



Presented By:- Branch:- Ai&ml

Team no:A1

Team Members: Poojitha Gogisetty (Team Lead)

P. Nasreen

K. Krishna Teja

M. Mohan Sai Ram

The Virtual Career Counselor: Harnessing Generative AI and AWS for Personalized Pathways

Project Description:

The Virtual Career Counselor is a web-based application designed to empower individuals by providing personalized career advice, course recommendations, and insights into job market trends. Utilizing Flask for backend development and integrating the Groq API for generative AI-driven recommendations, the application allows users to register, log in, and explore tailored services that help navigate their career paths effectively.

To ensure secure data management, the system employs AWS DynamoDB for efficient data storage and retrieval, alongside AWS Identity and Access Management (IAM) for enforcing secure access control. This solution enhances user experience by offering real-time notifications and personalized insights, enabling users to develop essential skills, explore suitable courses, and gain valuable information on in-demand skills and salary expectations in the evolving job market.

Scenario 1: Career Path Exploration

Users can input a career name to receive a comprehensive overview of the skills needed, recommended courses, and potential job roles associated with that career path. This feature empowers users to make informed decisions about their professional futures.

Scenario 2: Personalized Course Recommendations

The Virtual Career Counselor leverages Groq's AI capabilities to analyze user preferences and generate tailored course recommendations, helping users discover suitable educational pathways. When a user submits their preferences, the system quickly provides a list of recommended courses with descriptions, enhancing their learning journey.

Scenario 3: Job Market Insights

The platform offers real-time insights into job market trends related to specific careers, including in-demand skills, salary trends, and job availability by region. By accessing this data, users gain valuable knowledge to navigate their career choices effectively.

Pre-requisites:

1. AWS Account Setup: [AWS Account Setup](#)
2. Groq API Understanding: [Groq API Documentation](#)
3. Understanding IAM: [IAM Overview](#)
4. Amazon EC2 Basics: [EC2 Tutorial](#)
5. DynamoDB Basics: [DynamoDB Introduction](#)
6. Git Version Control: [Git Documentation](#)

Project Flow:

1. AWS Account Setup and Login
 - Activity 1.1: Set up an AWS account if not already done.
 - Activity 1.2: Log in to the AWS Management Console
2. DynamoDB Database Creation and Setup
 - Activity 2.1: Create a DynamoDB table named usertable to store user data.
 - Activity 2.2: Define and configure attributes in usertable.
3. Backend Development and Application Setup
 - Activity 3.1: Develop the Flask backend to handle core application functionalities
 - Activity 3.2: Configure API integration with the Groq API to support Generative AI-powered responses.
4. Generative AI Model Integration
 - Activity 4.1: Configure Groq API endpoints within the Flask application for three core functionalities:
 - Course Recommendations: Generate a list of personalized courses based on user preferences.
 - Career Path Generation: Generate detailed career paths
 - Job Market Trends: Retrieve up-to-date insights
5. IAM Role Setup
 - Activity 5.1: Create IAM roles to control access to AWS resources, ensuring secure connections to DynamoDB and Groq API.
 - Activity 5.2: Attach appropriate policies for resource access, ensuring least privilege access for enhanced security.
6. Application Routes and Core Functionalities Development
 - Activity 6.1: Establish routes in the Flask app to serve the following functionalities:

- Home (/): Display the landing page
- Registration (/register): Implement user registration
- Login (/login): Authenticate users and initiate sessions for personalized services.
- Logout (/logout): End user sessions and ensure data protection upon logout.
- Counsel (/counsel): Display an interactive page
- Generate Recommendations (/generate_recommendations): Use Groq API to generate a list of recommended courses based on user preferences.
- Career Path Generation (/generate_career_path): Create detailed career paths based on specific career interests.
- Job Market Trends (/job_market_trends): Provide real-time job market insights to help users make informed career decisions.

7. EC2 Instance Setup and Deployment

- Activity 7.1: Launch an EC2 instance to host the Flask application, with security configurations for HTTP and SSH access.
- Activity 7.2: Upload Flask application files, templates, and configuration files to the EC2 instance.
- Activity 7.3: Run the Flask app on EC2 and ensure it's accessible to users online.

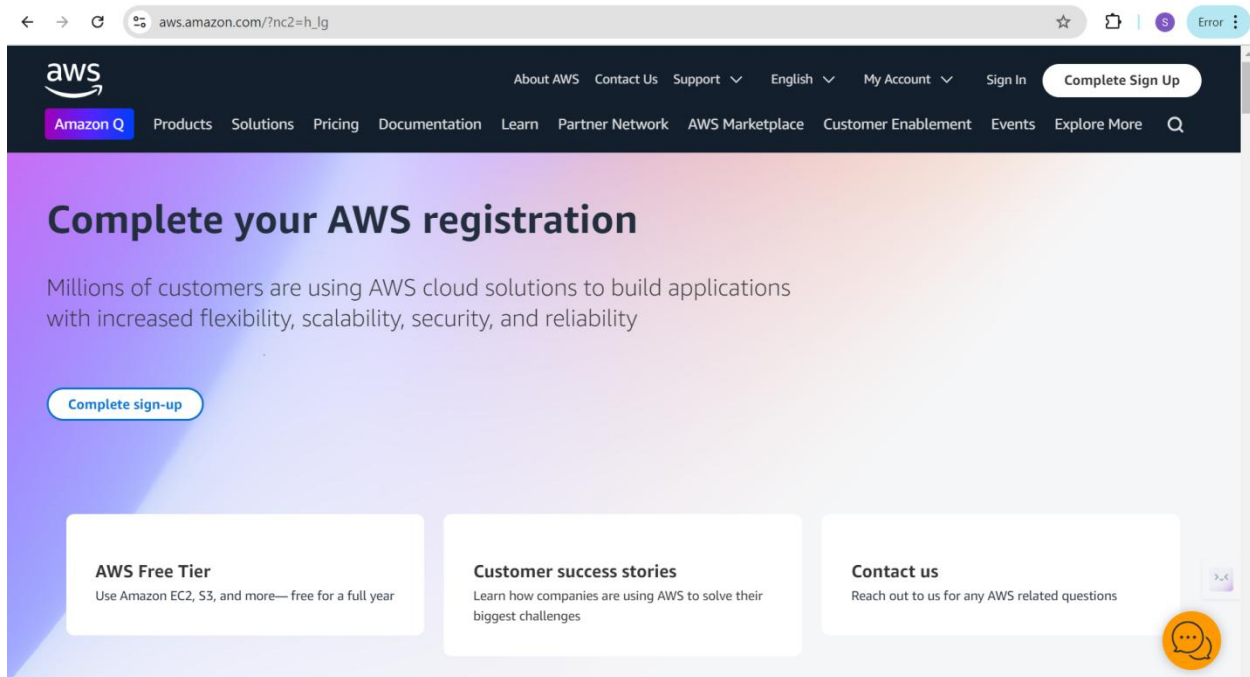
8. Testing and Deployment:



AWS Account Setup and Login

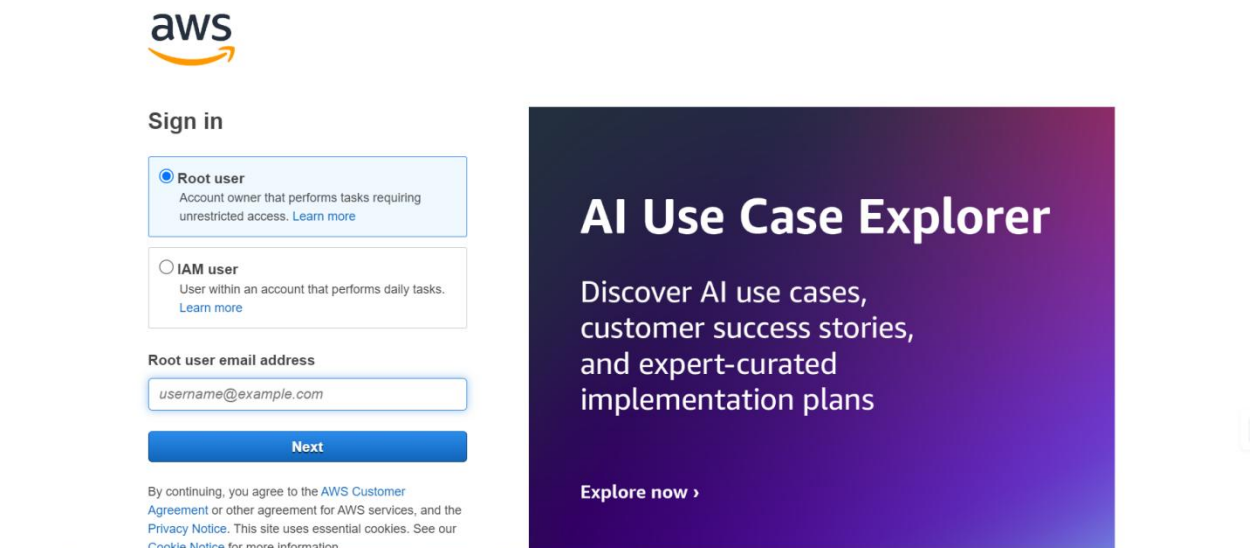
Set up an AWS account if not already done.

- Sign up for an AWS account and configure billing settings.



Log in to the AWS Management Console

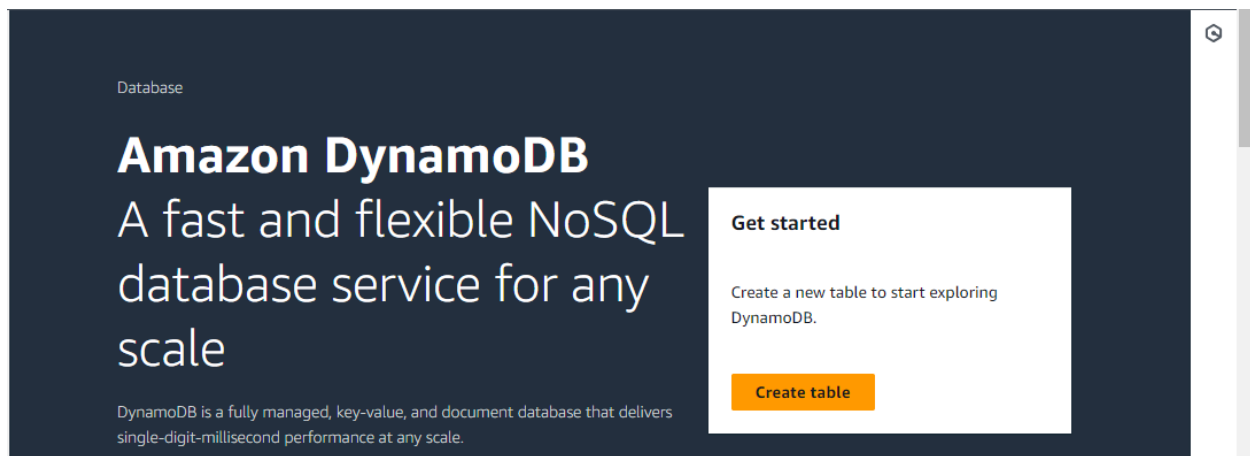
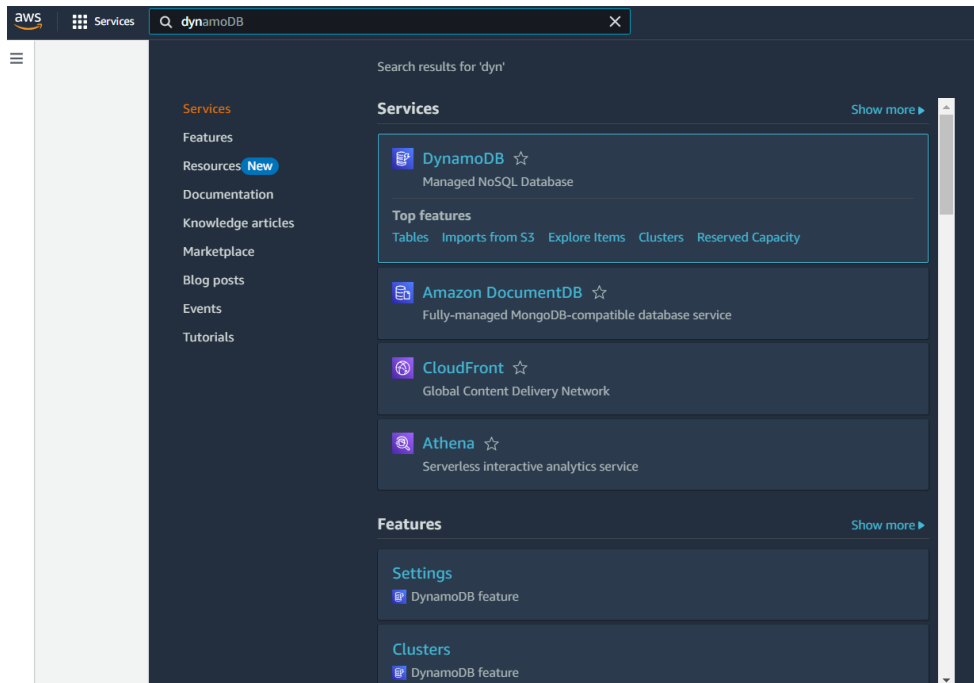
- After setting up your account, log in to the [AWS Management Console](#).



DynamoDB Database Creation and Setup

Navigate to the DynamoDB

- In the AWS Console, navigate to DynamoDB and click on create tables.



Create a DynamoDB table for storing registration details and book requests.

- Create UsersTable with partition key “Email” with type String.

[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.

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 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.

Table settings

☒ Default settings

The fastest way to create your table. You can modify these settings now or after your table has been created.

☐ Customize settings

Use these advanced features to make DynamoDB work better for your needs.

Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value	Editable after creation
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

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.

Cancel
Create table

Backend Development and Application Setup:

Develop the Flask backend to handle core application functionalities, including user registration, login, and session management.

Description of the code :

- Flask App Initialization

```

app.py > login
1  from flask import Flask, render_template, request, redirect, url_for, session
2  from groq import Groq # type: ignore
3  from config import usertable # Import the usertable from config
4  import bcrypt

5
6  app = Flask(__name__)
7  app.secret_key = '482cc9641e82a0e3#####b1b4a88ad'
8

```

Configure API integration

- Configure API integration with the Groq API to support Generative AI-powered responses. This includes setting up API calls to generate career paths, course recommendations, and job market trends tailored to individual user profiles.

```

11 # Directly include your Groq API key here
12 GROQ_API_KEY = "gsk_ub9VpJbEND4#####RkZEbgGI5y"
13 client = Groq(api_key=GROQ_API_KEY)

```

Generative AI Model Integration:

Configure Groq API endpoints within the Flask application for three core functionalities:

Course Recommendations: Generate a list of personalized courses based on user preferences. This includes specifying prompts to the model for high relevance.

```
22 # Function to generate course recommendations using Groq API
23 def generate_course_recommendation(user_preferences):
24     recommendations = client.chat.completions.create(
25         messages=[
26             {
27                 "role": "system",
28                 "content": f"""
29                 Based on the following user preferences, recommend 5 suitable courses:
30                 User preferences: {user_preferences}
31                 Provide the output as a list of course titles and brief descriptions.
32                 """
33             },
34         ],
35         model="llama3-8b-8192",
36     )
37     return recommendations.choices[0].message.content.strip().split('\n')
```

Career Path Generation: Generate detailed career paths that highlight required skills, recommended courses, and potential job roles based on user-selected fields.

```
39 # Function to generate career path using Groq API
40 def generate_career_path(career_name):
41     career_path = client.chat.completions.create(
42         messages=[
43             {
44                 "role": "system",
45                 "content": f'''
46                 Generate a career path for the career: {career_name}. Include the following details:
47                 Key Skills: List the essential skills required for this career path.
48                 Recommended Courses: Suggest courses that will help develop the necessary skills and knowledge.
49                 Potential Job Roles: Provide a list of potential job roles that someone following this career path might pursue.
50                 '''
51             },
52         ],
53         model="llama3-8b-8192",
54     )
55     return career_path.choices[0].message.content
```

Job Market Trends: Retrieve up-to-date insights on in-demand skills, salary ranges, top hiring companies, and job availability by region to inform users about industry trends.

```

55 # Function to generate job market trends using Groq API
56 def generate_job_market_trends(career_name):
57     job_market_trends = client.chat.completions.create(
58         messages=[{
59             "role": "system",
60             "content": f'''
61                 Provide real-time job market trends for the career: {career_name}. Include the following details:
62                 - In-Demand Skills: List the skills that are currently in high demand for this career.
63                 - Salary Trends: Provide the current salary ranges and trends for various roles in this career path.
64                 - Top Hiring Companies: Identify the top companies hiring for roles in this field.
65                 - Job Availability by Region: Give an overview of job availability in different geographic regions, highlighting where op
66             ...
67         }],
68         model="llama3-8b-8192",
69     )
70     return job_market_trends.choices[0].message.content

```

Config.py:

```

config.py > ...
1 class Config:
2     AWS_ACCESS_KEY_ID = 'AKIA2#####OKDF4C'
3     AWS_SECRET_ACCESS_KEY = '1u+GQa#####/uXAS5zOZD'
4     AWS_REGION_NAME = 'ap-south-1'
5     SECRET_KEY = '24194554c#####61b6d87eb4de5'
6

```

Description: The Config class contains configuration settings for the Virtual Career Counsellor application, including AWS access credentials and the secret key for session management. These settings enable secure access to AWS services like DynamoDB, IAM and EC2 while ensuring the application's overall security.

Secret.py:

```

secret.py > ...
1 import os
2 import binascii
3
4 # Generate a random secret key
5 secret_key = binascii.hexlify(os.urandom(24)).decode()
6 print(secret_key)
7

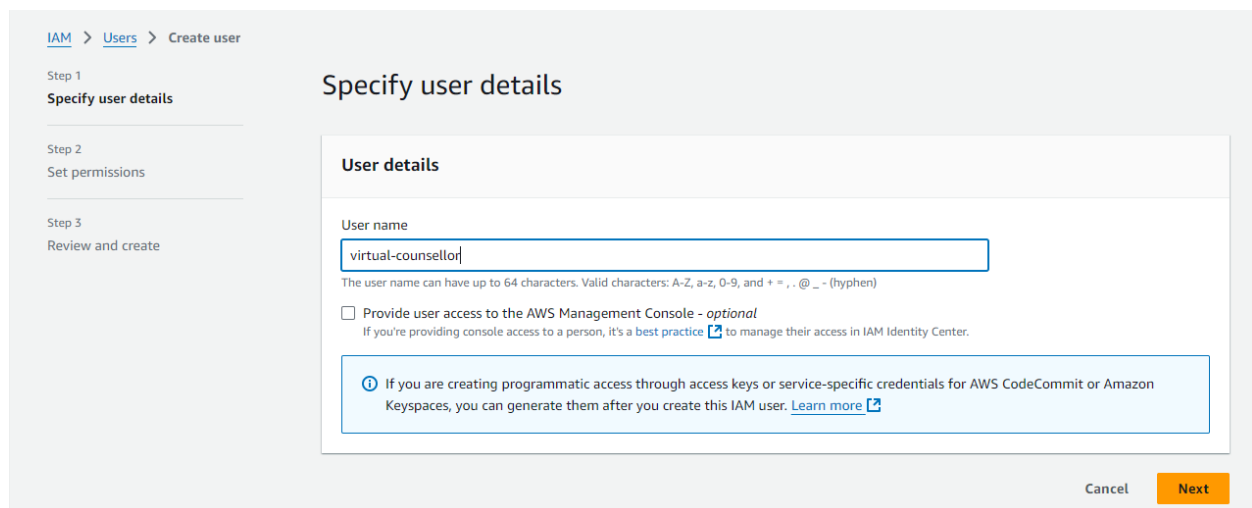
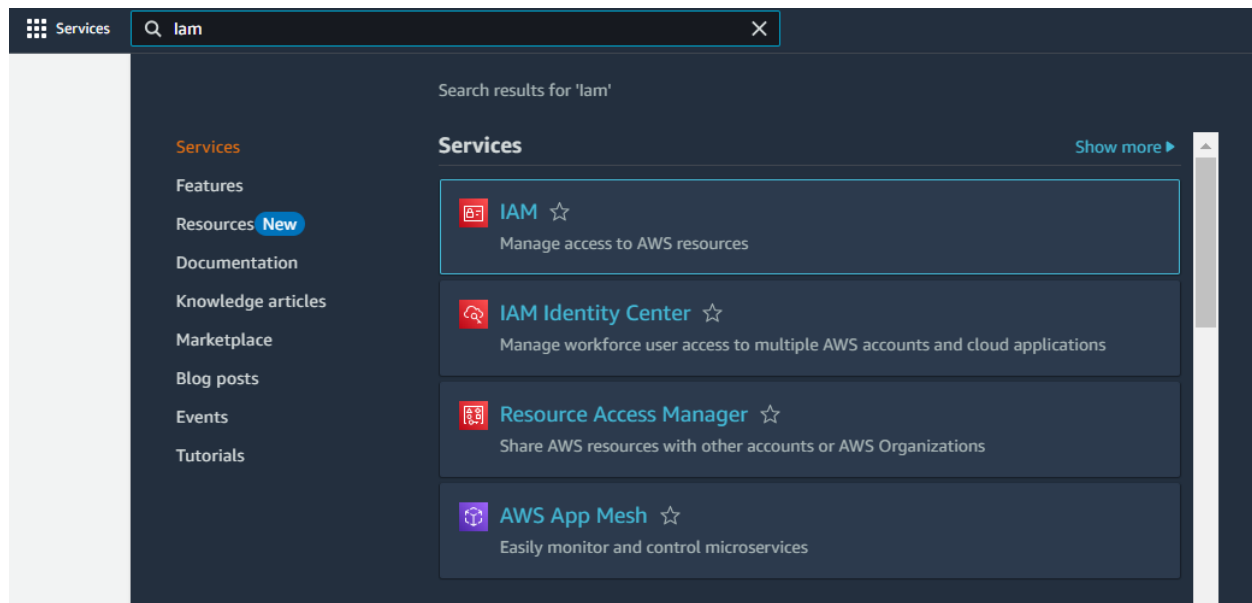
```

Description: This code generates a random secret key for use in applications by creating a random sequence of 24 bytes using `os.urandom(24)`. The bytes are then converted to a hexadecimal string, ensuring a secure key suitable for cryptographic operations.

IAM Role Setup

Create IAM Role.

- In the AWS Console, go to IAM and create a new IAM Role for EC2 to interact with DynamoDB and SNS.



Step 2: Add permissions

[Edit](#)

Permissions policy summary

Policy name ?	Type	Attached as
AmazonDynamoDBFullAccess	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](#)

✓ User created successfully

[View user](#)


You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[IAM](#) > [Users](#)

Users (1) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.


[Delete](#)
[Create user](#)

< 1 > [Settings](#)

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age
<input type="checkbox"/>	virtual-counsellor	/	0	-	-	-

Attach Policies.

Attach the following policies to the role:

- AmazonDynamoDBFullAccess: Allows EC2 to perform read/write operations on DynamoDB.

[IAM](#) > [Users](#) > Create user

Step 1

[Specify user details](#)

Step 2

Set permissions

Step 3

Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1245)

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



[Create policy](#)

Filter by Type

All types

< 1 2 3 4 5 6 7 ... 63 >

Permissions policies (1/1245)

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



[Create policy](#)

Filter by Type



All types

9 matches

< 1 >



	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	AmazonDynamoDBFullAcc...	AWS managed	0
<input type="checkbox"/>	AmazonDynamoDBFullAcc...	AWS managed	0
<input type="checkbox"/>	AmazonDynamoDBReadO...	AWS managed	0
<input type="checkbox"/>	AWSApplicationAutoscalin...	AWS managed	0
<input type="checkbox"/>	AWSLambdaDynamoDBEx...	AWS managed	0
<input type="checkbox"/>	AWSLambdaInvocation-Dy...	AWS managed	0
<input type="checkbox"/>	DynamoDBCloudWatchCo...	AWS managed	0
<input type="checkbox"/>	DynamoDBKinesisReplicati...	AWS managed	0

[IAM](#) > [Users](#) > Create user

Step 1
[Specify user details](#)


Step 2
[Set permissions](#)

Step 3
Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		
User name virtual-counsellor	Console password type None	Require password reset No

Permissions summary		
Name 	Type	Used as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy

Application Routes and Core Functionalities Development

Establish routes in the Flask app to serve the following functionalities:

1. Home (/): Display the landing page with an overview of features and links to register or log in.

```

72 # Route for home page
73 @app.route('/')
74 def home():
75     return render_template('home.html')

```

Description: A route is defined for the home page (/) that renders the home.html template for all users, regardless of their authentication status. This route serves as the main landing page for the application.

2. Registration (/register): Implement user registration, securing passwords using bcrypt hashing and saving details to DynamoDB.

```

77 # Route for displaying the registration page
78 @app.route('/register', methods=['GET', 'POST'])
79 def register():
80     if request.method == 'POST':
81         email = request.form.get('email')
82         username = request.form.get('username')
83         gender = request.form.get('gender')
84         mobile_number = request.form.get('mobile_number')
85         password = request.form.get('password')
86
87         # Hash the password
88         hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
89
90         # Save user details to DynamoDB
91         usertable.put_item(
92             Item={
93                 'email': email,
94                 'username': username,
95                 'gender': gender,
96                 'mobile_number': mobile_number,
97                 'password': hashed_password.decode('utf-8') # Store the hashed password as a string
98             }
99         )
100
101         return redirect(url_for('home'))
102
103     return render_template('register.html') # Adjust with your registration form template
  
```

Description: The /register route displays the registration page and handles user registration through both GET and POST methods. With a GET request, it renders the register.html template for user input, while a POST request processes the form data, hashes the password with bcrypt, and saves user details to DynamoDB. After successful registration, the user is redirected to the home page.

1. Login (/login): Authenticate users and initiate sessions for personalized services.

```

105 # Route for displaying the login page
106 @app.route('/login', methods=['GET', 'POST'])
107 def login():
108     if request.method == 'POST':
109         email = request.form.get('email')
110         password = request.form.get('password')
111
112         # Fetch user details from DynamoDB
113         response = usertable.get_item(
114             Key={'email': email}
115         )
116
117         user = response.get('Item')
118         if user and bcrypt.checkpw(password.encode('utf-8'), user['password'].encode('utf-8')):
119             # Redirect to counsel page after successful login
120             return redirect(url_for('counsel'))
121         else:
122             # Handle login failure (e.g., redirect back with an error)
123             return redirect(url_for('home')) # Modify this to show an error message
124
125     return render_template('login.html') # Adjust with your login form template
  
```


Description: The /login route displays the login page with a form for user credentials. On form submission (POST), it retrieves user data from DynamoDB, verifies the password using bcrypt, and redirects authenticated users to the counsel page. If authentication fails, the user is redirected to the home page.

2. **Logout (/logout):** End user sessions and ensure data protection upon logout.

```
127 # Route for user logout
128 @app.route('/logout')
129 def logout():
130     session.clear() # Clear session data
131     return redirect(url_for('home'))
```

Description: The /logout route handles user logout by clearing session data to end the user session. Once logged out, the user is redirected to the home page.

3. **Counsel (/counsel):** Display an interactive page for generating AI-driven recommendations and insights.

```
133 # Make sure user is logged in before accessing the counsel page
134 @app.route('/counsel')
135 def counsel():
136     return render_template('counsel.html')
```

Description: The /counsel route displays the counsel.html page, intended for logged-in users only. It ensures that the user is authenticated before granting access to the page content.

4. **Generate Recommendations (/generate_recommendations):** Use Groq API to generate a list of recommended courses based on user preferences.

```
138 # Route to generate course recommendations
139 @app.route('/generate_recommendations', methods=['POST'])
140 def generate_recommendations_route():
141     user_preferences = request.form.get('preferences')
142     recommendations = generate_course_recommendation(user_preferences)
143     return render_template('show_recommendations.html', preferences=user_preferences, recommendations=recommendations)
144
```

Description: The /generate_recommendations route processes a POST request to generate course recommendations based on user preferences. It retrieves preferences from the form, calls the generate_course_recommendation function, and displays the recommended courses on the show_recommendations.html page.

5. **Career Path Generation (/generate_career_path):** Create detailed career paths based on specific career interests.

```
145 # Route to generate and show career path
146 @app.route('/generate_career_path', methods=['POST'])
147 def generate_career_path_route():
148     career_name = request.form.get('career_name')
149     career_path = generate_career_path(career_name)
150     return render_template('show_career_path.html', career_name=career_name, career_path=career_path)
```

Description: The /generate_career_path route handles a POST request to generate a career path based on a specified career name. It retrieves the career name from the form, generates the career path using the generate_career_path function, and displays the results on the show_career_path.html page.

6. **Job Market Trends (/job_market_trends):** Provide real-time job market insights to help users make informed career decisions.

```
152 # Route to fetch and display job market trends
153 @app.route('/job_market_trends', methods=['POST'])
154 def job_market_trends_route():
155     career_name = request.form.get('career_name')
156     trends = generate_job_market_trends(career_name)
157     return render_template('show_job_market_trends.html', career_name=career_name, trends=trends)
```

Description: The /job_market_trends route processes a POST request to fetch job market trends based on a specified career name. It retrieves the career name from the form, generates the trends using the generate_job_market_trends function, and displays the results on the show_job_market_trends.html page.

7. **Deployment Code:**

8.

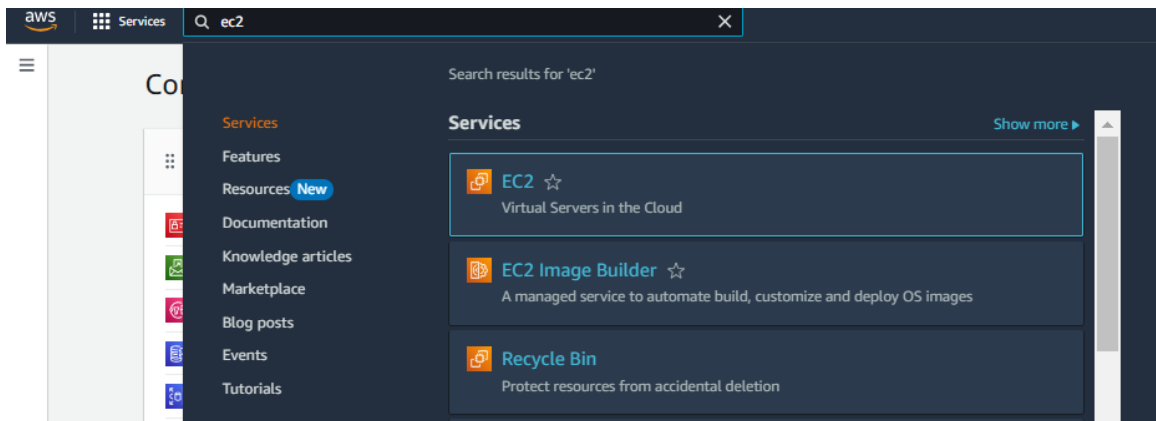
```
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
```

Description: This code snippet serves as the main entry point for the Flask application. When the script is executed directly, it starts the Flask development server in debug mode, allowing for live reloading and detailed error messages, which is useful for development and testing.

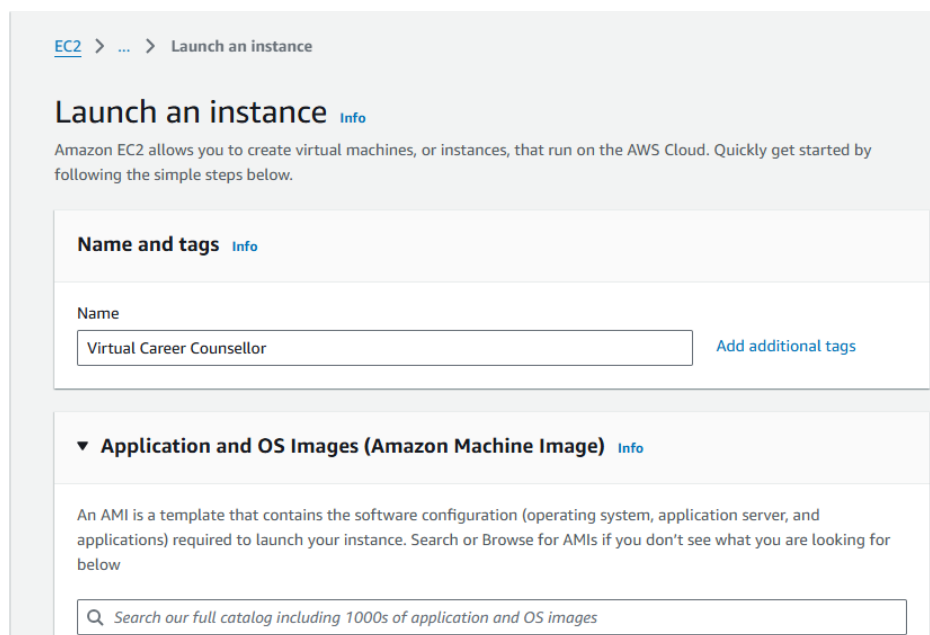
EC2 Instance Setup and Deployment:

Launch an EC2 instance to host the Flask application.

- Launch EC2 Instance
 - In the AWS Console, navigate to EC2 and launch a new instance.



- Click on Launch instance to launch EC2 instance



- Choose Amazon Linux 2 or Ubuntu as the AMI and t2.micro as the instance type (free-tier eligible).

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Li

SUS

Search

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Free tier eligible

ami-04a37924ffe27da53 (64-bit (x86), uefi-preferred) / ami-0846b753e2af0da6e (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-04a37924ffe27da53

Username

ec2-user

Verified provider

- Create and download the key pair for Server access.

▼ Instance type

Info | Get advice

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login)

Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

Create new key pair

Create key pair



Key pair name

Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type



RSA

RSA encrypted private and public key pair



ED25519

ED25519 encrypted private and public key pair

Private key file format



.pem

For use with OpenSSH



.ppk

For use with PuTTY



When prompted, store the private key in a secure and accessible location on


Cancel

Create key pair

Configure security groups for HTTP, and SSH access.


▼ Network settings
 [Info](#)

VPC - required
 [Info](#)


vpc-067f87b6f50f8036d
 (default) ▼
 

172.31.0.0/16

Subnet
 [Info](#)

subnet-0920393c2e8aed10b
 ▼
 

VPC: vpc-067f87b6f50f8036d Owner: 741448921864
 Availability Zone: ap-south-1a Zone type: Availability Zone
 IP addresses available: 4091 CIDR: 172.31.32.0/20

[Create new subnet](#)


Auto-assign public IP
 [Info](#)

Enable
 ▼

Additional charges apply when outside of free tier allowance

Firewall (security groups)
 [Info](#)


A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups
 [Info](#)

Select security groups
 ▼


[Compare security](#)

Instances (1)
 [Info](#)

Last updated less than a minute ago
 

Connect

Instance state ▼

Actions ▼

[Launch instances](#)


All states ▼

< 1 >
 

<input type="checkbox"/>	Name 	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability
<input type="checkbox"/>	Virtual Career ...	i-013b1ef4a51db2434	 Running  	t2.micro	 Initializing	View alarms 	ap-south-1

Setting up Inbound and Outbound rules

i-0cbd6b6ccb2afc1f4 (medtarck-ec2)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

▼ Security details

IAM Role

–

Owner ID

741448921864

Launch time

Wed Oct 23 2024 11:23:53 GMT+0530 (India Standard Time)

i-0f6f060a65171c78a (quiz-app)

▼ Inbound rules

Filter rules

< 1 >

Name	Security group rule ID	Port range	Protocol	Source
–	sgr-035f84939e1b2c81b	80	TCP	0.0.0.0/0
–	sgr-0a65c1024fe8d85fe	3306	TCP	0.0.0.0/0
–	sgr-0cde2a0a45ccc0352	22	TCP	0.0.0.0/0
–	sgr-0766684fdd95a7fce	443	TCP	0.0.0.0/0

- Add Type : HTTP > Source : Anywhere
- Add Type : HTTPS > Source : Anywhere

▼ Outbound rules

Filter rules

< 1 >

Name	Security group rule ID	Port range	Protocol	Destination
–	sgr-0aec66d8309fa06b1	All	All	0.0.0.0/0

Testing and Deployment

Deploy to EC2

1. Connect EC2 terminal.
2. Set up any necessary environment variables, including database connection strings.
3. Configure the web server to serve your application.
4. Start your application and ensure it's accessible via the EC2 instance's public IP or domain.
5. Run the below commands on ec2 terminal
6. `sudo yum update -y`
7. `sudo yum install python3 -y`
8. `sudo pip3 install virtualenv`
9. `python3 -m venv venv`
10. `source venv/bin/activate`
11. `pip install flask`
- Functional Testing
 - Test the `app.py` application for functionality, including database interactions and frontend features.
 - Run the Flask app `python3 app.py`
 - It will give you the link

Access the website through:

PublicIPs: <http://11.213.65.14:5000>

Deployment

- Deploy the application in a production environment, ensuring high availability and performance.

Click on the link above and it will take you to the webpage

Home Page:

Virtual Career Counselor

[Register](#)[Login](#)

Discover Your Perfect Career Path

Personalized guidance for your future, powered by Generative AI.

[Get Started](#)[Login](#)

How It Works

- 

Sign Up

Create your account to access personalized career guidance.
- 


Get Recommendations

Receive tailored career paths, courses, and job market trends.
- 


Skill Gap Analysis

Identify skills needed and resources to bridge gaps.


What We Offer

- 

Personalized Career Paths

Receive AI-driven career suggestions based on your skills and interests.
- 

Course Recommendations

Get tailored course recommendations to advance your career.
- 

Job Market Trends

Stay updated with the latest trends in the job market to make informed decisions.

Success Stories

- "The Virtual Career Counselor helped me find my dream job in tech!"*
— Sarah M.
- "Thanks to the course recommendations, I upskilled and got promoted!"*
— John D.
- "The job market insights were a game-changer for my career switch."*
— Emily R.

Description: The HTML template for the Virtual Career Counselor home page features a header with navigation links for registration and login, an engaging hero section to attract users, and detailed sections explaining the service offerings. It also includes testimonials to showcase user success stories and a footer with contact information and social media links for further engagement.

Register Page:

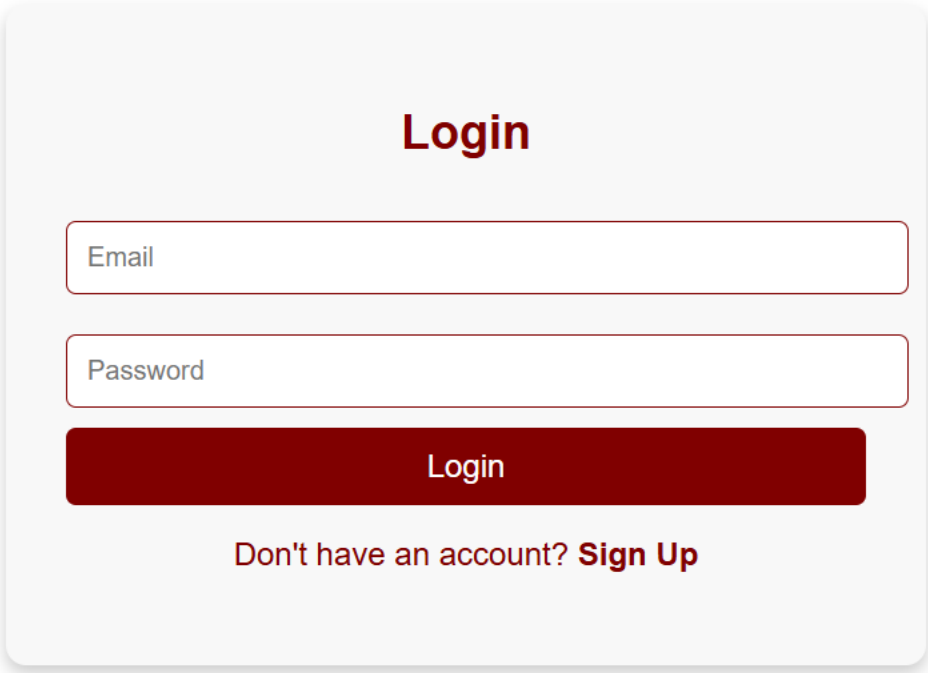
Create an Account

Register

Already have an account? [Login here](#)

Description: The registration page provides a user-friendly form for new users to create an account by entering their email, username, gender, mobile number, and password. It features a background image for aesthetics, a submit button for registration, and a link for existing users to log in..

Login Page:



The login form is a light gray rounded rectangle centered on the page. It contains the following elements:

- Login**: A large, bold, dark red heading centered at the top of the form.
- Email**: A white rectangular input field with a thin dark red border and the placeholder text "Email" in gray.
- Password**: A white rectangular input field with a thin dark red border and the placeholder text "Password" in gray.
- Login**: A solid dark red rectangular button with the word "Login" in white, centered below the password field.
- Don't have an account? Sign Up**: A line of text centered below the button, where "Don't have an account?" is in gray and "Sign Up" is in dark red.

Description: The login page features a straightforward form for users to enter their email and password to access their account. It includes a visually appealing background image, a submit button for logging in, and a link directing new users to the registration page.

Your Career Dashboard

Explore career options, get recommendations, and stay updated with job trends.



"Your career journey starts here. Take the next step!"

© 2025 Virtual Career Counselor. All rights reserved.

Counsel page:

Virtual Career Counselor

Enter a career name for a career path:

Get Career Path

Enter your preferences for course recommendations:

Get Course Recommendations

Enter a career name for job market trends:

Get Job Market Trends

© 2024 Virtual Career Counselor. All Rights Reserved.

Description: The Counsel page serves as an interactive hub for users to obtain personalized career insights and recommendations. Users can input specific career names or preferences to generate tailored career paths and course suggestions, while also accessing the latest job market trends. The user-friendly interface ensures a seamless experience, empowering individuals to make informed decisions about their professional journeys.

Career Path Page:

AI Engineer Career Path

AI Engineer Career Path:

AI Engineering is a rapidly growing field focused on building, training, and deploying artificial intelligence (AI) systems. It combines computer science, mathematics, and domain expertise to solve real-world problems.

Essential Skills:

- **Programming:** Python is the dominant language, followed by R, Java, and C++.
- **Machine Learning Algorithms:** Deep understanding of supervised, unsupervised, and reinforcement learning algorithms.
- **Data Structures & Algorithms:** Strong foundation in data structures and algorithms for efficient AI development.
- **Mathematics:** Linear algebra, calculus, probability, and statistics are crucial for understanding AI concepts.
- **Cloud Computing:** Experience with cloud platforms like AWS, Azure, or GCP for deploying and scaling AI models.
- **Big Data Technologies:** Familiarity with tools like Hadoop, Spark, or Kafka for handling large datasets.
- **Version Control:** Proficiency with Git for collaboration and managing code changes.
- **Communication & Problem-Solving:** Ability to clearly communicate complex technical concepts and solve challenging problems.

Recommended Courses:

- **Online Courses:**
 - **Coursera:** Machine Learning by Andrew Ng, Deep Learning Specialization
 - **edX:** Artificial Intelligence, Introduction to Computer Science and Programming Using Python
 - **Udacity:** Machine Learning Engineer Nanodegree, Deep Learning Nanodegree
- **University Programs:**
 - Master's in Computer Science with specialization in AI
 - Master's in Artificial Intelligence
 - PhD in Computer Science or related field

Potential Job Roles:

- **Machine Learning Engineer:** Develops and deploys machine learning models for various applications.
- **Data Scientist:** Analyzes large datasets to extract insights and build predictive models.
- **AI Research Scientist:** Conducts research on cutting-edge AI algorithms and techniques.
- **AI Product Manager:** Defines and drives the development of AI-powered products.
- **Computer Vision Engineer:** Develops AI systems for image and video analysis.
- **Natural Language Processing (NLP) Engineer:** Builds AI systems for understanding and generating human language.
- **Robotics Engineer:** Integrates AI into robotics systems for autonomous operation.
- **AI Consultant:** Advises organizations on implementing and utilizing AI solutions.

Career Progression:

1.

Entry-Level: Machine Learning Engineer Intern, Data Analyst

2.

Mid-Level: Machine Learning Engineer, Data Scientist

3.

Senior-Level: Senior Machine Learning Engineer, AI Research Scientist, AI Architect

4.

Leadership: AI Product Manager, Director of AI, Chief AI Officer. This career path is a guide and can be adapted based on individual interests and experiences. Continuous learning and staying updated with the latest advancements in AI are essential for success in this dynamic field. Remember that soft skills like communication, teamwork, and problem-solving are equally important alongside technical expertise.

[Back](#) [Logout](#)

Description: The Career Path page of the Virtual Career Counselor outlines essential steps and resources tailored for users pursuing a specific career, providing clarity and direction in their journey. It features easy navigation options to return home or log out, ensuring a seamless user experience.

Course Recommendation:

Virtual Career Counselor

Recommended Courses

Based on your preference: UPSC

Here are 5 suitable courses based on the user's interest in UPSC (Union Public Service Commission) exam:

****Course List****

- **UPSC Exam Preparation Crash Course****: This course covers all the essential topics and strategies for cracking the UPSC exam. It includes comprehensive study material, online sessions, and personalized mentoring.
- **Civil Services Masterclass****: This course provides in-depth coverage of all the subjects covered in the UPSC exam, along with tips and strategies for acing the exam. It includes interactive sessions, quizzes, and practice exercises.
- **UPSC Syllabus Revision Course****: Designed to help you refresh your knowledge of the UPSC syllabus, this course covers all the topics in detail. It includes video lectures, online study material, and practice tests.
- **UPSC Current Affairs and General Studies****: This course focuses on keeping you updated with the latest current affairs and general studies topics relevant to the UPSC exam. It includes daily newsletters, live sessions, and access to a library of past questions.
- **Mastering Mental Ability for UPSC****: This course is designed to help you develop your mental ability and logical thinking, which are essential skills for the UPSC exam. It includes cognitive training, reasoning exercises, and personality development modules.

These courses are designed to help you prepare for the UPSC exam and improve your chances of success.

[Go back](#) [Logout](#)

© 2024 Virtual Career Counselor. All rights reserved.
Contact: support@virtualcareer counselor.com

Description: The Course Recommendations page of the Virtual Career Counselor provides personalized course suggestions based on user preferences, displayed in an organized list format. Users can easily navigate back to the home page or log out, enhancing their experience while exploring their educational opportunities.

Job Market Trends:

Virtual Career Counselor

Job Market Trends for ias officer

• IAS Officer Career Market Trends:

• ****In-Demand Skills****

1. Data Analysis and Interpretation
2. Communication and Public Speaking
3. Policy Development and Implementation
4. Leadership and Strategic Planning
5. Crisis Management and Conflict Resolution
6. Budgeting and Financial Management
7. Digital Literacy and Tech Savviness
8. Emotional Intelligence and Empathy
9. Research and Analysis
10. Diplomacy and International Relations

• ****Salary Trends****

1. Junior IAS Officer: ₹8-12 lakhs per annum
2. Senior IAS Officer: ₹12-20 lakhs per annum
3. Deputy Secretary (IAS): ₹20-30 lakhs per annum
4. Secretary (IAS): ₹30-50 lakhs per annum
5. Additional Secretary (IAS): ₹50-70 lakhs per annum

• Note: These salary ranges are approximate and may vary based on factors like location, experience, and industry.

• ****Top Hiring Companies****

1. Government of India (Various Departments and Ministries)
2. State Governments (Various Departments and Ministries)
3. Public Sector Undertakings (PSUs)
4. Private Companies working with the Government (Various Industries)
5. International Organizations (United Nations, World Bank, etc.)

• ****Job Availability by Region****

1. ****Delhi and National Capital Region (NCR)**** High job availability in government departments, ministries, and PSUs.
2. ****Mumbai and Western Region**** Opportunities in private companies, industrial estates, and ports.
3. ****Southern Region (Kerala, Tamil Nadu, Karnataka)**** Job availability in government departments, IT sector, and education institutions.
4. ****Eastern Region (West Bengal, Odisha, Bihar)**** Opportunities in government departments, industries, and education institutions.
5. ****North Eastern Region (Assam, Meghalaya, etc.)**** Job availability in government departments, industries, and education institutions.

• ****Additional Insights****

- * There is a growing demand for IAS officers in the private sector, particularly in industries related to finance, human resources, and public policy.
- * Candidates with expertise in specific areas, such as agriculture, environment, or healthcare, are in high demand.
- * Women and candidates from diverse backgrounds are actively being recruited by the government and private sector.
- * With the increasing focus on digital transformation, IAS officers with technical skills and IT knowledge are highly sought after.

• Keep in mind that these trends and insights are subject to change and may vary based on various factors. It's essential to stay updated and adaptable to succeed in this career.

[Back](#)
[Logout](#)

© 2024 Virtual Career Counselor. All rights reserved.
Contact: support@virtualcareer counselor.com

Description: The Job Market Trends page of the Virtual Career Counselor presents insightful trends related to the user's chosen career path, displayed as a clear and organized list. Users can navigate back to the main page or log out, facilitating easy access to important career information.

Database:

Usertable from Dynamodb:

DynamoDB > Explore items > usertable

Tables (2)

Any tag key

Any tag value

Find tables

< 1 > ⚙

○ traveluser ☆

● **usertable** ☆

usertable

Autopreview View table details

► **Scan or query items**
Expand to query or scan items.

✓ Completed. Read capacity units consumed: 0.5

Items returned (3)

⌂ Actions Create item

< 1 > ⚙

<input type="checkbox"/>	email (String)	gender	mobile_num...	password	username
<input type="checkbox"/>	sunder.34m@gmail.com	male	null	\$2b\$12\$LS...	Sunder
<input type="checkbox"/>	shiva@gmail.com	male	null	\$2b\$12\$ph...	shiva
<input type="checkbox"/>	sirichakkala@gmail.com	female	null	\$2b\$12\$ro...	Siri

Conclusion:

The Virtual Career Counselor has been successfully developed as a cloud-based application, leveraging AWS services and Groq's Generative AI capabilities to provide users with personalized career guidance. Key AWS components like DynamoDB and IAM enable secure and scalable data storage and access management, ensuring a robust framework that can support a growing user base. With a Flask backend managing core functionalities, the application allows users to register, log in, and explore personalized career insights, enhancing their experience through easy navigation and secure data handling. Comprehensive testing confirmed seamless performance across features, including user authentication, data security, and API-based recommendations.

The integration with Groq's API powers the platform's core offering—providing tailored career paths, course recommendations, and job market trend analyses. This AI-driven approach enables the Virtual Career Counselor to deliver dynamic and relevant insights based on individual user profiles, helping users make informed decisions about their career development. Ultimately, the Virtual Career Counselor showcases how AI and cloud services can together drive innovation in personalized career guidance, making the path to career exploration more intuitive and accessible.