Project Description:

Capturemoments is a digital platform designed to streamline the booking of Photographers for University and personal events. It eliminates the traditional manual scheduling system by offering an automated, user-friendly interface where People and faculty can browse, book and receive real-time updates from Phographers.

The System is developed using Flask for backend operations, Amazon Ec2 for hosting, DynamoDB for storing booking details, and SNS (Simple Notification Service) for email notifications. The platform ensures efficient booking, timely communication, and scalable operations, especially during peak seasons like graduation, festivals, and academic events.

Scenario 1: Photographer Booking Form Submission
The system provides a simple booking form interface. A user selects the photographer type, user ID, event date, and price. Once the form is submitted, the backend Flask application stores the data into DynamoDB.

Scenario 2: Successful Booking Confirmation
Upon submission, users are redirected to a confirmation screen that displays:

- Booking success message
- Photographer type
- Date
- Price

Scenario 3: Viewing Booking History
Users can navigate to the Booking History Page to view their previous bookings. The history includes:

- Photographer Type
- User ID
- Date
- Price

Scenario 4: Homepage for Navigation
The Homepage welcomes users and provides direct links to all important pages:

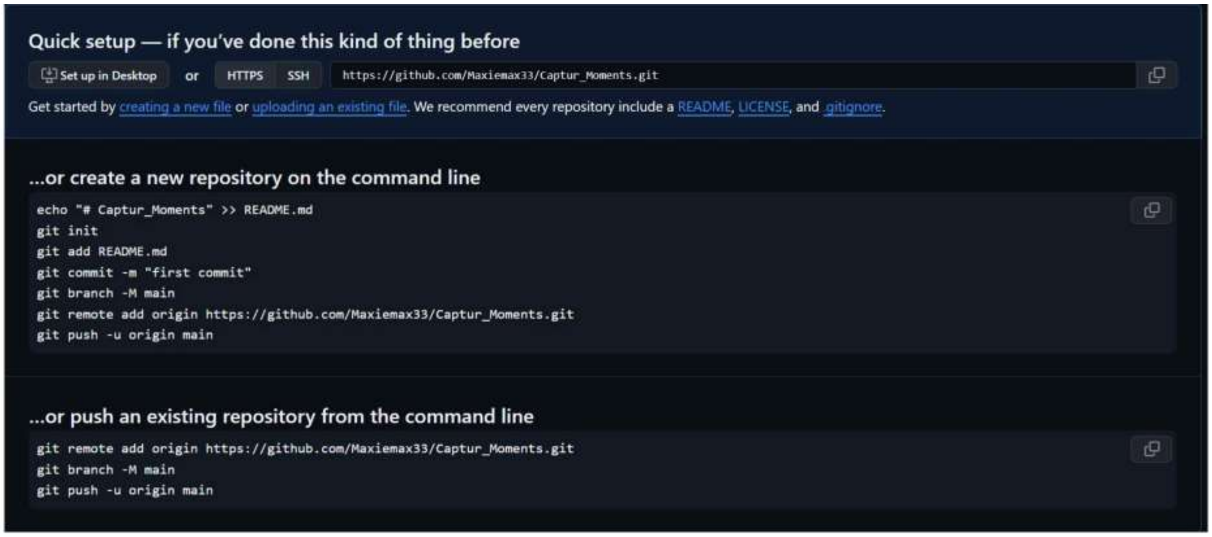
- Home
- Book Now
- Booking History

Pre-Requisites

1. .AWS Account Setup: AWS Account Setup
2. Understanding IAM: IAM Overview
3. Amazon EC2 Basics: EC2 Tutorial
4. DynamoDB Basics: DynamoDB Introduction
5. SNS Overview: SNS Documentation
6. Git Version Control: Git Documentation

Project Work Flow

1.UPLOAD ALL FILES ON GITHUB



Quick setup — if you've done this kind of thing before

```
https://github.com/Maxiemax33/Captur_Moments.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

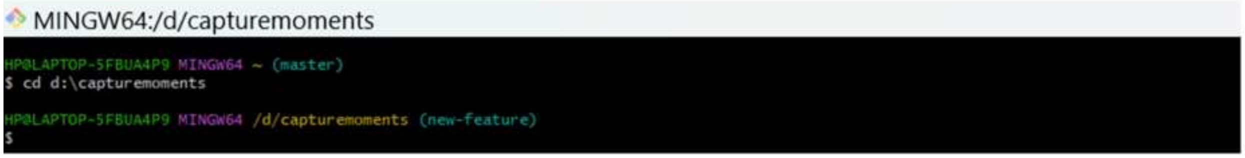
...or create a new repository on the command line

```
echo "# Captur_Moments" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Maxiemax33/Captur_Moments.git
git push -u origin main
```

...or push an existing repository from the command line

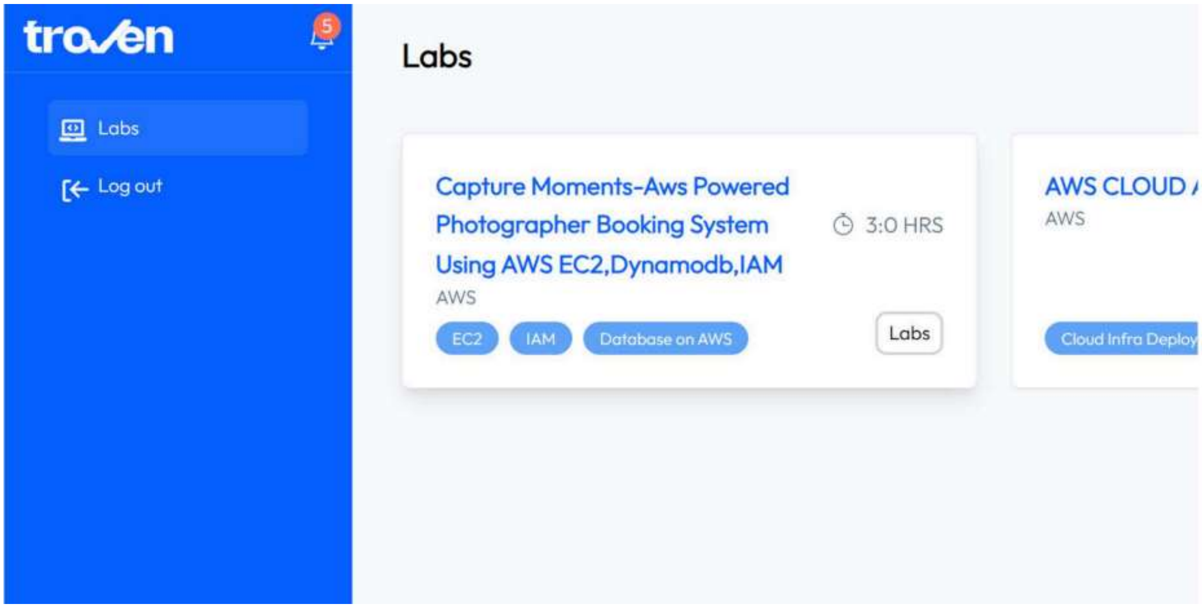
```
git remote add origin https://github.com/Maxiemax33/Captur_Moments.git
git branch -M main
git push -u origin main
```

Git Bash path

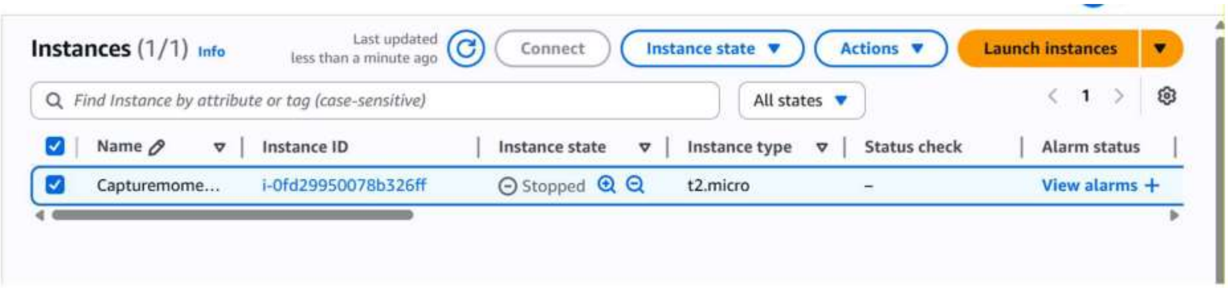


```
HP@LAPTOP-5FBUA4P9 MINGW64 ~ (master)
$ cd d:\capturemoments

HP@LAPTOP-5FBUA4P9 MINGW64 /d/capturemoments (new-feature)
$
```

2.NOW BEGIN WITH YOUR TROVEN LOGIN AND ACCESS



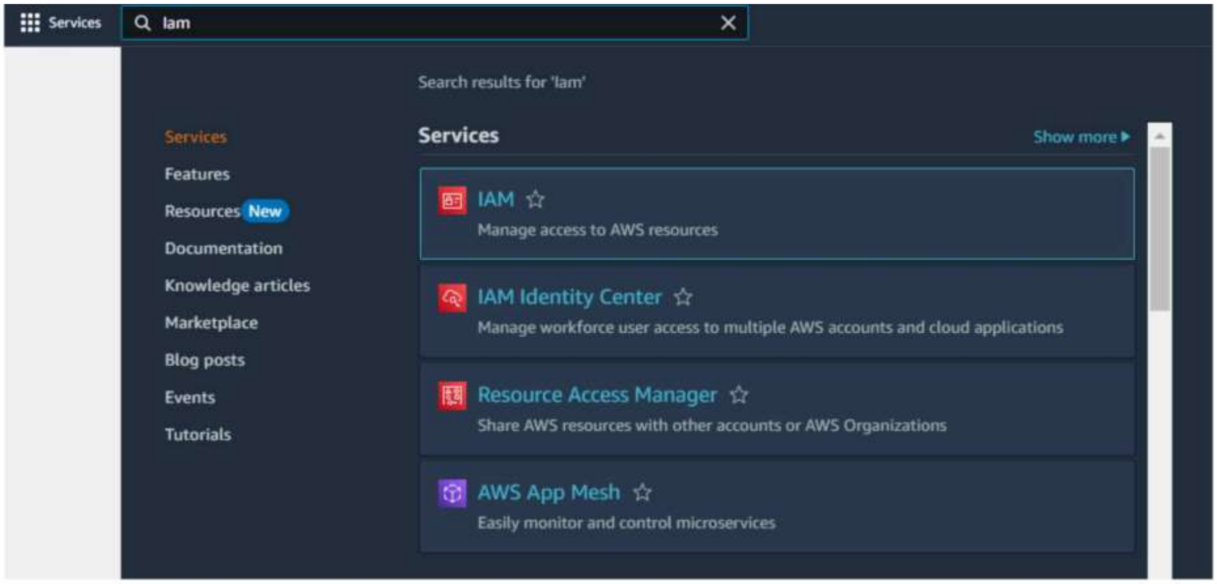
3.INITIALISED THE EC2 INSTANCE


Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	Capturemome...	i-0fd29950078b326ff	Stopped	t2.micro	–	View alarms +

4.CREATED THE IAM ROLE

In CreaƟon of IAM Role



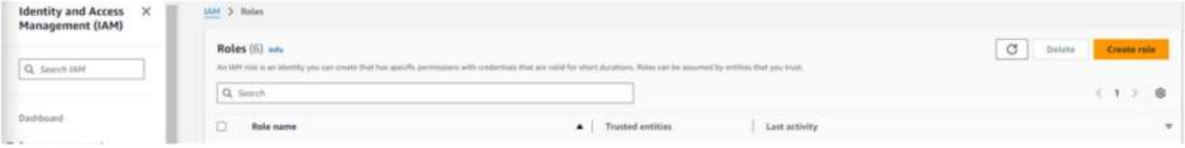
Search results for 'Iam'

IAM ☆
Manage access to AWS resources

IAM Identity Center ☆
Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager ☆
Share AWS resources with other accounts or AWS Organizations

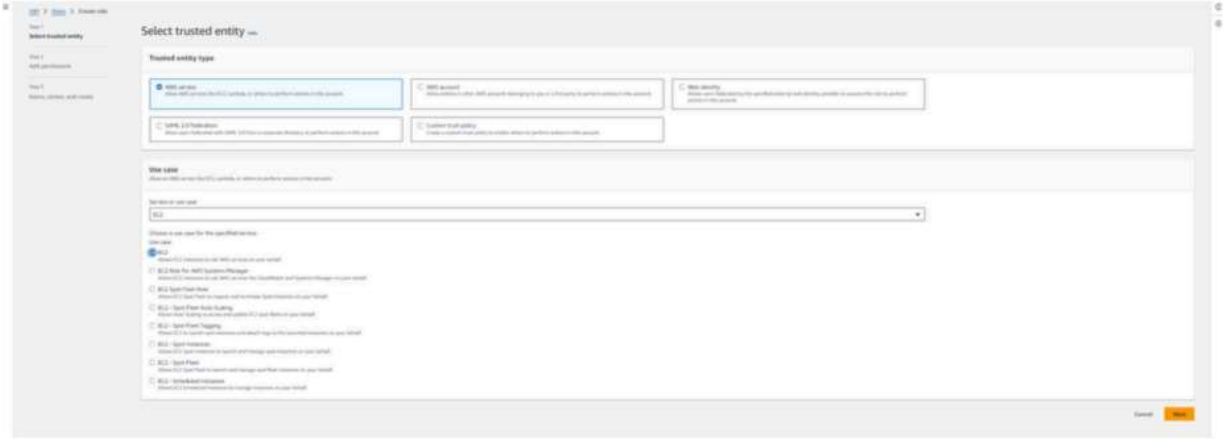
AWS App Mesh ☆
Easily monitor and control microservices



Roles (6)

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



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service [illegible]

○ AWS account [illegible]

○ Web identity [illegible]

○ SAML 2.0 federation [illegible]

○ Custom trust policy [illegible]

Use case

[illegible]

Service or use case

EC2

Choose a use case for the specified service.

Use case

● EC2 [illegible]

○ EC2 Role for AWS Systems Manager [illegible]

○ EC2 Spot Fleet Role [illegible]

○ EC2 - Spot Fleet Auto Scaling [illegible]

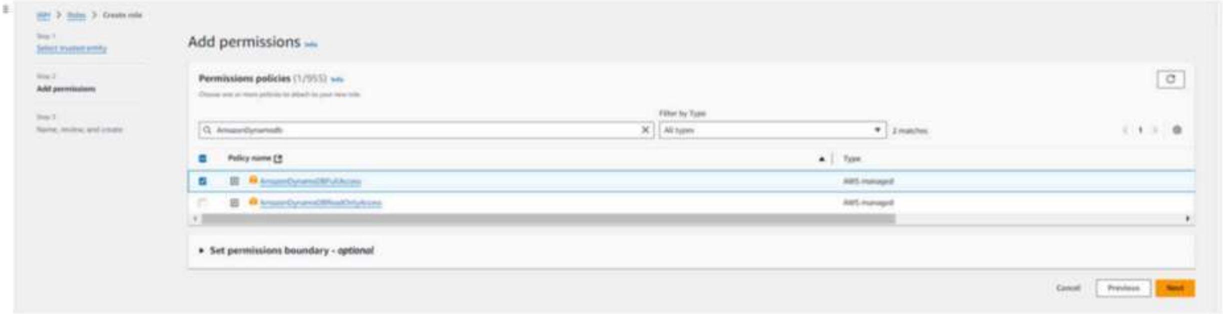
○ EC2 - Spot Fleet Tagging [illegible]

○ EC2 - Spot Instances [illegible]

○ EC2 - Spot Fleet [illegible]

○ EC2 - Scheduled Instances [illegible]

Cancel Next



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/955) Info

Choose one or more policies to attach to your new role.

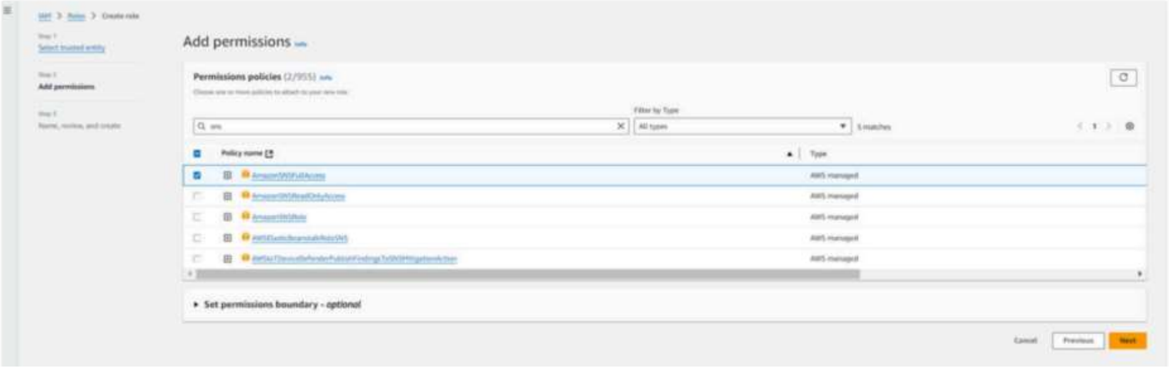
Filter by Type

AmazonDynamodb | All types | 2 matches

	Policy name	Type
☑	AmazonDynamoDBFullAccess	AWS managed
□	AmazonDynamoDBReadOnlyAccess	AWS managed

▸ **Set permissions boundary - *optional***

Cancel Previous Next



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (2/955) Info

Choose one or more policies to attach to your new role.

Filter by Type

sns | All types | 5 matches

	Policy name	Type
☑	AmazonSNSFullAccess	AWS managed
□	AmazonSNSReadOnlyAccess	AWS managed
□	AmazonSNSRole	AWS managed
□	AWSElasticBeanstalkRoleSNS	AWS managed
□	AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction	AWS managed

▸ **Set permissions boundary - *optional***

Cancel Previous Next

Trusted entity type: AWS service

- Use case:
- EC2 Attached permission:
- By ModifyIAM Role AmazonDynamoDBFullAccess

5.CREATED DYNAMODB TABLES

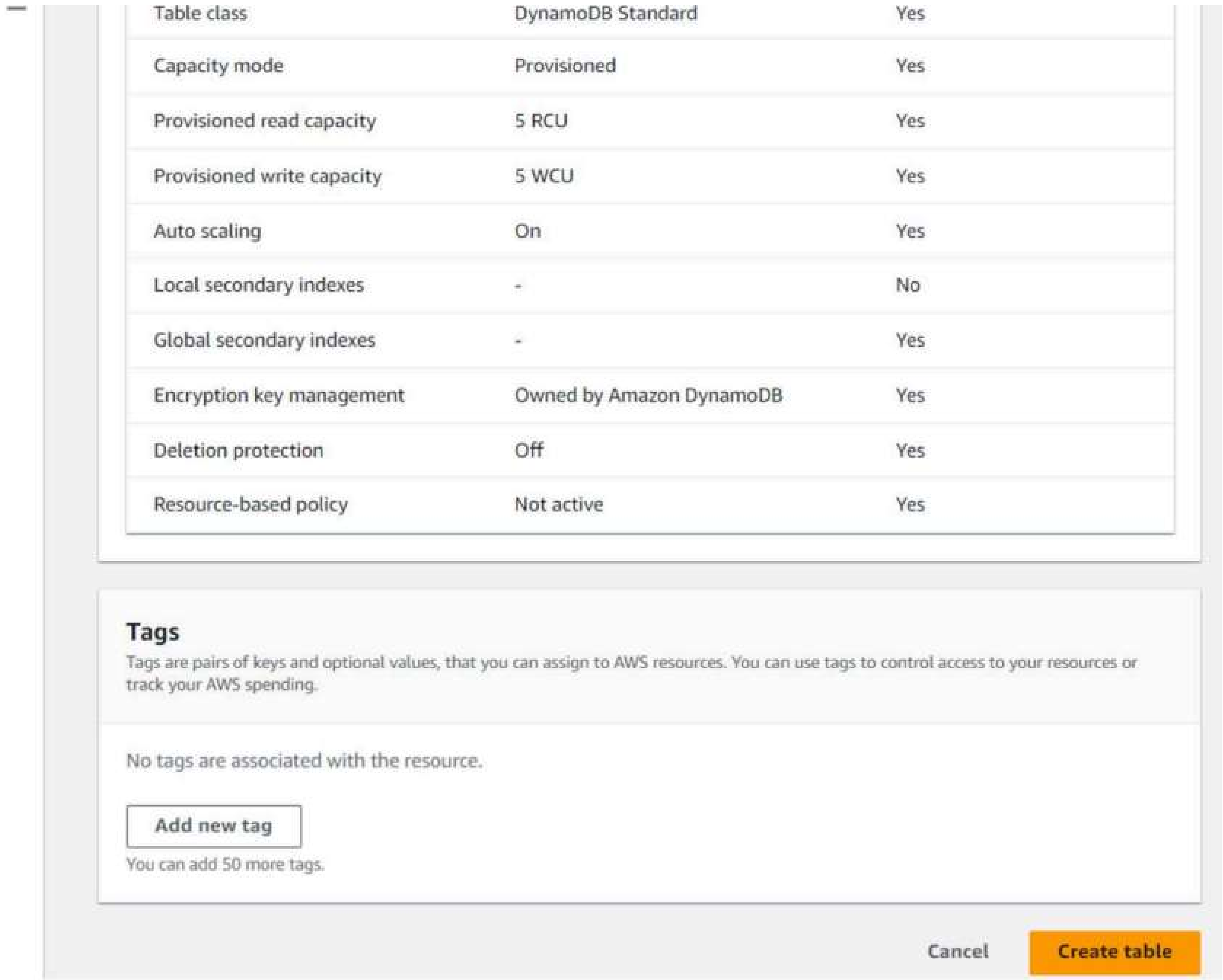


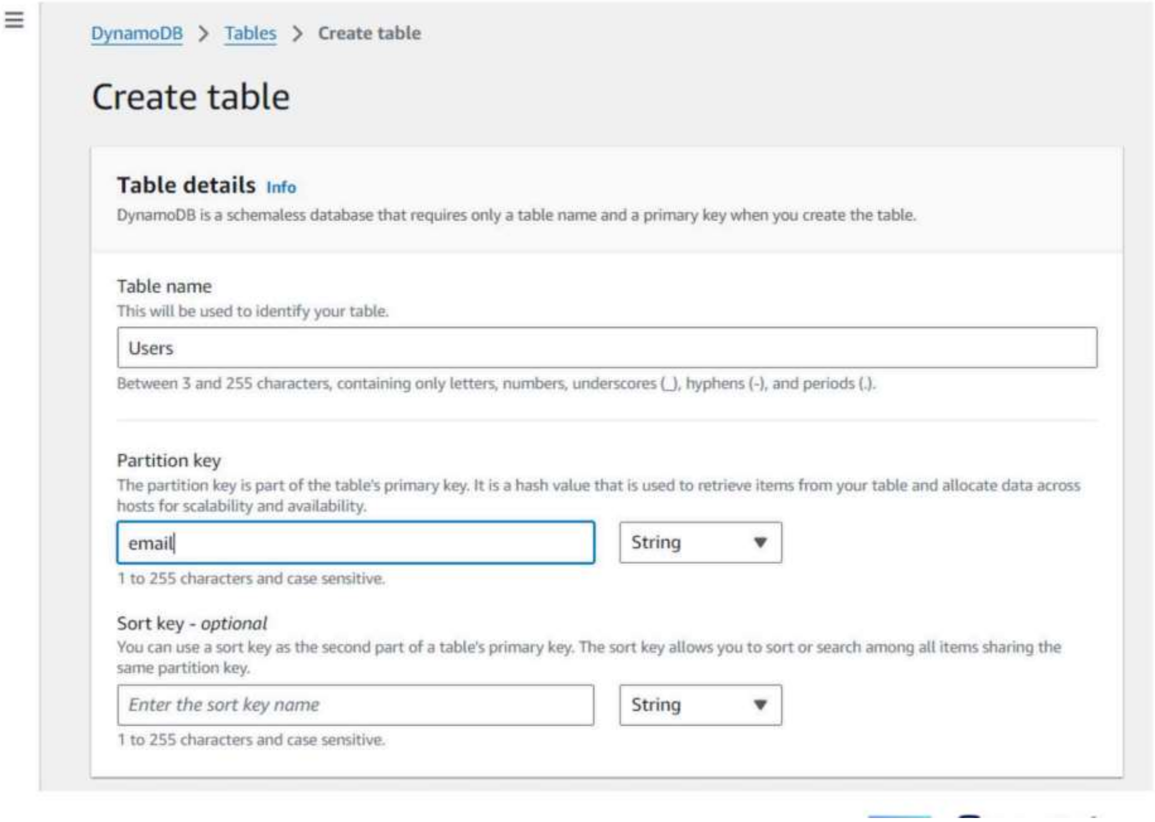
Table class	DynamoDB Standard	Yes
Capacity mode	Provisioned	Yes
Provisioned read capacity	5 RCU	Yes
Provisioned write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes
Resource-based 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.

You can add 50 more tags.



DynamoDB > Tables > Create table

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.

Users

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.

email | String ▼

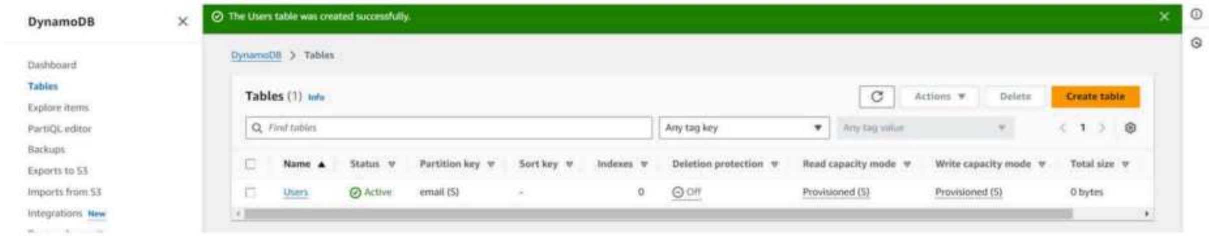
1 to 255 characters and case sensitive.

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.



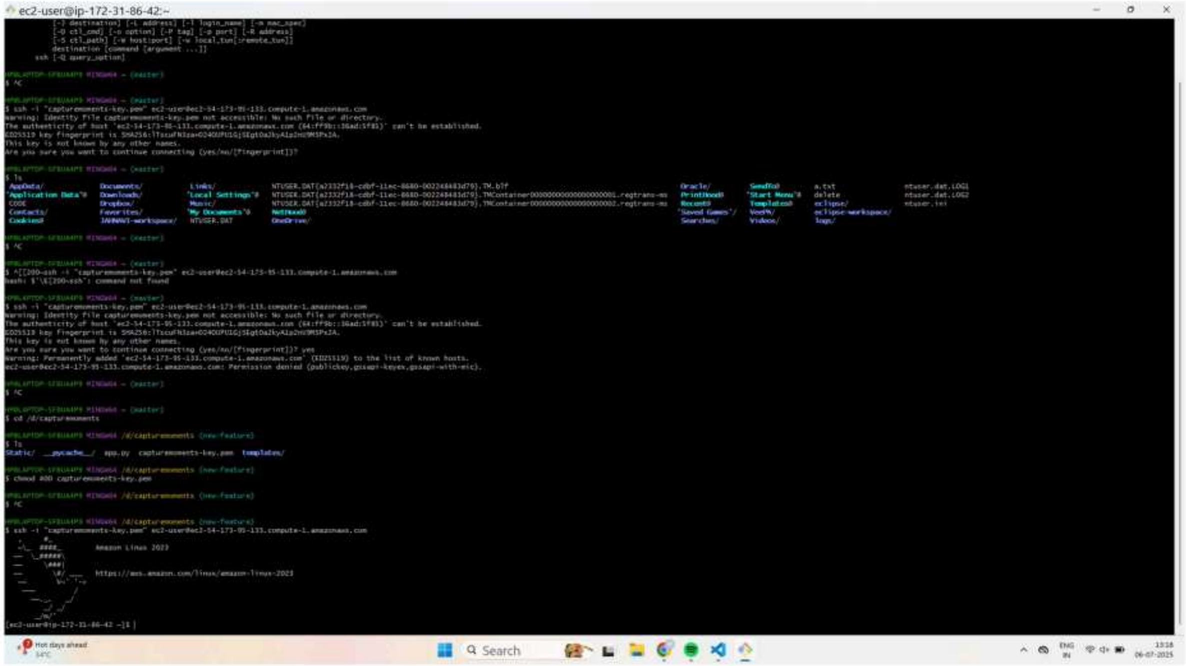
The Users table was created successfully.

DynamoDB > Tables

Tables (1) Info

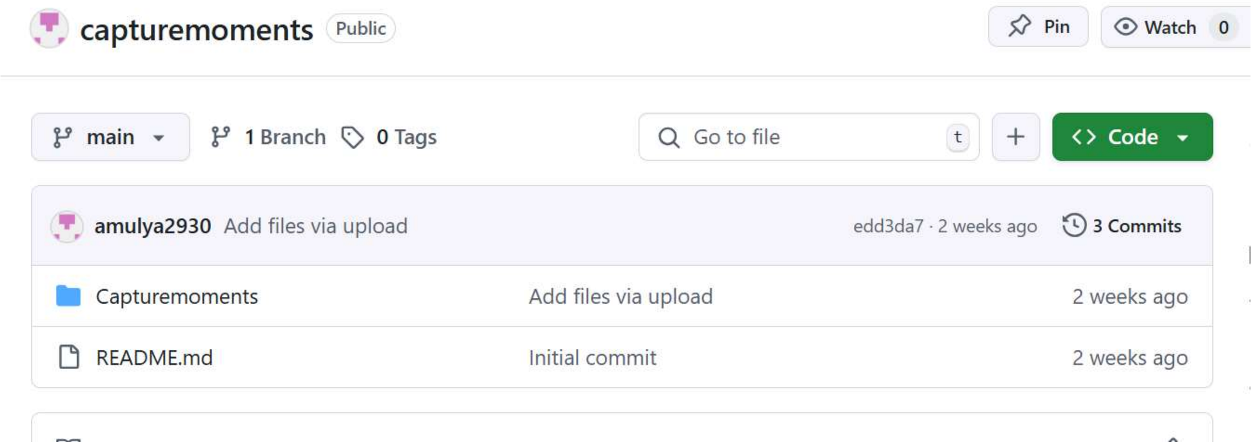
☐	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode	Total size
☐	Users	✓ Active	email (S)	-	0	Off	Provisioned (5)	Provisioned (5)	0 bytes

6.STARTED INSTANCE IN GITBASH AND INSTALL NECESSARY DEPENDENCIES



```
ec2-user@ip-172-31-86-42:~
           [-J destination] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-P tag] [-p port] [-R address]
           [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           destination [command [argument ...]]
       ssh [-Q query_option]

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ^C

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ssh -i "capturemoments-key.pem" ec2-user@ec2-54-173-95-133.compute-1.amazonaws.com
Warning: Identity file capturemoments-key.pem not accessible: No such file or directory.
The authenticity of host 'ec2-54-173-95-133.compute-1.amazonaws.com (64:ff9b::36ad:5f85)' can't be established.
ED25519 key fingerprint is SHA256:lTscuFN3zaeO24OUPU1GjSEgtDa2kyA1p2nU9MSPx2A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ls
 AppData/               Documents/              Links/                NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf                                   Oracle/          SendTo@        a.txt                ntuser.dat.LOG1
'Application Data'@     Downloads/             'Local Settings'@      NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000001.regtrans-ms   PrintHood@     'Start Menu'@    delete               ntuser.dat.LOG2
 CODE                   Dropbox/                Music/                NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000002.regtrans-ms   Recent@         Templates@     eclipse/             ntuser.ini
 Contacts/              Favorites/             'My Documents'@        NetHood@                                                                                 'Saved Games'/   VeePN/          eclipse-workspace/
 Cookies@               JAHNAVI-workspace/      NTUSER.DAT            OneDrive/                                                                                 Searches/        Videos/         logs/

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ^C

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ^[[200~ssh -i "capturemoments-key.pem" ec2-user@ec2-54-173-95-133.compute-1.amazonaws.com
bash: $'\E[200~ssh': command not found

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ssh -i "capturemoments-key.pem" ec2-user@ec2-54-173-95-133.compute-1.amazonaws.com
Warning: Identity file capturemoments-key.pem not accessible: No such file or directory.
The authenticity of host 'ec2-54-173-95-133.compute-1.amazonaws.com (64:ff9b::36ad:5f85)' can't be established.
ED25519 key fingerprint is SHA256:lTscuFN3zaeO24OUPU1GjSEgtDa2kyA1p2nU9MSPx2A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-95-133.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
ec2-user@ec2-54-173-95-133.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ ^C

HP@LAPTOP-SFBUA4P9 MINGW64 ~ (master)
$ cd /d/capturemoments

HP@LAPTOP-SFBUA4P9 MINGW64 /d/capturemoments (new-feature)
$ ls
Static/  __pycache__/  app.py  capturemoments-key.pem  templates/

HP@LAPTOP-SFBUA4P9 MINGW64 /d/capturemoments (new-feature)
$ chmod 400 capturemoments-key.pem

HP@LAPTOP-SFBUA4P9 MINGW64 /d/capturemoments (new-feature)
$ ^C

HP@LAPTOP-SFBUA4P9 MINGW64 /d/capturemoments (new-feature)
$ ssh -i "capturemoments-key.pem" ec2-user@ec2-54-173-95-133.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-86-42 ~]$ 
```

UPLOADED GITHUB REPOSITORY FILES



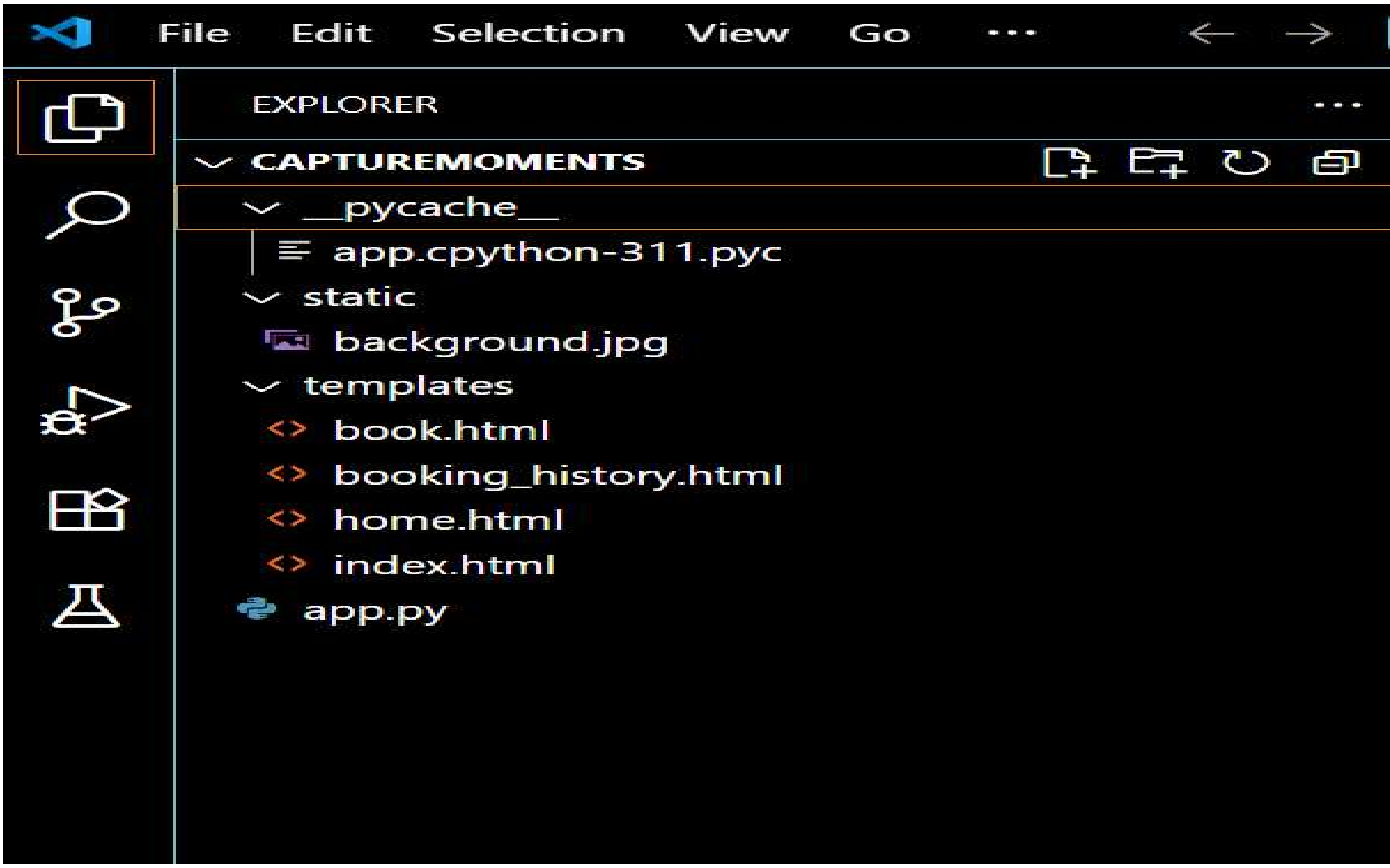
capturemoments Public

amulya2930 Add files via upload edd3da7 · 2 weeks ago 3 Commits

Capturemoments	Add files via upload	2 weeks ago
README.md	Initial commit	2 weeks ago

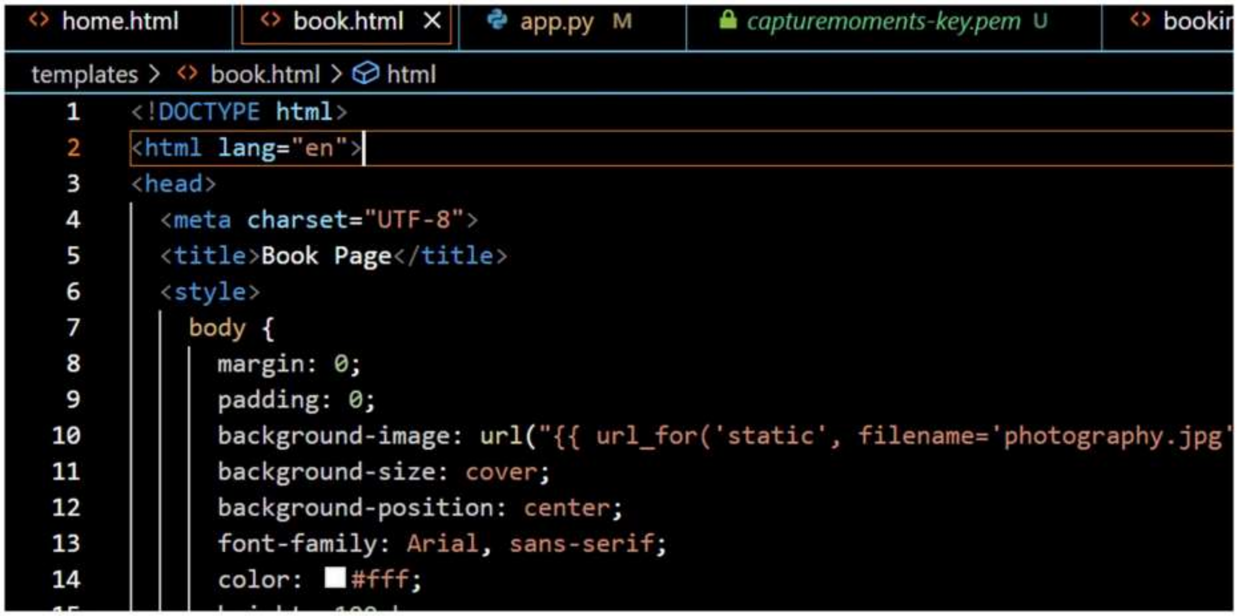
CAPTURE MOMENTS BOOKING SYSTEM PROJECT

MILESTONE 1: The Project Files in VISUAL STUDIO



```
CAPTUREMOMENTS
  __pycache__
    app.cpython-311.pyc
  static
    background.jpg
  templates
    book.html
    booking_history.html
    home.html
    index.html
  app.py
```

MILESTONE 2: WORKING OF book.html code



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Book Page</title>
  <style>
    body {
      margin: 0;
      padding: 0;
      background-image: url("{{ url_for('static', filename='photography.jpg'
      background-size: cover;
      background-position: center;
      font-family: Arial, sans-serif;
      color: #fff;
```

<meta charset="UTF-8">

- Ensures your webpage uses UTF-8 encoding to support most characters and symbols

<title>Book Page</title>

- Title of the browser tab.

<style>...</style>

This block defines CSS styling for the page.

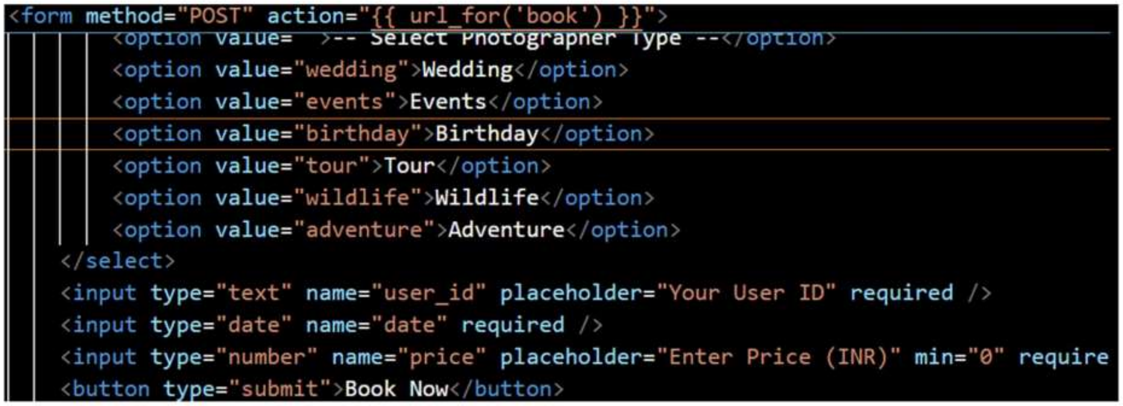
Background: Dynamically loads a photo using Flask's url_for() from the static folder.

url_for('static', filename='photography.jpg')

.

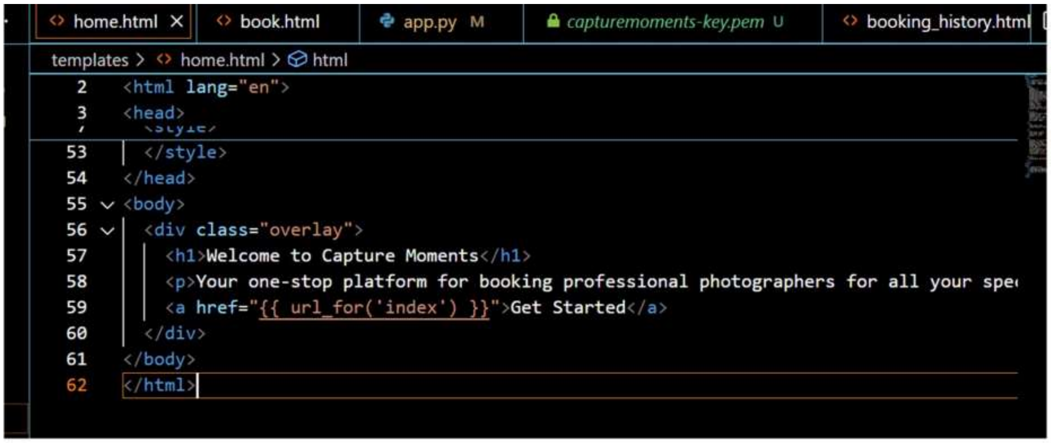
Working of the booking system form is

- Booking photographer
- Select the event
- Selecting the date of event
- Entering the price
- Book Now



```
<form method="POST" action="{{ url_for('book') }}">
            <option value="">-- Select Photographer Type --</option>
            <option value="wedding">Wedding</option>
            <option value="events">Events</option>
            <option value="birthday">Birthday</option>
            <option value="tour">Tour</option>
            <option value="wildlife">Wildlife</option>
            <option value="adventure">Adventure</option>
        </select>
        <input type="text" name="user_id" placeholder="Your User ID" required />
        <input type="date" name="date" required />
        <input type="number" name="price" placeholder="Enter Price (INR)" min="0" require
        <button type="submit">Book Now</button>
```

MILESTONE 3: INTRODUCTION OF THE WEBSITE AT home.html

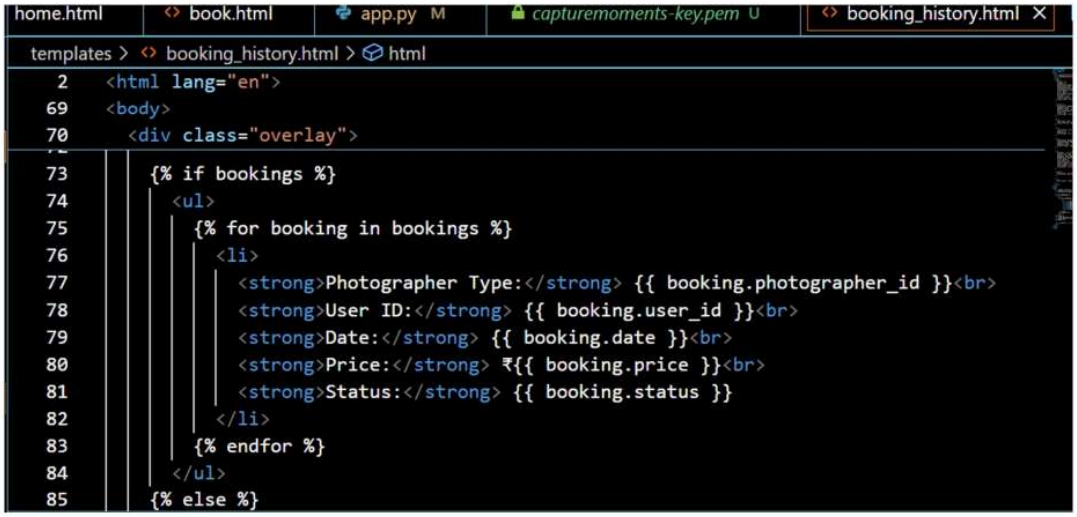


```
<html lang="en">
<head>
  <style>
  </style>
</head>
<body>
  <div class="overlay">
    <h1>Welcome to Capture Moments</h1>
    <p>Your one-stop platform for booking professional photographers for all your spe
    <a href="{{ url_for('index') }}">Get Started</a>
  </div>
</body>
</html>
```

- It visualise the welcome home page of capture moments project

This Home page gives a small description of the Capture Moments booking System Project.

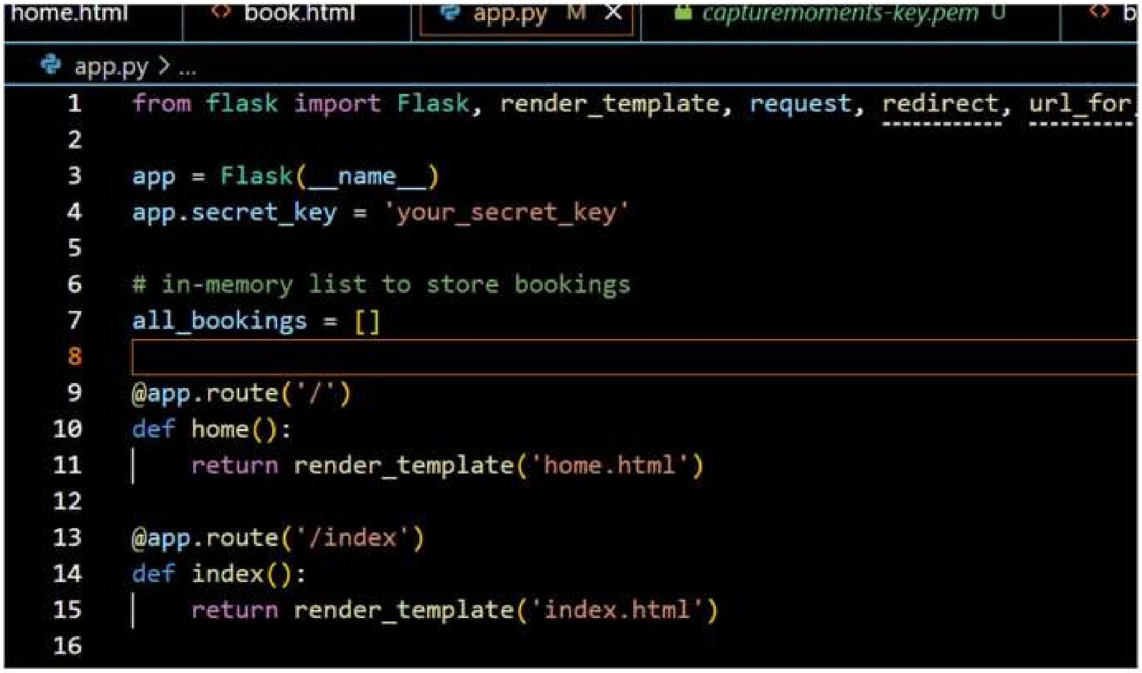
MILESTONE4: STORING THE DATA OF BOOKING SYSTEM AND SHOWS THE COPY of booking_history. html



```
<html lang="en">
<body>
  <div class="overlay">
    {% if bookings %}
      <ul>
        {% for booking in bookings %}
          <li>
            <strong>Photographer Type:</strong> {{ booking.photographer_id }}<br>
            <strong>User ID:</strong> {{ booking.user_id }}<br>
            <strong>Date:</strong> {{ booking.date }}<br>
            <strong>Price:</strong> ₹{{ booking.price }}<br>
            <strong>Status:</strong> {{ booking.status }}
          </li>
        {% endfor %}
      </ul>
    {% else %}
```

MILESTONE 5: WORKING OF app.py

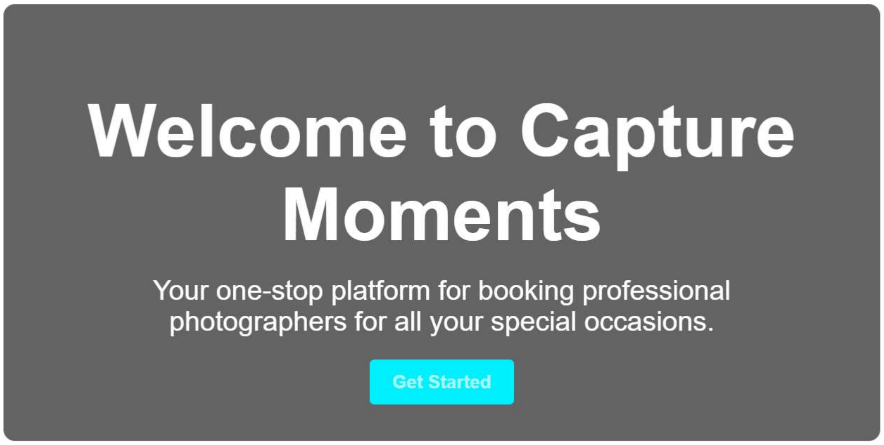
- APP.py Runs with the flask app .
- APP.py Manages the html codes to run properly .
- APP.py is the major role in hosting the website .



```python
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)
app.secret_key = 'your_secret_key'

# in-memory list to store bookings
all_bookings = []

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/index')
def index():
    return render_template('index.html')

```

FINAL OUTPUT OF THE CAPTURE MOMENTS WEBSITE

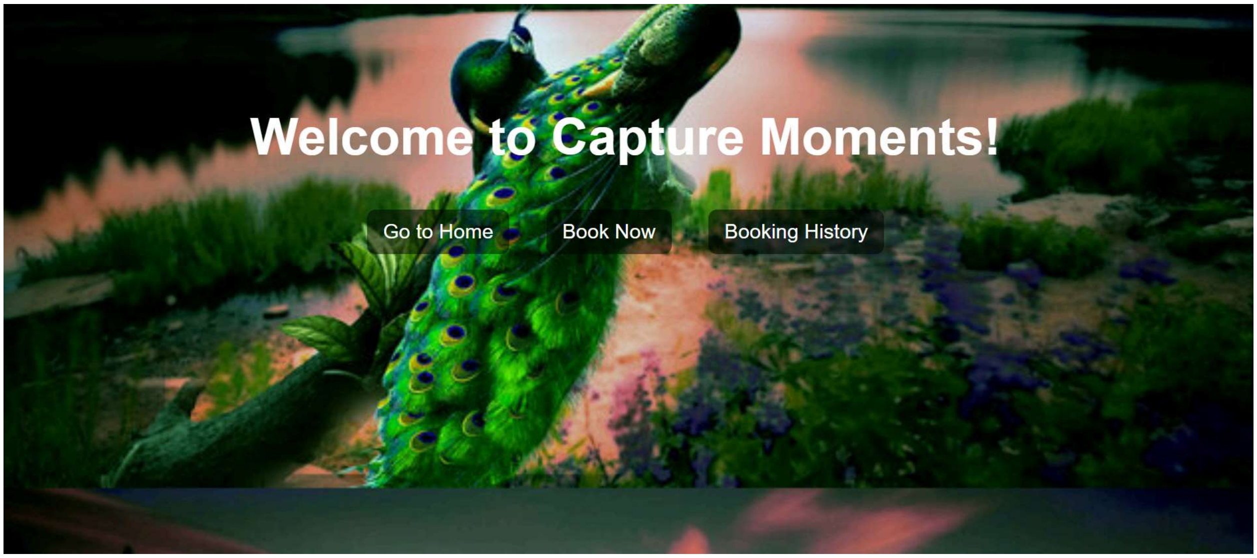
- HOME PAGE



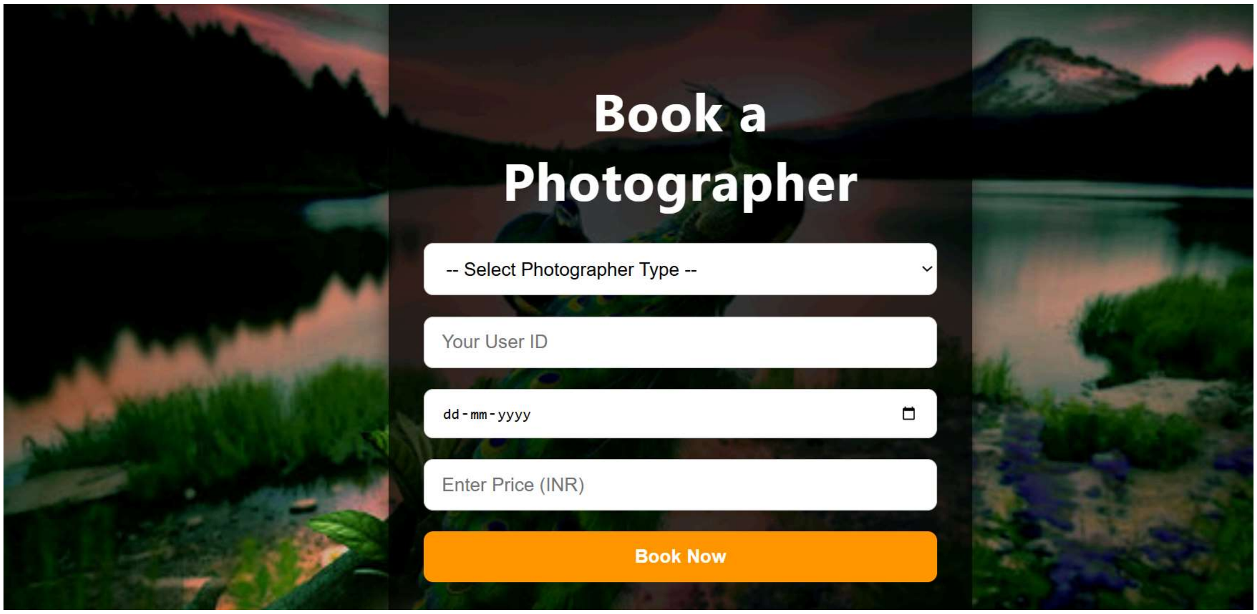
Welcome to Capture Moments

Your one-stop platform for booking professional photographers for all your special occasions.

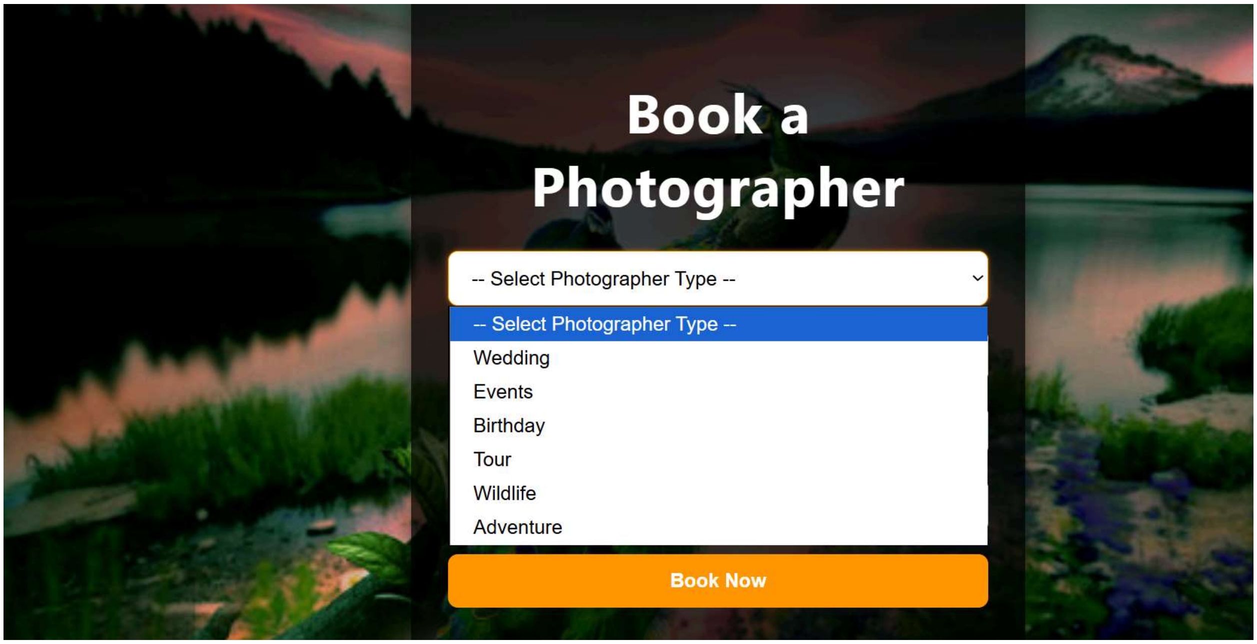
- **INDEX PAGE**

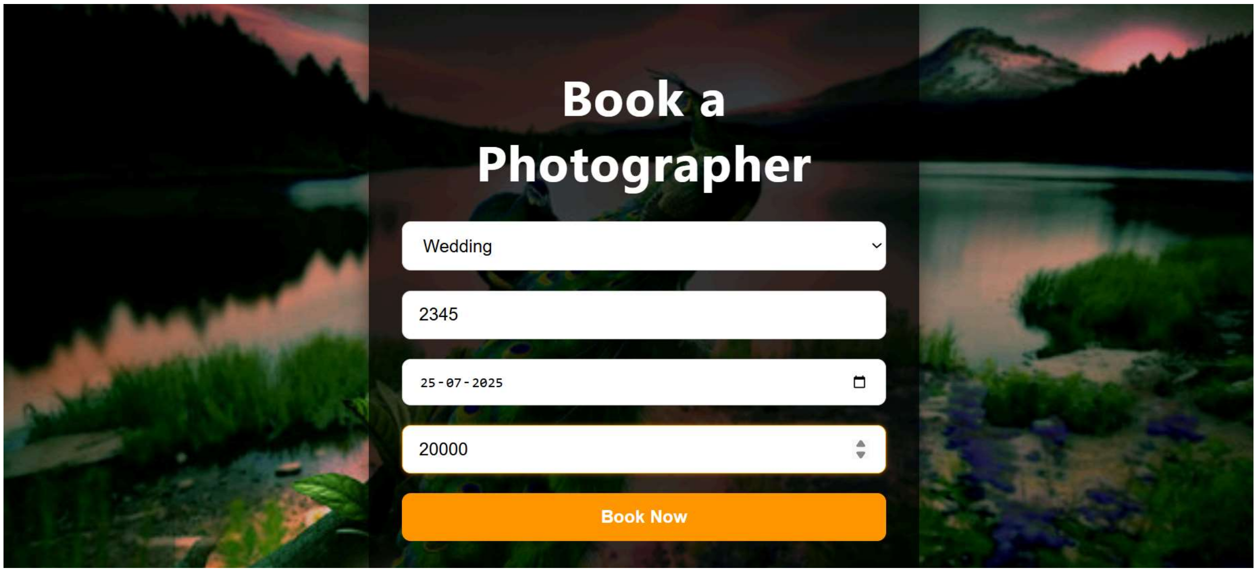


- **BOOKING PHOTOGRAPHERS PAGE**



Process of booking phographers




Book a Photographer

Wedding

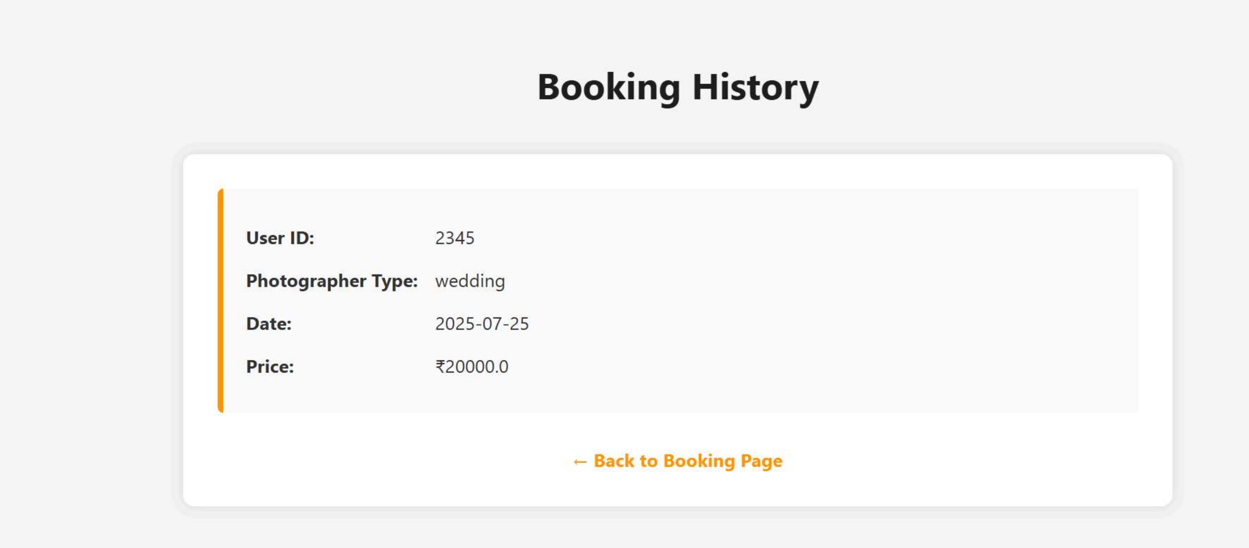
2345

25-07-2025

20000

- ☆ **<u>BOOKING SUCCESSFUL PAGE</u>**

<u>BOOKING HISTORY</u>



Booking History

User ID:	2345
Photographer Type:	wedding
Date:	2025-07-25
Price:	₹20000.0

CONCLUSION:

capture moments booking system is aprocess booking photographers to the events like birthday,wedding,event etc. the scheduling the photographers to the particular date.