rsoaccount-new/68034028d24717cbb9270727 @ rsosandboxnew96".

The screenshot shows the "Topics" page under the Amazon SNS service. The left sidebar shows navigation options: Dashboard, Topics (selected), Subscriptions, Mobile (Push notifications, Text messaging (SMS)), and CloudShell. The main content area displays a table with one row, indicating "No topics". It includes a search bar, columns for "Name" and "Type", and buttons for "Edit", "Delete", "Publish message", and "Create topic". A blue banner at the top states "Amazon SNS now supports High Throughput FIFO topics. Learn more". The browser status bar at the bottom indicates the URL "us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/topics", the region "United States (N. Virginia)", and the AWS account number "rsoaccount-new/68034028d24717cbb9270727 @ rsosandboxnew96".

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

The "Create topic" wizard is displayed, showing the following steps:

- Details**:
  - Type**: Standard (selected)
  - 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
- Name**: Medtrack
- Display name - optional**: My Topic

A red error message at the top indicates: "Error code: AccessDeniedException - Error message: User: arnawssts: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-b273-946668080c88 because no identity-based policy allows the kms:DescribeKey action".

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

The "Topics" page for the "Amazon SNS" service shows the newly created topic "Medtrack".

The topic details are as follows:

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

A green success message states: "Topic Medtrack created successfully. You can create subscriptions and send messages to them from this topic.".

Screenshot of the AWS SNS console showing the details of a topic named "Medtrack".

**Details:**

- Name: Medtrack
- Display name: -
- ARN: arn:aws:sns:us-east-1:600627341644:Medtrack
- Topic owner: 600627341644
- 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 creation of a new subscription for the "Medtrack" topic.

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

**Protocol:** Email

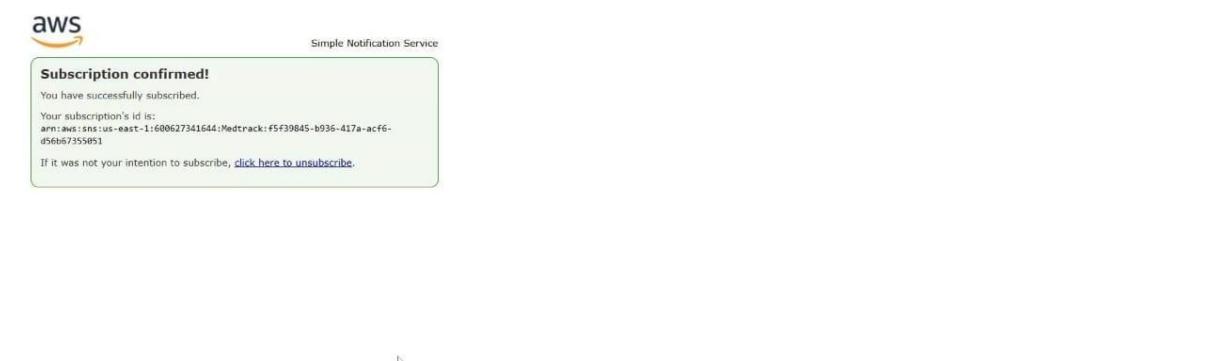
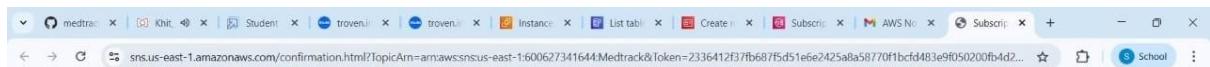
**Endpoint:** An email address that can receive notifications from Amazon SNS.  
228x1a4526@khitguntur.ac.in

**Info:** After your subscription is created, you must confirm it.

**Subscription filter policy - optional** (Info)  
This policy filters the messages that a subscriber receives.

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

**Actions:** Cancel, Create subscription



The screenshot shows the AWS Management Console for the Amazon SNS service. The URL in the address bar is [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](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 left sidebar shows navigation options like Dashboard, Topics, Subscriptions, Mobile, Push notifications, and Text messaging (SMS). The main content area shows a subscription named "Subscription: f5f39845-b936-417a-acf6-d56b67355051". A blue banner at the top says "New Feature: Amazon SNS now supports High Throughput FIFO topics. Learn more". A green banner below it says "Subscription to Medtrack created successfully. The ARN of the subscription is arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051". The subscription details table includes columns for ARN, Status (Pending confirmation), Protocol (EMAIL), and other fields like Endpoint and Topic. At the bottom, there are tabs for "Subscription filter policy" and "Redrive policy (dead-letter queue)".

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/armawssnsus-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051>. The page displays the details of a subscription named **f5f39845-b936-417a-acf6-d56b67355051** to the **Medtrack** topic. The ARN is listed as `arn:aws:sns:us-east-1:600627341644:Medtrack:f5f39845-b936-417a-acf6-d56b67355051`. The status is **Confirmed**. The endpoint is `228x1a4526@khitguntur.ac.in` and the protocol is **EMAIL**. The topic is **Medtrack**. The subscription principal is `arn:aws:iam::600627341644:role/rsoaccount-new`. Below the details, there are tabs for **Subscription filter policy** and **Redrive policy (dead-letter queue)**. The **Subscription filter policy** section indicates "No filter policy configured for this subscription. To apply a filter policy, edit this subscription." An **Edit** button is present.

## 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 displays a table of instances with one entry: **medtrack-server** (Instance ID: `i-0e439b38571daa87c`, Instance state: **Running**, Instance type: **t2.micro**, Status check: **2/2 checks passed**, Availability Zone: **us-east-1c**, Public IP: `ec2-13-2-`). Below the table, the details for the **i-0e439b38571daa87c (medtrack-server)** instance are shown. The **Security** tab is selected, showing the IAM Role **EC2\_MedTrack\_Role** and Owner ID **600627341644**. The **Launch time** is listed as **Fri Jul 04 2025 11:10:58 GMT+0530 (India Standard Time)**. The bottom of the screen shows the Windows taskbar with various icons and the date/time **04-07-2025 13:53**.

us-east-1.console.aws.amazon.com/ec2/home/?region=us-east-1#ConnectToInstance:instanceId=i-0e439b38571daa87c

EC2 Instances i-0e439b38571daa87c Connect to instance

**Connect Info**

Connect to an instance using the browser-based client.

**EC2 Instance Connect** Session Manager SSH client EC2 serial console

**Instance ID**  
i-0e439b38571daa87c (medtrack-server)

**Connect using a Public IP**  
Connect using a public IPv4 or IPv6 address

**Connect using a Private IP**  
Connect using a private IP address and a VPC endpoint

**Public IPv4 address**  
13.222.1.50

**IPv6 address**

**Username**  
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

**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.

Cancel Connect

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/sshd?region=us-east-1&conn... © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 13:53 ENG IN 04-07-2025

us-east-1.console.aws.amazon.com/ec2-instance-connect/sshd/home/?region=us-east-1&connType=standard&instanceId=i-0e439b38571daa87c&osUser=ec2-user&sshPort=22&addre... Star School

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 ~$
```

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 13:54 ENG IN 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 ~]# 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)  
PublicIPs: 13.222.1.50 PrivateIPs: 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
                                           1/1
                                           1/1

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

```
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-]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)

PublicIP: 13.222.1.50 PrivateIP: 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 14:04 04-07-2025

```
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-]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)

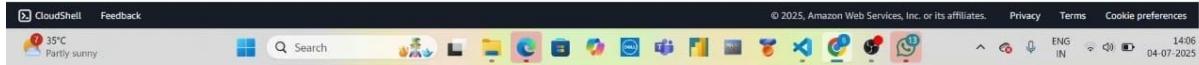
PublicIP: 13.222.1.50 PrivateIP: 172.31.17.153

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 boto3
  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>=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 kB)
[██████████] 11.8 kB 36.1 MB/s eta 0:00:01
```

i-0e439b38571daa87c (medtrack-server)

Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



i-0e439b38571daa87c (medtrack-server)

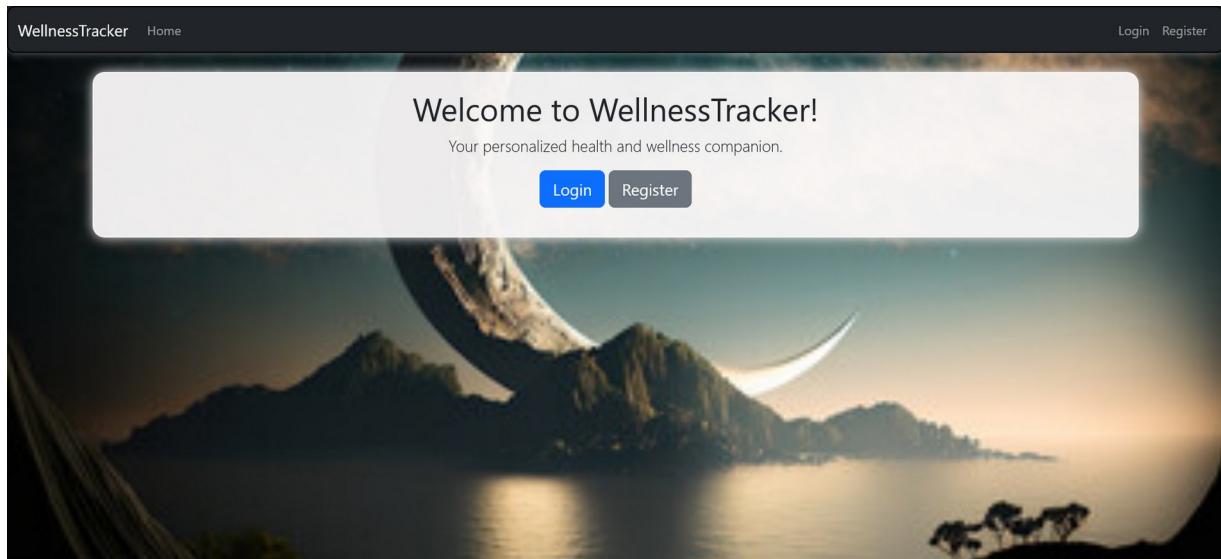
Public IPs: 13.222.1.50 Private IPs: 172.31.17.153



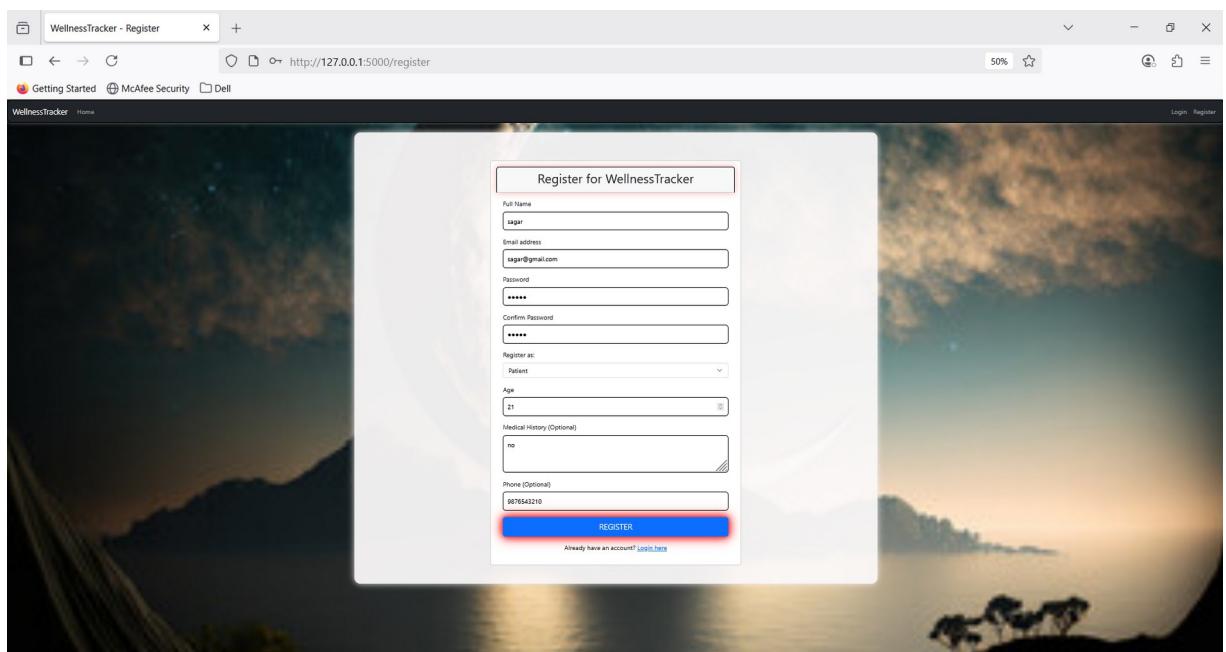
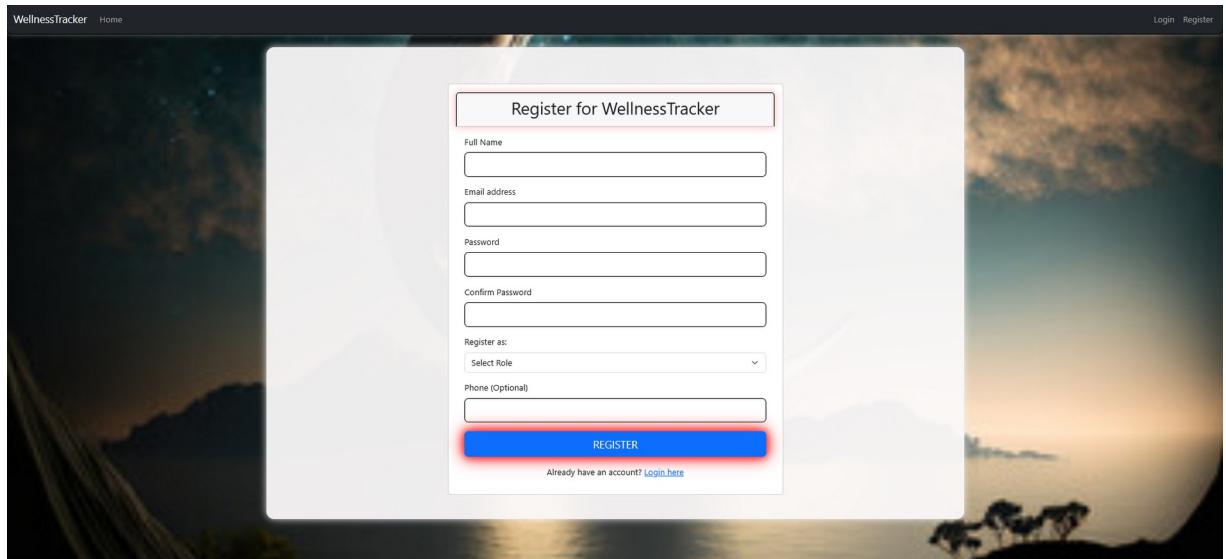


## Screenshots

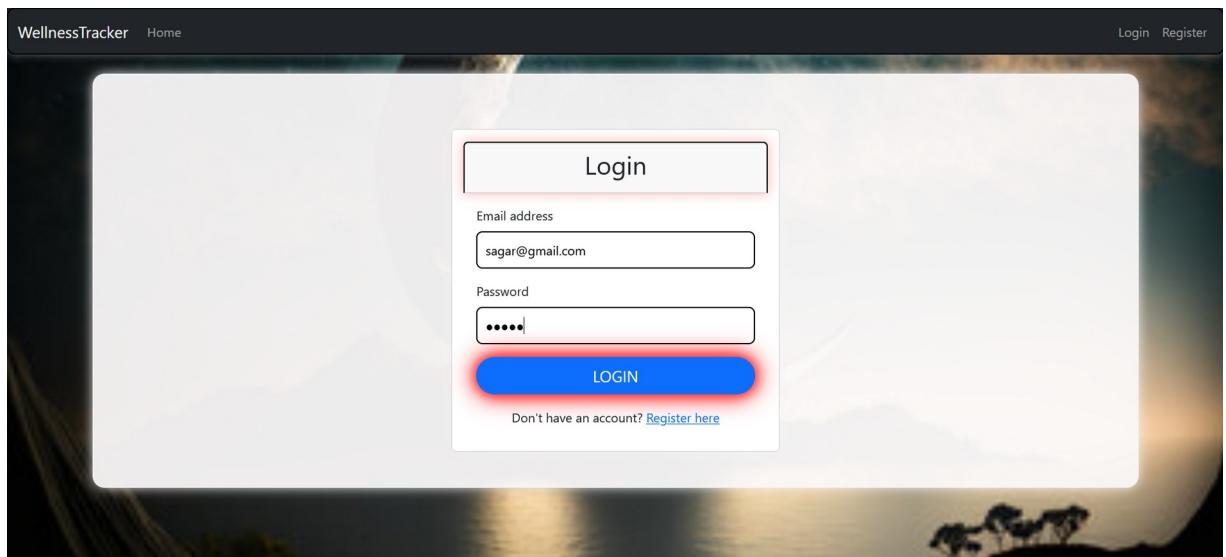
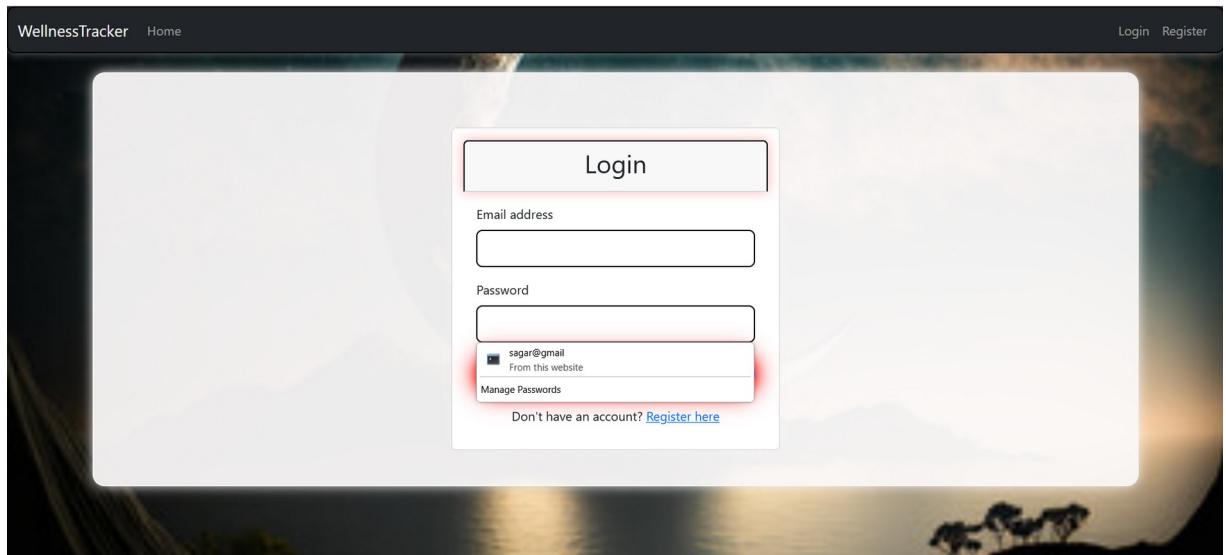
### 1. Homepage



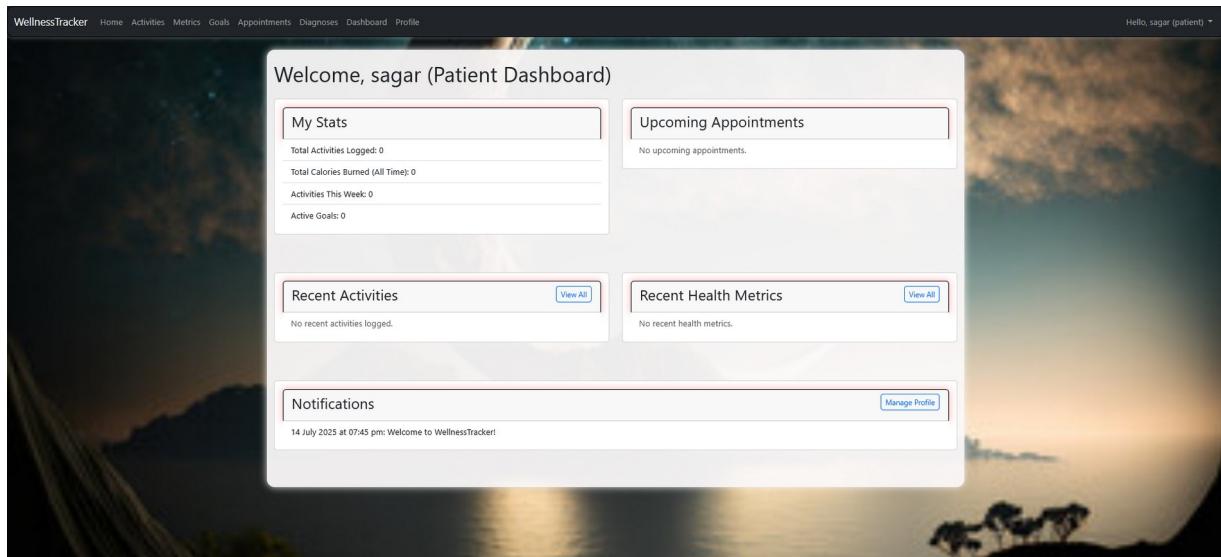
## 2. Register



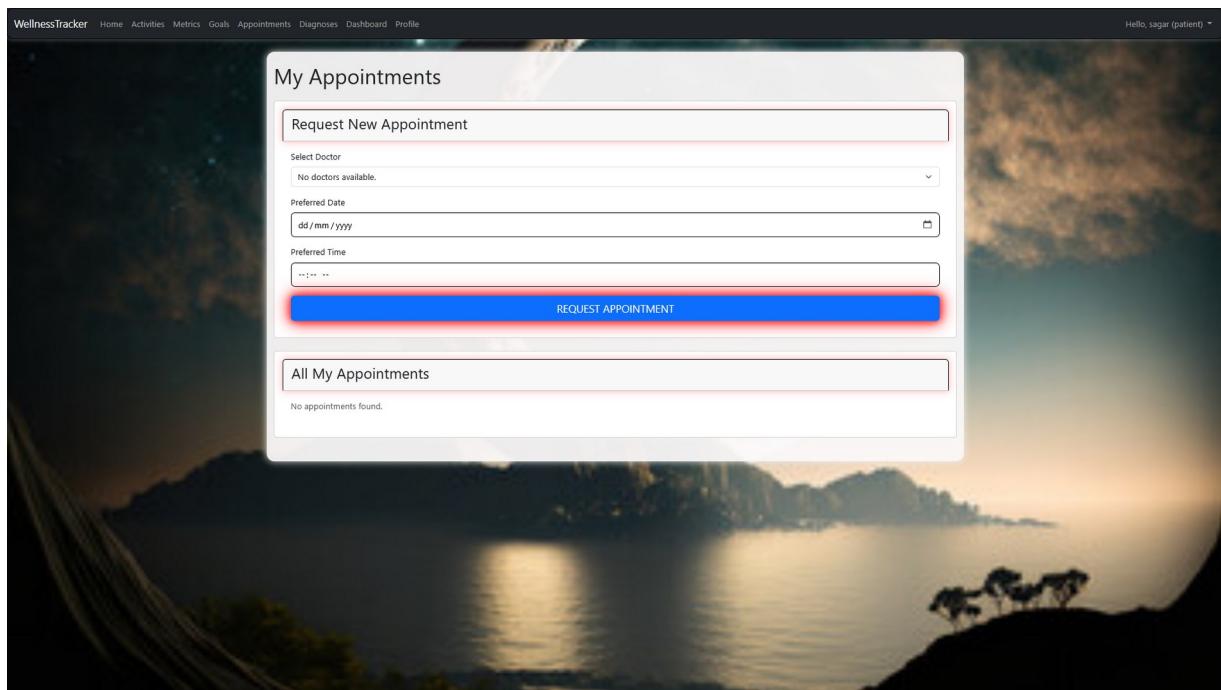
### 3.Login



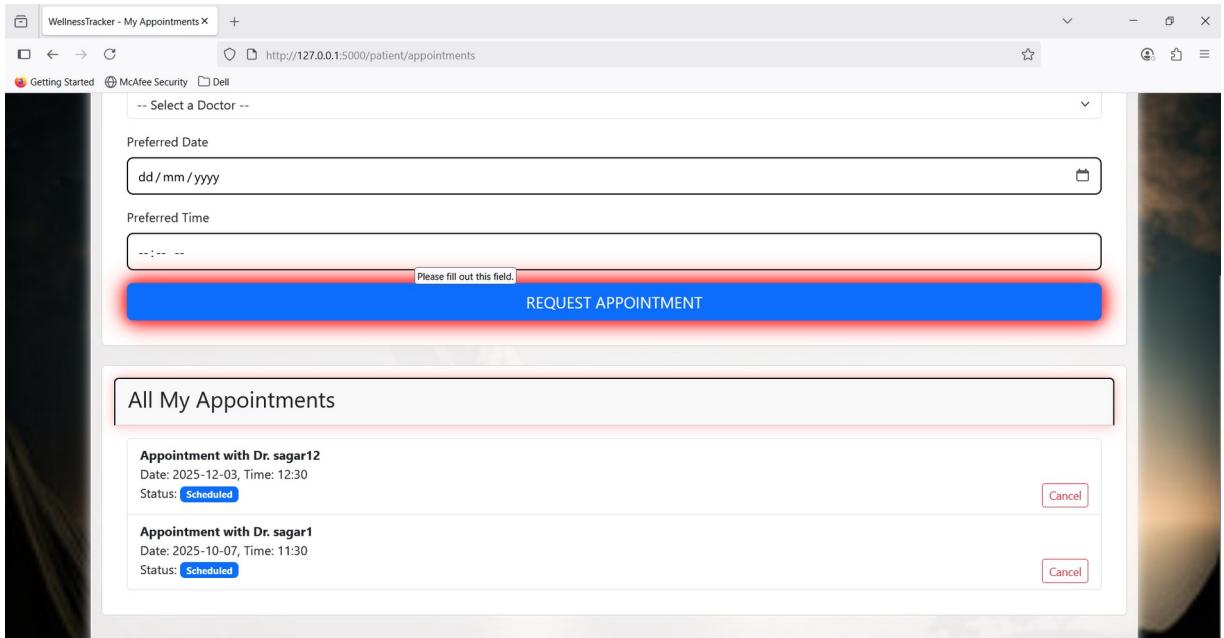
## 4. Patient Dashboard



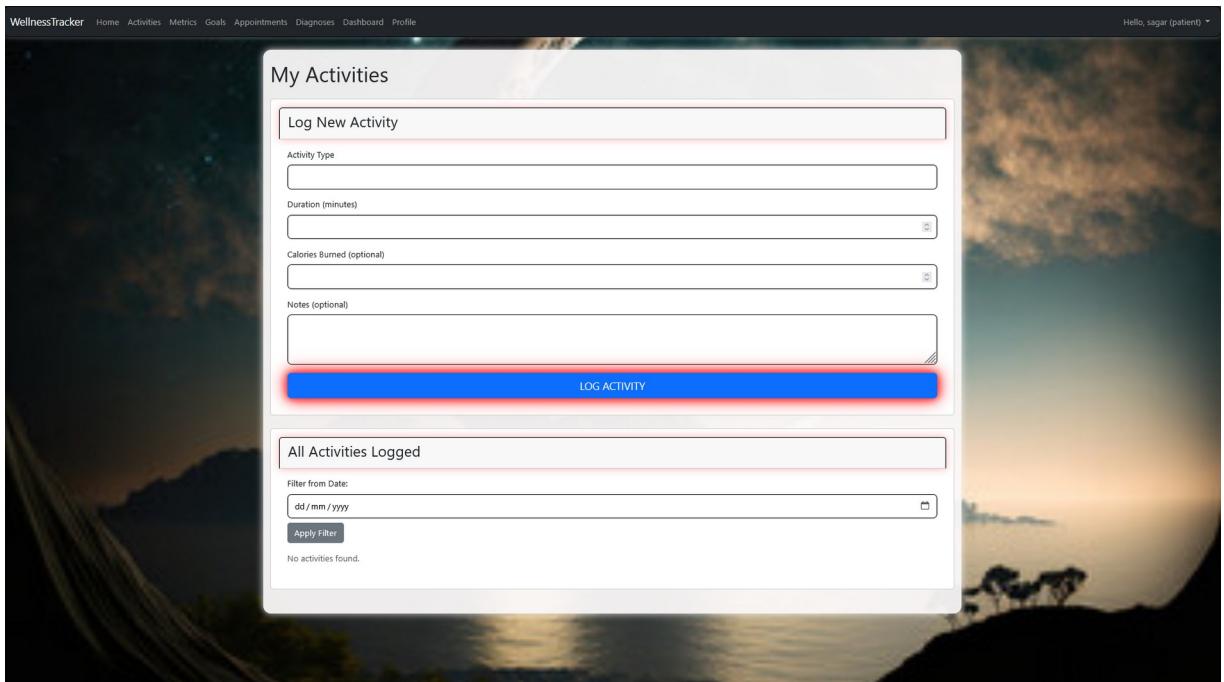
## 5. Appointment Booking Form



## 6. View Appointments



## 7. View My Activity



## 8. Doctor Dashboard

Welcome, Dr. sagar1 (Doctor Dashboard)

**My Stats**

- Total Patients Under Care: 2
- Specialization: Endocrinology
- Years of Experience: 0

**Upcoming Appointments**

- Appointment with sagar on 2025-03-11 at 11:30 (Status: Scheduled) [Manage](#)
- Appointment with sagar123 on 2025-03-12 at 12:30 (Status: Scheduled) [Manage](#)

**Recent Notifications**

- 14 July 2025 at 09:49 pm: Welcome to MedTrack as a Doctor!
- 14 July 2025 at 09:51 pm: New appointment requested by sagar on 2025-03-11 at 11:30.
- 14 July 2025 at 09:52 pm: New appointment requested by sagar123 on 2025-03-12 at 12:30.

**My Patients**

- sagar123 (sagar123@gmail.com) [View Details](#)
- sagar (sagar@gmail.com) [View Details](#)

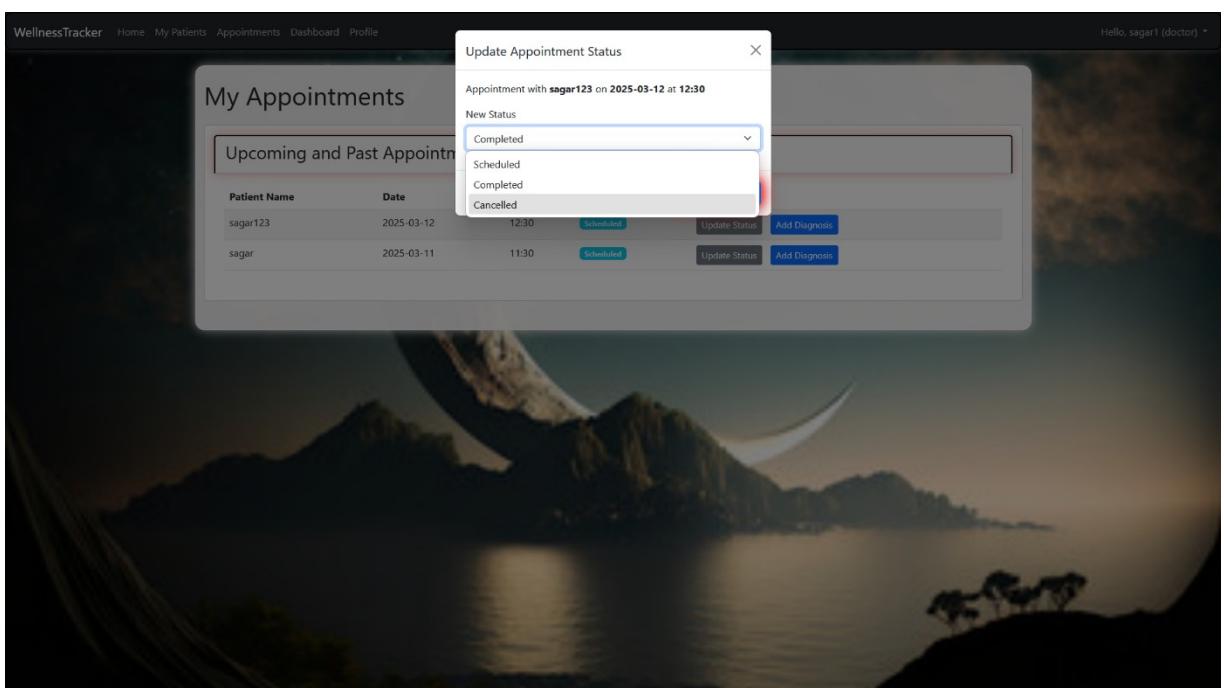
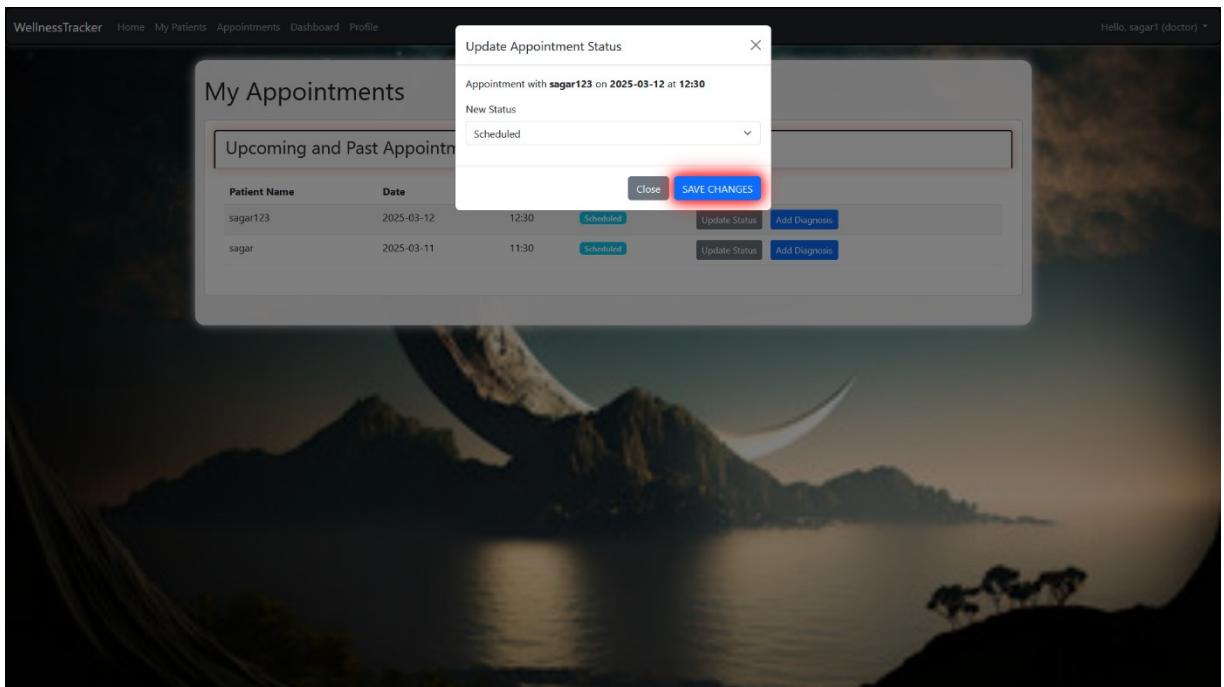
## 9. View Your Appointments

My Appointments

**Upcoming and Past Appointments**

Patient Name	Date	Time	Status	Actions
sagar123	2025-03-12	12:30	Scheduled	<a href="#">Update Status</a> <a href="#">Add Diagnosis</a>
sagar	2025-03-11	11:30	Scheduled	<a href="#">Update Status</a> <a href="#">Add Diagnosis</a>

# 10. Submit Appointment

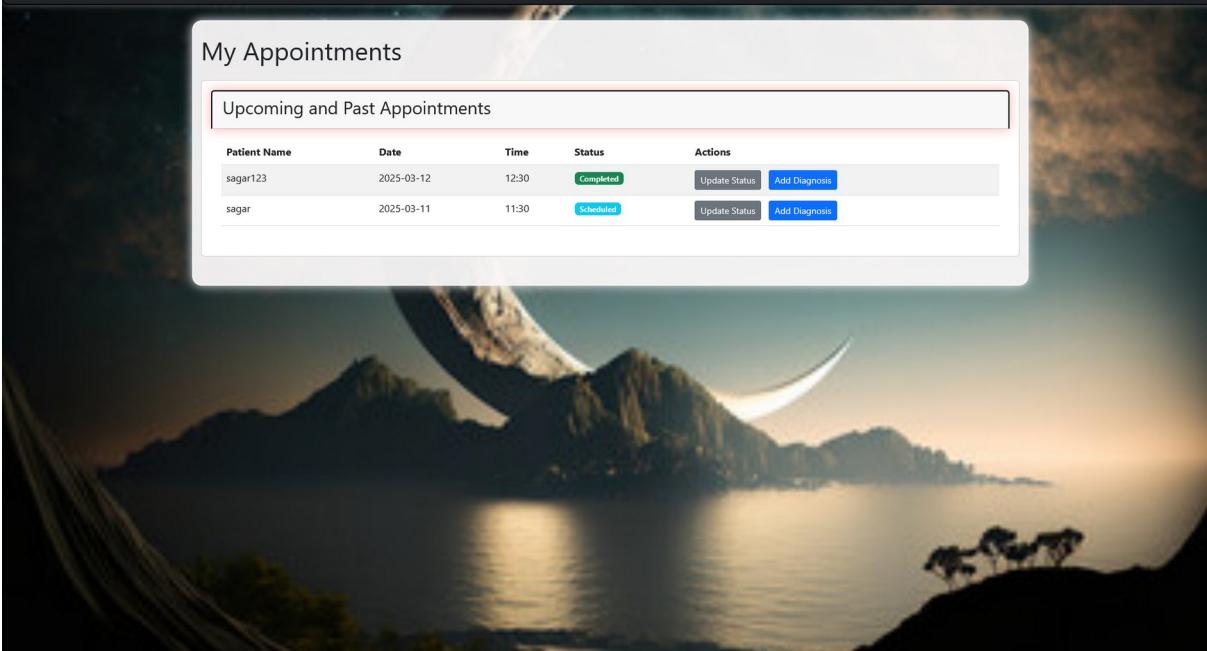


WellnessTracker Home My Patients Appointments Dashboard Profile Hello, sagar1 (doctor) ▾

### My Appointments

Upcoming and Past Appointments

Patient Name	Date	Time	Status	Actions
sagar123	2025-03-12	12:30	Completed	<a href="#">Update Status</a> <a href="#">Add Diagnosis</a>
sagar	2025-03-11	11:30	Scheduled	<a href="#">Update Status</a> <a href="#">Add Diagnosis</a>



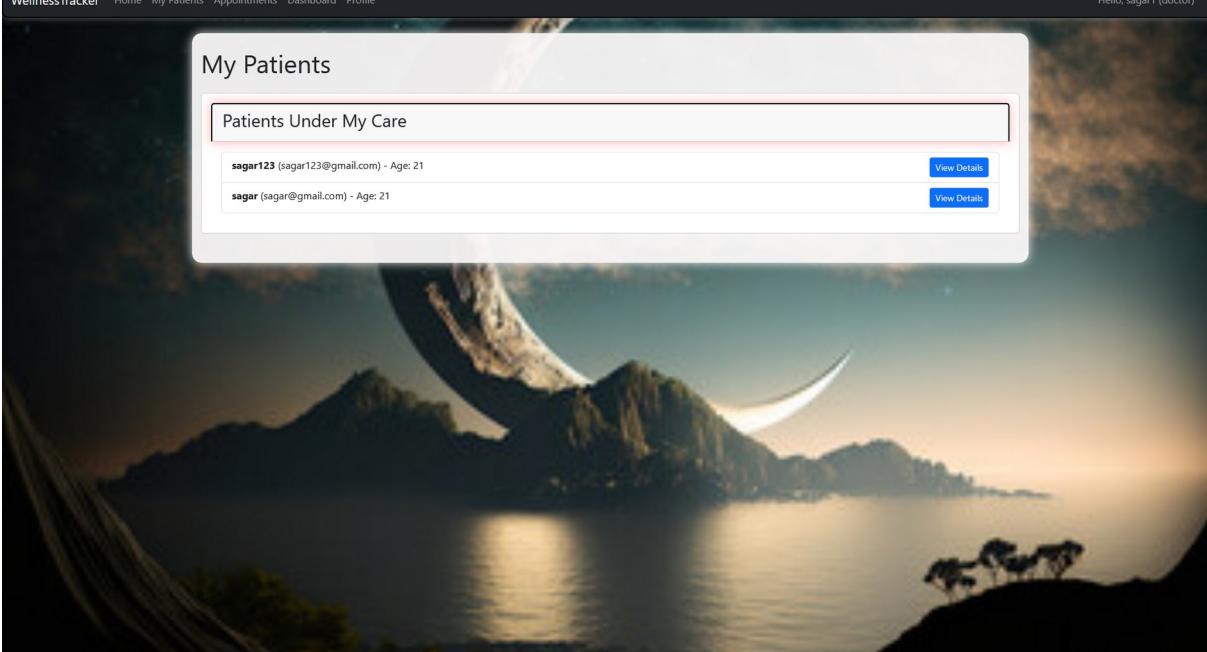
## 11. View My Patients

WellnessTracker Home My Patients Appointments Dashboard Profile Hello, sagar1 (doctor) ▾

### My Patients

Patients Under My Care

sagar123 (sagar123@gmail.com) - Age: 21	<a href="#">View Details</a>
sagar (sagar@gmail.com) - Age: 21	<a href="#">View Details</a>



## **Demo Video**

<https://drive.google.com/file/d/1vvJaIRBO4IN91AlxtviOcKQ4doMKDKO/view>

## **GitHub Repository**

<https://github.com/228x1a4270/MEDTRACK-SMAPLE>

## **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:** KONDAPALLI SAGAR

**Roll No:** 228X1A4270

**Branch :** AI&ML

**College Name :** KHIT

**SmartBridge AWS Internship Project - 2025**