



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

Project Description:

The old way of career counseling with an increase in demand for personal guidance in a rapidly changing market tends to be less reliable and fruitful due to high costs and limited guidance. My project the “Virtual Career Counselor” address these career related doubts by harnessing Generative AI and use AWS for personalized Pathways. This platform allows students to signup, login and interact by generating career roadmaps and skill gap analysis based on their current skill set and profile. The project uses Flask for backend development, is hosted on AWS EC2 instance and integrated with Amazon DynamoDB for dynamic data management. The Groq Cloud AI API has been used for generating the guidance results regarding the doubts raised by the students related to their career. The AWS SNS sends email notifications from email address name “CareerAI” upon signup as a student or roadmap generation. This in turns increases as well as enhances user engagement with their support. This project not only aims to produce results as mere outputs but also explores various career streams ensuring good, qualitative and fast as well as intelligent access to professional growth for all.

Scenario 1: AI-Powered Career Roadmap Generation

The AWS EC2 provides a reliable infrastructure in the “Virtual Career Counselor” of handling complex AI queries from multiple users at the same time. For example, a student can log in, navigate to the "setup-goal" page, and input their academic background and professional interests. The system then calls the Groq Cloud AI API to analyze this data and generate a multiple step career roadmap—all in real-time. Flask manages the backend processes, ensuring that the AI’s response is formatted into a clear, actionable plan that helps the user understand exactly which skills or certifications they need next.

Scenario 2: Seamless Progress and Registration Notifications

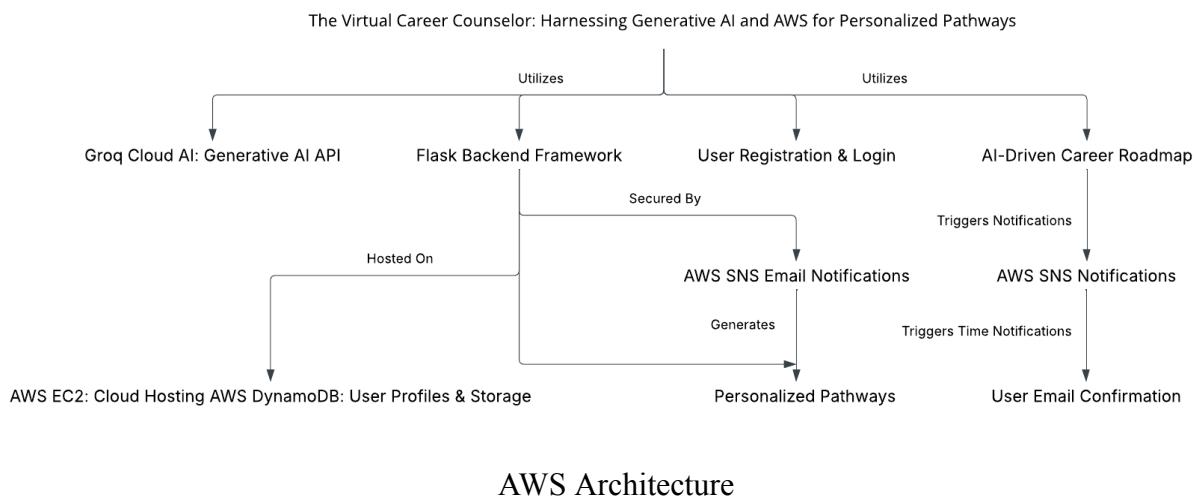
When a user registers or saves a new career path, the “Virtual Career Counselor” leverages AWS SNS to send instant email notifications from email address name “CareerAI” to confirm their progress. For instance, once a user completes their profile, Flask processes the data, and SNS sends a customized email with their login credentials or a summary of their newly generated career roadmap. This real-time notification system reduces uncertainty and keeps users motivated, while DynamoDB securely stores their profile and saved roadmaps, allowing them to pick up exactly where they left off during their next session.



SMARTBRIDGE

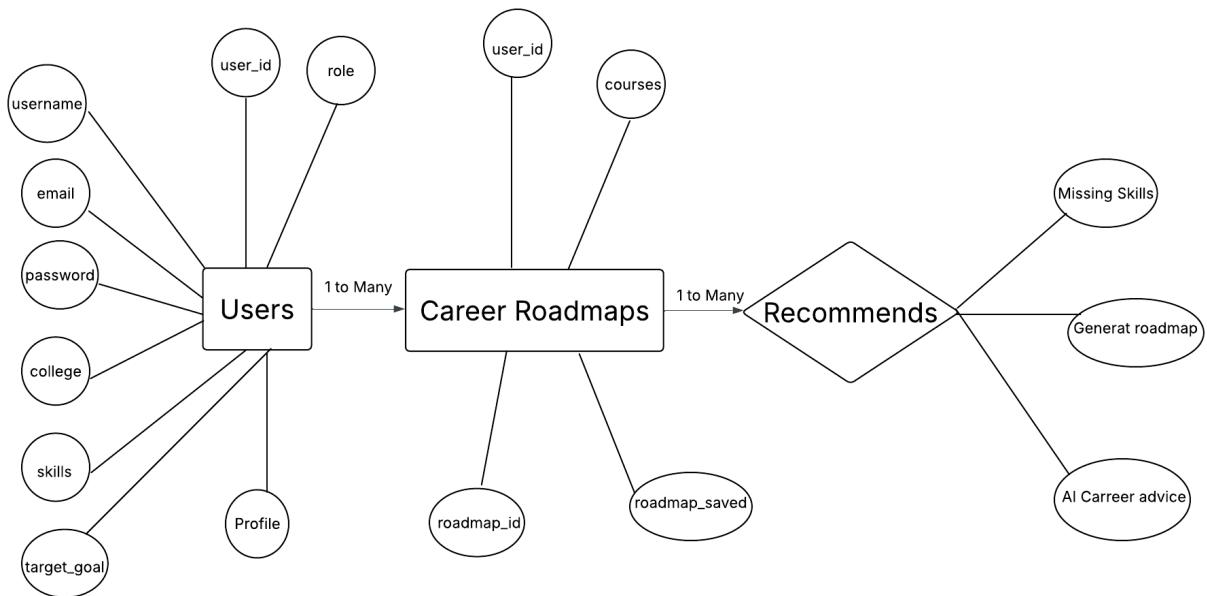
Scenario 3: Scalable Data Access and Profile Management

The “Virtual Career Counselor” offers users a seamless interface to manage their professional profiles and track their growth over time. After logging in, a user can view their saved career goals, skills, and previously generated AI roadmaps. Flask dynamically fetches this information from DynamoDB, ensuring that user data is updated instantly whenever they gain a new skill or change their career focus. Meanwhile, the EC2-hosted application remains stable and responsive even as more students join the platform, providing an uninterrupted and personalized experience that adapts to each individual's unique journey.





- Entity Relationship (ER) Diagram:



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](#)



SMARTBRIDGE

Technical Prerequisites:

1. AWS Infrastructure & Security

- AWS Account Access: An active AWS account to manage cloud resources.
- IAM Role Configuration: A custom IAM Role (e.g., custom_user_role) with AmazonDynamoDBFullAccess and AmazonSNSFullAccess policies attached to your EC2 instance.
- Key Pair Management: An RSA .pem key (e.g., career-key.pem) for secure SSH/SCP access to your server.

2. Development Environment

- Python & Package Manager: Python 3.x installed along with pip3 for managing dependencies.
- Core Libraries: Installation of flask (web framework), boto3 (AWS SDK for Python), and groq (AI API client).
- Code Editor: VS Code for local development and debugging before cloud deployment.

3. Database & Messaging Setup

- Amazon DynamoDB: A NoSQL table named Users created to store student profiles and credentials.
- Amazon SNS: A standard SNS Topic (e.g., CareerNotifications) with an active email subscription for automated alerts.

4. Generative AI Integration

- Groq Cloud API Key: A valid API key to access high-performance LLMs like Llama-3.3-70b-versatile for roadmap generation.



- **Project Workflow:**

- 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 userstable to store user data.
 - Activity 2.2: Define and configure attributes in userstable.

- 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.



SMARTBRIDGE

- 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



Milestone 1: AWS Account Setup and Login

- **Activity 1.1:** Set up an AWS account if not already done.
 - Sign up for an AWS account and configure billing settings.

The screenshot shows the AWS Sign Up page. At the top right, there is a language selection dropdown set to "English". The main heading is "Sign up for AWS". On the left, there is a promotional banner for trying AWS at no cost for up to 6 months, featuring illustrations of a server rack and a rocket launching from a stack of cubes. On the right, there are fields for "Root user email address" (containing "221030116@juitsolan.in") and "AWS account name" (containing "KS"). Below these are buttons for "Verify email address" and "Sign in to an existing AWS account". A note at the bottom states: "This site uses essential cookies. See our [Cookie Notice](#) for more information."

The screenshot shows the AWS Sign In page. The "User type" section has "Root user" selected. The "Email address" field contains "221030116@juitsolan.in". To the right of the sign-in form, there is a promotional banner for the "Builder Center" with the text "Find your community on Builder Center" and "Discover a network of Builders who get it. Share knowledge, solve problems, and grow together." There is a "Connect now" button and a "New to AWS? Sign up" link at the bottom.



SMARTBRIDGE

- Billing and Cost Management Setup:

The screenshot shows the AWS Billing and Cost Management home page. The left sidebar includes sections for Home, Getting Started, Dashboards, Billing and Payments (with Bills, Payments, Credits, Purchase Orders), Cost and Usage Analysis (with Cost Explorer, Cost Explorer Saved Reports, Cost Anomaly Detection, Free Tier, Data Exports, Customer Carbon Footprint Tool), and CloudShell, Feedback, and Console Mobile App links. The main content area displays a summary of costs: Month-to-date cost is \$0.02 (compared to last month for same period), Total forecasted cost for current month is Data unavailable, Last month's total cost is \$0.00 (Jan 1 – 10). It also features a Cost monitor section showing OK status, 1 active budget(s), and 1 monitor(s) active. A Cost breakdown section allows grouping costs by category, and a Recommended actions (2) section provides links for viewing bills and budgets.

- Activity 1.2: Log in to the AWS Management Console

➤ Log in to the AWS Management Console

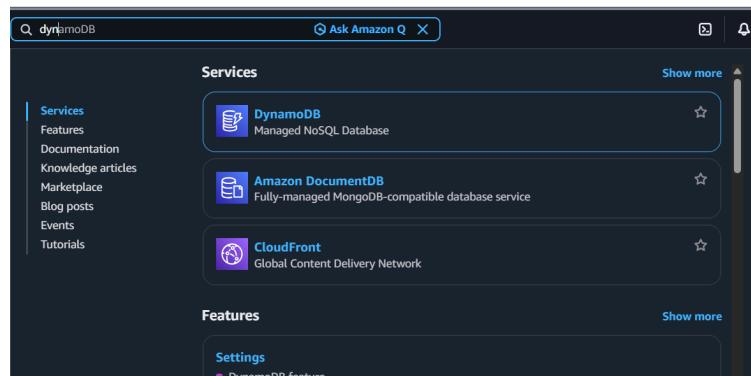
The screenshot shows the AWS Console Home page. The left sidebar lists Recently visited services: EC2, Simple Notification Service, DynamoDB, IAM, and Billing and Cost Management. Below this is a Welcome to AWS section with a Getting started with link. The main content area includes an Applications section (0 applications, Create application button, Region: US East (N. Virginia)), an AWS Health section (Open issues), and a Cost and usage section (Upgrade plan button, Credits cover your free plan costs). Navigation links at the bottom include View all services, Go to myApplications, and a footer with © 2026, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, and Cookie preferences.



Milestone 2: DynamoDB Database Creation and Setup

- **Activity 2.1:** Navigate to the DynamoDB

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



The screenshot shows the AWS DynamoDB Tables page. The left sidebar has a 'DynamoDB' section with 'Tables' selected, showing options like 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', 'Imports from S3', 'Integrations', 'Reserved capacity', and 'Settings'. Below that is a 'DAX' section with 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main content area displays a message: 'Share your feedback on Amazon DynamoDB. Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.' Below this is a table header for 'Tables (0) Info' with columns: Name, Status, Partition key, Sort key, Indexes, Replication Regions, Deletion protection, Favorite, and Read capacity. A search bar and filters for 'Find tables' and 'Filter by tag' are at the top of the table. A large 'Create table' button is prominently displayed. The bottom of the page includes links for CloudShell, Feedback, and Console Mobile App, along with copyright information: © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.



SMARTBRIDGE

- **Activity 2.2:** Create a DynamoDB table for storing signup details.

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' step is active. In the 'Table name' section, 'Users' is entered as the table name. Under 'Partition key', 'username' is defined as a String type. A note states: '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.' In the 'Sort key - optional' section, there is a placeholder 'Enter the sort key name' and a dropdown set to 'String'. A note states: '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.' At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

➤ Create “Users” table with partition key “username” with type String.

The screenshot shows the 'Tables' page in the AWS DynamoDB console. On the left, a sidebar lists 'DynamoDB' under 'Tables', with options like Dashboard, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Below that is a 'DAX' section with Clusters, Subnet groups, Parameter groups, and Events. The main area shows a table titled 'Tables (1) Info'. The table has columns: Name, Status, Partition key, Sort key, Indexes, Replication Regions, Deletion protection, Favorite, and Read capacity mode. One row is shown for the 'Users' table, which is currently 'Creating'. The status is 'Creating', the partition key is 'username (\$)', and the sort key is '-'. The table was last updated on February 8, 2026, at 19:12 (UTC+5:30). Notifications are set to 0. Actions include 'Actions', 'Delete', and 'Create table'. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.



The Users table was created successfully.

DynamoDB Tables (1) Info

	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode
<input type="checkbox"/>	Users	Active	username (\$)	-	0	0	Off		On-demand	On-demand

Last updated: February 8, 2026, 19:12 (UTC+5:30)

Actions Delete Create table

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Milestone 3: Backend Development and Application Setup:

- **Activity 3.1 :** Develop the Flask backend to handle core application functionalities, including user registration, login, and session management.
- Description of the code :
- Flask App Initialization

```

❶ app_aws.py
1  import boto3
2  from flask import Flask, render_template, request, redirect, url_for, session, flash, jsonify
3  from functools import wraps
4  from groq import Groq
5
6  app = Flask(__name__)
7  app.secret_key = "career_counselor_secret_key".
8
9
10 REGION_NAME = "us-east-1"
11
12 dynamodb = boto3.resource('dynamodb', region_name=REGION_NAME)
13 users_table = dynamodb.Table('Users')
14
15 |
16 sns_client = boto3.client('sns', region_name=REGION_NAME)
17 SNS_TOPIC_ARN = "arn:aws:sns:us-east-1:366256583005:CarrerNotifications"
18
19 GROQ_CLIENT = Groq(api_key="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")
20

```

```

❷ app_aws.py
91  @app.route('/login', methods=['GET', 'POST'])
92  def login():
93      if request.method == 'POST':
94          username = request.form.get('username')
95          password = request.form.get('password')
96          role_selection = request.form.get('role')
97
98          user = get_user_data(username)
99          if user and user.get('password') == password:
100              if user.get('role') == role_selection:
101                  session.update({
102                      'username': username,
103                      'is_admin': user.get('is_admin', False)
104                  })
105                  if user.get('is_admin'):
106                      return redirect(url_for('admin_dashboard'))
107                  return redirect(url_for('home'))
108              else:
109                  flash(f"This account is not registered as a {role_selection}.", "danger")
110          else:
111              flash("Invalid username or password.", "danger")
112      return render_template('login.html')
113

```



SMARTBRIDGE

```
app_aws.py
24  def login_required(f):
25      @wraps(f)
26      def decorated_function(*args, **kwargs):
27          if 'username' not in session:
28              return redirect(url_for('login'))
29          return f(*args, **kwargs)
30      return decorated_function
31
32  def admin_required(f):
33      @wraps(f)
34      def decorated_function(*args, **kwargs):
35          if not session.get('is_admin'):
36              flash("Unauthorized Access", "danger")
37              return redirect(url_for('login'))
38          return f(*args, **kwargs)
39      return decorated_function
40
41  def get_user_data(username):
42      try:
43          response = users_table.get_item(Key={'username': username})
44          return response.get('Item', {})
45      except Exception as e:
46          print(f"DynamoDB Error: {e}")
```

```
app_aws.py
67  @app.route('/signup', methods=['GET', 'POST'])
68  def signup():
69      if request.method == 'POST':
70          username = request.form.get('username')
71          password = request.form.get('password')
72          role = request.form.get('role')
73          try:
74              users_table.put_item(
75                  Item={
76                      'username': username,
77                      'password': password,
78                      'role': role,
79                      'is_admin': True if role == 'admin' else False
80                  },
81                  ConditionExpression='attribute_not_exists(username)'
82              )
83              send sns notification(username, f"Registered as {role}")
84              flash(f"{role.capitalize()} account created!", "success")
85              return redirect(url_for('login'))
86          except:
87              flash("Username already exists.", "danger")
88          return render_template('signup.html')
```

```
app_aws.py
231  @app.route('/logout')
232  def logout():
233      session.clear()
234      return redirect(url_for('login'))
235
236  if __name__ == '__main__':
237      app.run(debug=False, host='0.0.0.0', port=5000)
```



- **Activity 3.2:** Configure API integration with the Groq API to support Generative AI-powered responses.

- This includes setting up API calls to generate career roadmaps, course recommendations, and skill gaps analysis tailored to individual user profiles.



SMARTBRIDGE

```
183     @app.route('/generate-roadmap', methods=['POST'])
184     @login_required
185     def generate_roadmap():
186         username = session['username']
187         u = get_user_data(username)
188         prompt = (
189             f"Act as an elite Career Architect. Create a 4-phase roadmap for a student at {u.get('"
190             f"targeting a {u.get('target_goal')}) role. Current skills: {u.get('skills')}}. "
191             "Format your response as exactly 4 sections. Each section must start with 'PHASE X:' "
192             "followed by a short title and 2 bullet points for actions."
193         )
194
195         try:
196             completion = GROQ_CLIENT.chat.completions.create(
197                 model="llama-3.3-70b-versatile",
198                 messages=[{"role": "user", "content": prompt}]
199             )
200             roadmap_content = completion.choices[0].message.content
201             users_table.update_item(
202                 Key={'username': username},
203                 UpdateExpression="set roadmap_text = :r",
204                 ExpressionAttributeValues={':r': roadmap_content}
205             )
206             return jsonify({"success": True, "roadmap": roadmap_content})
207         except Exception as e:
208             return jsonify({"success": False, "error": str(e)}), 500
209
```

```
app_aws.py
210     @app.route('/api/chat', methods=['POST'])
211     @login_required
212     def chat():
213         user_query = request.json.get('message')
214         u = get_user_data(session['username'])
215         system_prompt = f"You are a counselor for a student at {u.get('college')}. Short answers only."
216         completion = GROQ_CLIENT.chat.completions.create(
217             model="llama-3.3-70b-versatile",
218             messages=[{"role": "system", "content": system_prompt}, {"role": "user", "content": user_query}]
219         )
220         return jsonify({"response": completion.choices[0].message.content})
221
```



Milestone 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.



SMARTBRIDGE

- **Career Roadmaps Generation:** Generate detailed career roadmaps.

```
183 @app.route('/generate-roadmap', methods=['POST'])
184 @login_required
185 def generate_roadmap():
186     username = session['username']
187     u = get_user_data(username)
188     prompt = (
189         f"Act as an elite Career Architect. Create a 4-phase roadmap for a student at {u.get('
190         f'targeting a {u.get('target_goal')}) role. Current skills: {u.get('skills')}). "
191         "Format your response as exactly 4 sections. Each section must start with 'PHASE X:' "
192         "followed by a short title and 2 bullet points for actions."
193     )
194
195     try:
196         completion = GROQ_CLIENT.chat.completions.create(
197             model="llama-3.3-70b-versatile",
198             messages=[{"role": "user", "content": prompt}]
199         )
200         roadmap_content = completion.choices[0].message.content
201         users_table.update_item(
202             Key={'username': username},
203             UpdateExpression="set roadmap_text = :r",
204             ExpressionAttributeValues={':r': roadmap_content}
205         )
206         return jsonify({"success": True, "roadmap": roadmap_content})
207     except Exception as e:
208         return jsonify({"success": False, "error": str(e)}), 500
209 
```

- **AI Chat Bot:** Retrieve up-to-date insights on in-demand skills, salary ranges, top hiring companies, and job availability by interacting with AI chatbot powered by GroqAI.

```
app_aws.py
1 app_aws.py
210 @app.route('/api/chat', methods=['POST'])
211 @login_required
212 def chat():
213     user_query = request.json.get('message')
214     u = get_user_data(session['username'])
215     system_prompt = f"You are a counselor for a student at {u.get('college')}. Short answers o
216     completion = GROQ_CLIENT.chat.completions.create(
217         model="llama-3.3-70b-versatile",
218         messages=[{"role": "system", "content": system_prompt}, {"role": "user", "content": us
219     )
220     return jsonify({"response": completion.choices[0].message.content})
221 
```

- **Configuration Description:** It 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 ,EC2 while ensuring application's overall security.

```
app_aws.py
1 import boto3
2 from flask import Flask, render_template, request, redirect, url_for, session, flash, jsonify
3 from functools import wraps
4 from groq import Groq
5
6 app = Flask(__name__)
7 app.secret_key = "career_counselor_secret_key".
8
9
10 REGION_NAME = "us-east-1"
11
12 dynamodb = boto3.resource('dynamodb', region_name=REGION_NAME)
13 users_table = dynamodb.Table('Users')
14
15
16 sns_client = boto3.client('sns', region_name=REGION_NAME)
17 SNS_TOPIC_ARN = "arn:aws:sns:us-east-1:366256583005:CareerNotifications"
18
19 GROQ_CLIENT = Groq(api_key="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")
```

- **Secret Description:** This code generates a random secret key for use in applications. The bytes are then converted to a hexadecimal string, ensuring a secure key suitable for cryptographic operations

```
6 app = Flask(__name__)
7 app.secret_key = "career_counselor_secret_key".
8
9
```



SMARTBRIDGE

Milestone 5: IAM Role Setup

- **Activity 5.1: Create IAM Role.**

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

The screenshot shows the AWS IAM service dashboard. On the left, there's a sidebar with links for DynamoDB, Tables, DAX, and other services. The main area displays the 'Services' section with three cards: 'IAM' (Manage access to AWS resources), 'IAM Identity Center' (Manage workforce user access to multiple AWS accounts and cloud applications), and 'Resource Access Manager' (Share AWS resources with other accounts or AWS Organizations). Below this is the 'Features' section with cards for 'IAM Access analyzer for S3' (S3 feature), 'Groups' (IAM feature), and 'Roles' (IAM feature). A feedback card at the bottom asks if the results were helpful, with 'Yes' and 'No' buttons. The top right corner shows the region as 'United States (N. Virginia)' and the account ID as 'KS (3662-5658-3005)'.

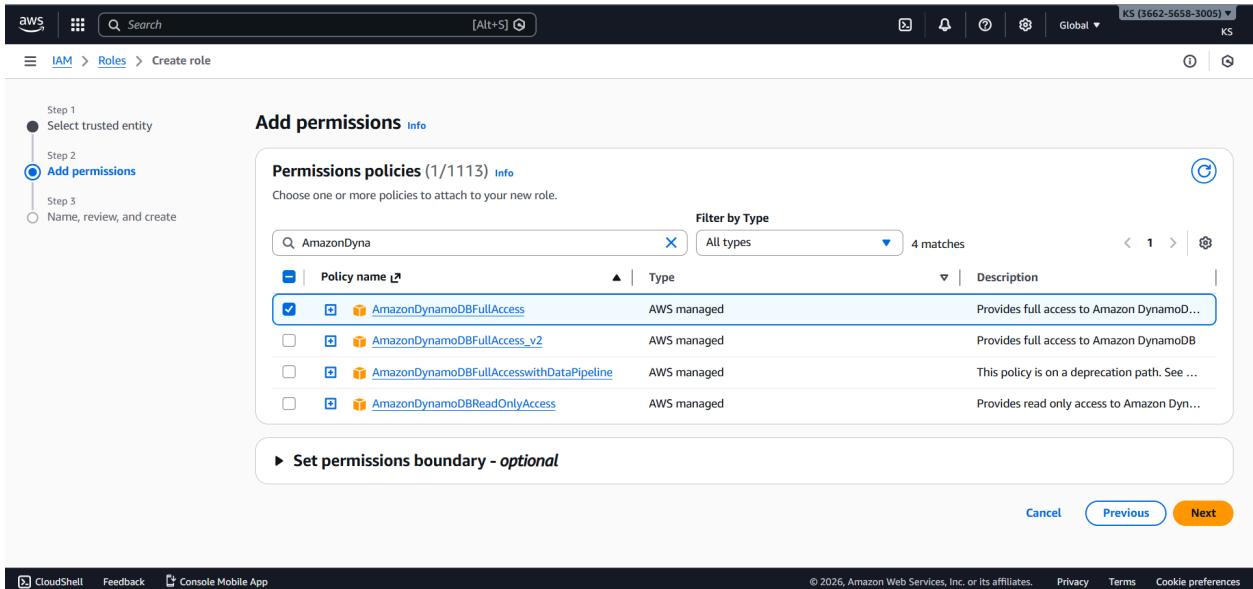
The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. It has three tabs: 'Select trusted entity' (selected), 'Add permissions', and 'Name, review, and create'. Under 'Trusted entity type', there are five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Under 'Use case', it says 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' and shows a dropdown for 'Service or use case' with 'EC2' selected. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.



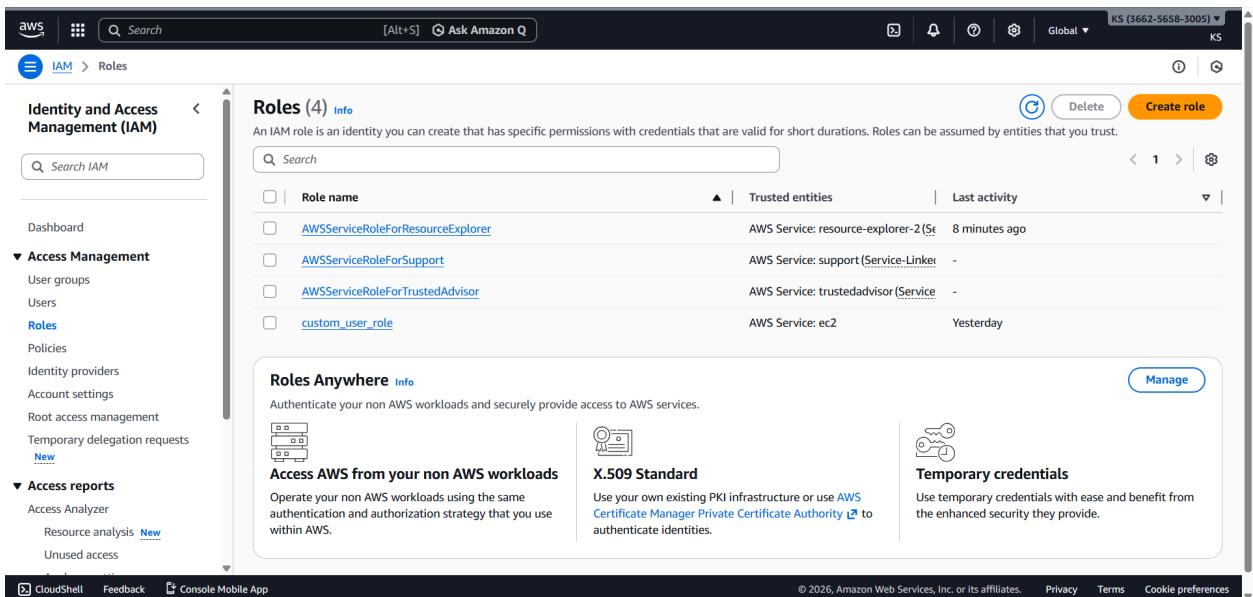
● Activity 5.2: Attach Policies.

➤ Attach the following policies to the role:

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



The screenshot shows the 'Add permissions' step of creating a new IAM role. The left sidebar indicates 'Step 1 Select trusted entity' is complete, 'Step 2 Add permissions' is selected, and 'Step 3 Name, review, and create' is pending. The main area is titled 'Add permissions' and shows a list of 'Permissions policies (1/1113)'. A search bar filters results for 'AmazonDyna'. One policy, 'AmazonDynamoDBFullAccess', is selected and highlighted. Other visible policies include 'AmazonDynamoDBFullAccess_v2', 'AmazonDynamoDBFullAccesswithDataPipeline', and 'AmazonDynamoDBReadOnlyAccess'. Below the list is a note about setting a 'permissions boundary - optional'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.



The screenshot shows the 'Roles' page in the AWS IAM console. The left sidebar under 'Access Management' lists 'User groups', 'Users', 'Roles' (which is expanded), 'Identity providers', 'Account settings', 'Root access management', and 'Temporary delegation requests'. The main area displays a table of roles with columns for 'Role name', 'Trusted entities', and 'Last activity'. Four roles are listed: 'AWSServiceRoleForResourceExplorer', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', and 'custom_user_role'. Below the table are sections for 'Roles Anywhere', 'Access AWS from your non AWS workloads' (using X.509 Standard), and 'Temporary credentials'.



SMARTBRIDGE

- **AmazonSNSFullAccess:** Grants EC2 the ability to send notifications via SNS.

The screenshot shows the 'Add permissions' step of creating a new IAM role. The left sidebar indicates three steps: Step 1 (Select trusted entity), Step 2 (Add permissions, which is selected and highlighted in blue), and Step 3 (Name, review, and create). The main area is titled 'Add permissions' and shows a search bar for 'Permissions policies'. A search term 'amazonsns' is entered, and the results are filtered by type ('All types') with 3 matches found. The 'AmazonSNSFullAccess' policy is selected, showing its details: 'AWS managed', 'Provides full access to Amazon SNS via t...', and a description. Below the table, a note says '▶ Set permissions boundary - optional'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted in orange.

The screenshot shows the 'Name, review, and create' step of creating a new IAM role. The left sidebar indicates three steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create, which is selected and highlighted in blue). The main area is titled 'Name, review, and create' and contains 'Role details'. Under 'Role name', the value 'custom_user_role' is entered. Under 'Description', the value 'Allows EC2 instances to call AWS services on your behalf.' is entered. Below the description, a note says 'Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=., @-[_{}\$%^&*]' is shown. At the bottom of the page, under 'Step 1: Select trusted entities', there is a 'Trust policy' section containing a JSON snippet of the trust policy. At the very bottom of the screen, there are links for CloudShell, Feedback, and Console Mobile App, along with standard footer links for Privacy, Terms, and Cookie preferences.



aws | Search [Alt+S] Global KS (3662-5658-3005) KS

IAM > Roles

Identity and Access Management (IAM)

Roles (4) Info

Role custom_user_role created.

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
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linker)	9 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
custom_user_role	AWS Service: ec2	-

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



SMARTBRIDGE

Milestone 6. Application Routes and Core Functionalities

Development

- **Activity 6.1:** Establish routes in the Flask app to serve the following

functionalities:

- **(/):** Display the landing page with an overview of features and links to signup or log in.

```
app_aws.py
63     @app.route('/')
64     def index():
65         return redirect(url_for('login'))
66
```

- **Home (/home):** Display the home page with an overview of features and links to signup or log in.

```
118     @app.route('/home')
119     @login_required
120     def home():
121         user = get_user_data(session['username'])
122         return render_template('home.html', user=user)
123
```

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.

- **Signup (/signup):** Implement user signup, securing passwords and saving details to DynamoDB.

```

67     @app.route('/signup', methods=['GET', 'POST'])
68     def signup():
69         if request.method == 'POST':
70             username = request.form.get('username')
71             password = request.form.get('password')
72             role = request.form.get('role')
73             try:
74                 users_table.put_item(
75                     Item=[{
76                         'username': username,
77                         'password': password,
78                         'role': role,
79                         'is_admin': True if role == 'admin' else False
80                     }],
81                     ConditionExpression='attribute_not_exists(username)'
82                 )
83                 # Notify admin of new registration
84                 send sns notification(username, f"Registered as {role}")
85                 flash(f"{role.capitalize()} account created!", "success")
86                 return redirect(url_for('login'))
87             except:
88                 flash("Username already exists.", "danger")
89             return render_template('signup.html')

```

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



SMARTBRIDGE

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

```
app_aws.py
91     @app.route('/login', methods=['GET', 'POST'])
92     def login():
93         if request.method == 'POST':
94             username = request.form.get('username')
95             password = request.form.get('password')
96             role_selection = request.form.get('role')
97
98             user = get_user_data(username)
99             if user and user.get('password') == password:
100                 if user.get('role') == role_selection:
101                     session.update({
102                         'username': username,
103                         'is_admin': user.get('is_admin', False)
104                     })
105                     if user.get('is_admin'):
106                         return redirect(url_for('admin_dashboard'))
107                         return redirect(url_for('home'))
108                     else:
109                         flash(f"This account is not registered as a {role_selection}.", "danger")
110                 else:
111                     flash("Invalid username or password.", "danger")
112             return render_template('login.html')
113
```

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

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

```
app_aws.py
231     @app.route('/logout')
232     def logout():
233         session.clear()
234         return redirect(url_for('login'))
235
```

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. **About (/about):** It contains information about the Career AI website..

```
app_aws.py
114 |     @app.route('/about')
115 |     def about():
116 |         return render_template('about.html')
117 |
```

4. **Setup-goal (/setup-goal):** It directs user to the setup career profile.

```
app_aws.py
124 |     @app.route('/setup-goal', methods=['GET', 'POST'])
125 |     @login_required
126 |     def setup_goal():
127 |         username = session['username']
128 |         if request.method == 'POST':
129 |             data = {
130 |                 'college': request.form.get('college'),
131 |                 'education': request.form.get('education'),
132 |                 'cgpa': request.form.get('cgpa'),
133 |                 'skills': request.form.get('skills'),
134 |                 'target_goal': request.form.get('target_goal')
135 |             }
136 |             users_table.update_item(
137 |                 Key={'username': username},
138 |                 UpdateExpression="set college=:c, education=:e, cgpa=:cg, skills=:s, target_goal=:t",
139 |                 ExpressionAttributeValues={
140 |                     ':c': data['college'], ':e': data['education'], ':cg': data['cgpa'],
141 |                     ':s': data['skills'], ':t': data['target_goal']
142 |                 }
143 |             )
144 |             send sns notification(username, f"Updated target to {data['target_goal']}"))
145 |             return redirect(url_for('dashboard'))
146 |         return render template('setup_goal.html', user=get user data(username))
```



SMARTBRIDGE

5. Dashboard (/dashboard): It directs you to Career roadmap page..

6. Generate-Roadmap (/generate-roadmap): It generates the career roadmap based upon your target goal..

```

183     @app.route('/generate-roadmap', methods=['POST'])
184     @login_required
185     def generate_roadmap():
186         username = session['username']
187         u = get_user_data(username)
188         prompt = (
189             f"Act as an elite Career Architect. Create a 4-phase roadmap for a student at {u.get('"
190             f"targeting a {u.get('target_goal')}) role. Current skills: {u.get('skills')}). "
191             "Format your response as exactly 4 sections. Each section must start with 'PHASE X:' "
192             "followed by a short title and 2 bullet points for actions."
193         )
194
195         try:
196             completion = GROQ_CLIENT.chat.completions.create(
197                 model="llama-3.3-70b-versatile",
198                 messages=[{"role": "user", "content": prompt}]
199             )
200             roadmap_content = completion.choices[0].message.content
201             users_table.update_item(
202                 Key={'username': username},
203                 UpdateExpression="set roadmap_text = :r",
204                 ExpressionAttributeValues={':r': roadmap_content}
205             )
206             return jsonify({"success": True, "roadmap": roadmap_content})
207         except Exception as e:
208             return jsonify({"success": False, "error": str(e)}), 500
209

```

7. Api-Chat(/api/chat): It helps you to chat with AI bot.

```

app_aws.py
210     @app.route('/api/chat', methods=['POST'])
211     @login_required
212     def chat():
213         user_query = request.json.get('message')
214         u = get_user_data(session['username'])
215         system_prompt = f"You are a counselor for a student at {u.get('college')}. Short answers only."
216         completion = GROQ_CLIENT.chat.completions.create(
217             model="llama-3.3-70b-versatile",
218             messages=[{"role": "system", "content": system_prompt}, {"role": "user", "content": user_query}]
219         )
220         return jsonify({"response": completion.choices[0].message.content})
221

```



SMARTBRIDGE

8. **Admin-Dashboard(/admin-dashboard)**: It contains details about no. of students enrolled with their username and email.

```
222 | @app.route('/admin-dashboard')
223 | @admin_required
224 | def admin_dashboard():
225 |     response = users_table.scan()
226 |     all_users = response.get('Items', [])
227 |     return render_template('admin_dashboard.html',
228 |                           all_users=all_users,
229 |                           user_count=len(all_users))
```

9. **Deployment Code:**

```
235
236 | if __name__ == '__main__':
237 |     app.run(debug=False, host='0.0.0.0', port=5000)
```

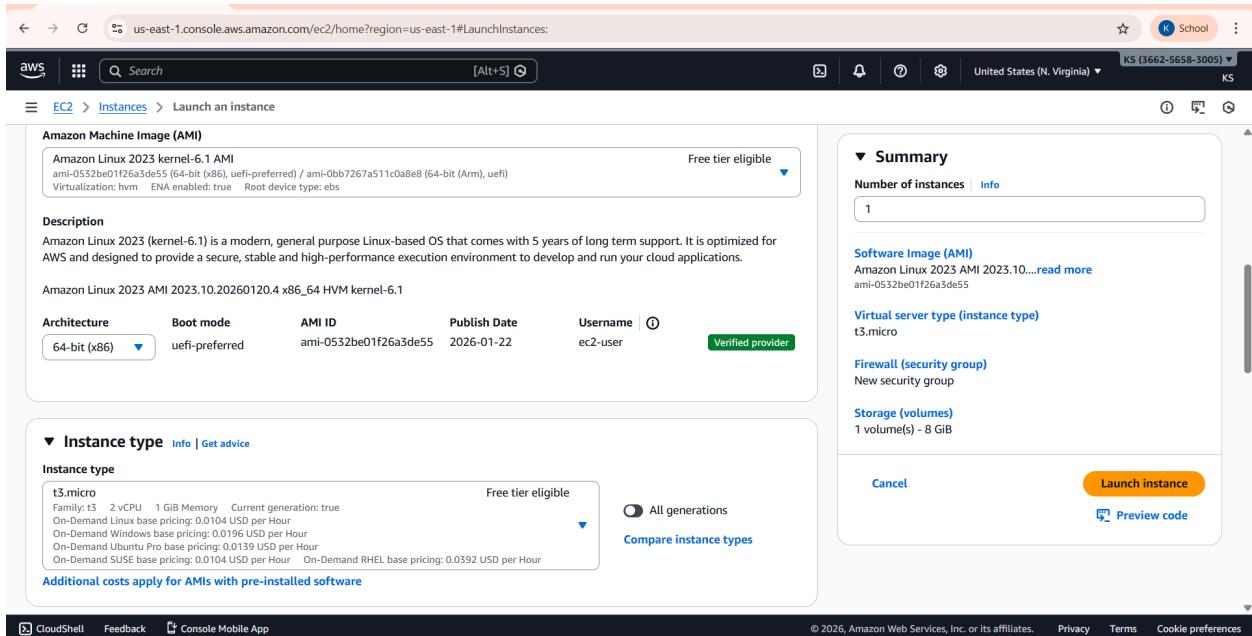
Description: This code snippet serves as the main entry point for the Flask application. When the script is executed directly, it starts the Flask development server in debug mode, allowing for live reloading and detailed error messages, which is useful for development and testing.

- **Milestone 7: EC2 Instance Setup and Deployment:**

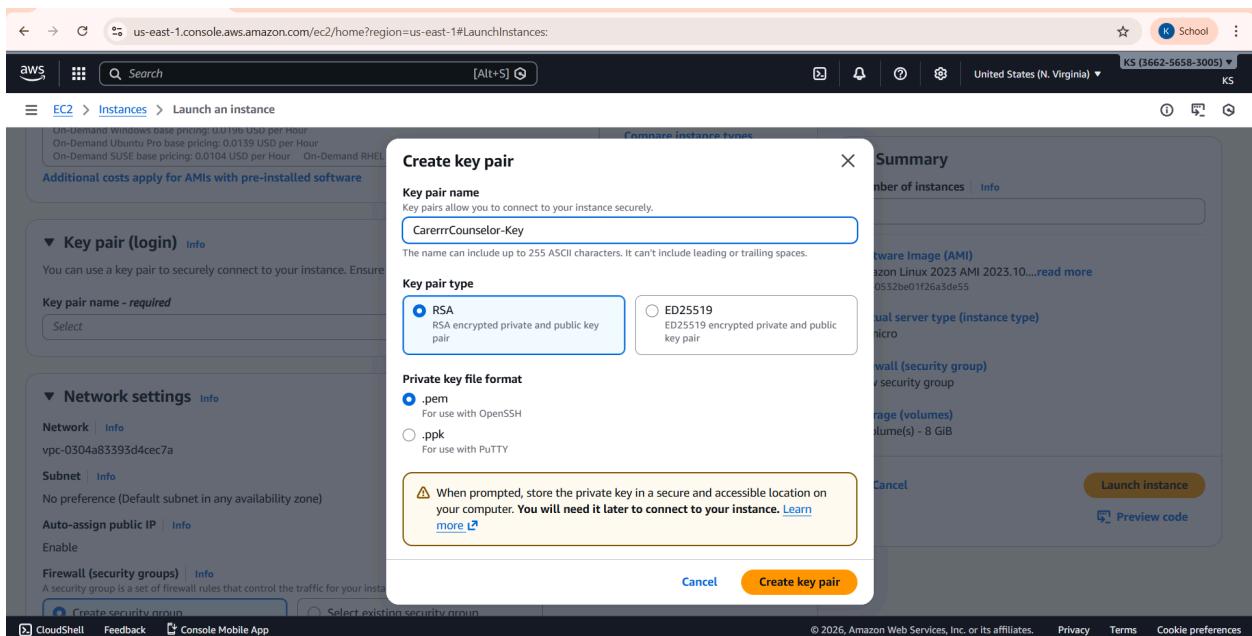
- **Activity 7.1: Launch EC2 Instance**

➤ In the AWS Console, navigate to EC2 and launch a new instance.

➤ Click on Launch instance to launch EC2 instance



The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Amazon Machine Image (AMI)' section, it lists 'Amazon Linux 2023 kernel-6.1 AMI' as the selected image. This image is 'Free tier eligible'. Below this, there's a 'Description' section stating that Amazon Linux 2023 is a modern, general purpose Linux-based OS. The 'Instance type' section shows a 't3.micro' instance selected, which has 2 vCPU, 1 GiB Memory, and is current generation. It is also 'Free tier eligible'. The 'Summary' panel on the right shows one instance being launched, with options for 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. At the bottom, there are 'Cancel', 'Launch instance', and 'Preview code' buttons.



The screenshot shows the 'Create key pair' step of the EC2 instance launch wizard. A modal window titled 'Create key pair' asks for a 'Key pair name', which is filled with 'Counselor-Key'. Below this, it says 'Key pairs allow you to connect to your instance securely.' Under 'Key pair type', 'RSA' is selected. It also lists 'ED25519' as an option. The 'Private key file format' section shows '.pem' selected, with a note that it's for use with OpenSSH. A warning message in the modal states: '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.' The background of the wizard shows the 'Summary' panel with one instance being launched, and the 'Launch instance' button is visible at the bottom right of the modal.



SMARTBRIDGE

- **Activity 7.2:** Configure security groups for HTTP, and SSH access.

The screenshot shows the AWS Network settings configuration page. It includes sections for Network, Subnet, Auto-assign public IP, Firewall (security groups), and Storage (volumes). The Firewall section shows a button to 'Create security group' (selected) and another to 'Select existing security group'. Below this, it says 'We'll create a new security group called "launch-wizard-1" with the following rules:' and lists 'Allow SSH traffic from Anywhere' (checked) and 'Allow HTTPS traffic from the internet' (unchecked). On the right side, there's a summary of the instance configuration: Software Image (AMI) - Amazon Linux 2023 AMI 2023.10..., Virtual server type (instance type) - t3.micro, Firewall (security group) - New security group, and Storage (volumes) - 1 volume(s) - 8 GiB. At the bottom are 'Cancel', 'Launch instance', and 'Preview code' buttons.

The screenshot shows the AWS Instances launch configuration page. It includes sections for IAM instance profile, Hostname type, DNS Hostname, Instance auto-recovery, Shutdown behavior, Stop - Hibernate behavior, and Termination protection. The IAM instance profile dropdown is set to 'custom_user_role arn:aws:iam::366256583005:instance-profile/custom_user_role'. On the right, there's a 'Create new IAM profile' button. The Summary section on the right shows 1 instance being launched, with details: Software Image (AMI) - Amazon Linux 2023 AMI 2023.10..., Virtual server type (instance type) - t3.micro, Firewall (security group) - New security group, and Storage (volumes) - 1 volume(s) - 8 GiB. At the bottom are 'Cancel', 'Launch instance', and 'Preview code' buttons.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Success
Successfully initiated launch of instance i-088638fd7729be1eb

▶ Launch log

Next Steps

Q What would you like to do next with this instance, for example "create alarm" or "create backup"

1 2 3 4 5 6 >

Create billing usage alerts To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Create billing alerts	Connect to your instance Once your instance is running, log into it from your local computer. 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. 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. Create EBS snapshot policy
Manage detailed monitoring CloudShell Feedback Console Mobile App	Create Load Balancer	Create AWS budget	Manage CloudWatch alarms

© 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-000d59a053300e8d7

☰ EC2 > Security Groups > sg-000d59a053300e8d7 - launch-wizard-1 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0f902bafada996df2	HTTPS	TCP	443	Custom	<input type="text" value="0.0.0.0/0"/> Delete
sgr-0e9aa578bf22803ec	HTTP	TCP	80	Custom	<input type="text" value="0.0.0.0/0"/> Delete
sgr-0341f925480063bc7	SSH	TCP	22	Anywh...	<input type="text" value="0.0.0.0/0"/> Delete
-	Custom TCP	TCP	5000	Anywh...	<input type="text" value="0.0.0.0/0"/> Delete

[Add rule](#)

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

[Cancel](#) [Preview changes](#) [Save rules](#)

[CloudShell](#) [Feedback](#) [Console Mobile App](#)



SMARTBRIDGE

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-088638fd7729be1eb

aws Search [Alt+S] United States (N. Virginia) KS (3662-5658-3005) KS

EC2 Instances i-088638fd7729be1eb Connect to instance

Connect Info

Connect to an instance using the browser-based client.

Instance ID i-088638fd7729be1eb (CareerCounselor-Server)	VPC ID vpc-0304a83393d4cec7a	Security groups sg-000d59a053300e8d7 (launch-wizard-1)	IAM role custom_user_role
---	---------------------------------	---	------------------------------

EC2 Instance Connect SSM Session Manager SSH client EC2 serial console

Instance ID
i-088638fd7729be1eb (CareerCounselor-Server)

Connection type

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
3.80.67.156

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.

ec2-user

Notes: You can use the default instance security group to connect to your instance. This security group allows traffic from the public internet to port 22 (SSH). You can also use the AWS CLI or AWS SDKs to connect to your instance using a private IP address and a VPC endpoint.

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:group-id=sg-000d59a053300e8d7

aws Search [Alt+S] United States (N. Virginia) KS (3662-5658-3005) KS

EC2 Security Groups sg-000d59a053300e8d7 - launch-wizard-1

Inbound security group rules successfully modified on security group (sg-000d59a053300e8d7 | launch-wizard-1)

Details

Security group name launch-wizard-1	Security group ID sg-000d59a053300e8d7	Description launch-wizard-1 created 2026-02-08T13:44:13.522Z	VPC ID vpc-0304a83393d4cec7a
Owner 366256583005	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Sharing VPC associations Related resources - new Tags

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0f902bafada996df2	IPv4	HTTPS	TCP	443
-	sgr-0e9aa578bf22803ec	IPv4	HTTP	TCP	80
-	sgr-0e11c2af7514e1275	IPv4	Custom TCP	TCP	5000

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



- SNS Setup:

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (with sub-options for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area displays a table titled 'Instances (1/1) Info'. It shows one instance: 'i-088638fd7729be1eb (CareerCounselor-Server)'. The instance is listed as 'Running' with an 't3.micro' instance type. A tooltip over the Public IPv4 address '3.80.67.156' indicates it has been copied. Other details shown include Private IPv4 addresses (172.31.18.168), Public DNS (ec2-3-80-67-156.compute-1.amazonaws.com), and Instance state (Running).

➤ Creating SNS Topics:

The screenshot shows the Amazon Simple Notification Service (SNS) landing page. At the top, there's a banner announcing 'New Feature: Amazon SNS now supports High Throughput FIFO topics. Learn more.' Below this, the page title is 'Amazon Simple Notification Service' with the subtitle 'Pub/sub messaging for microservices and serverless applications.' A detailed description follows, mentioning that Amazon SNS is highly available, durable, secure, and fully managed pub/sub messaging service that enables decoupling of microservices, distributed systems, and event-driven serverless applications. To the right, there's a 'Create topic' form with a 'Topic name' field containing 'CarrierNotifications' and a 'Next step' button. Below the form, a link says 'Start with an overview'. At the bottom, there's a 'Benefits and features' section and a 'Pricing' section stating that there are no upfront costs.



SMARTBRIDGE

Screenshot of the AWS SNS Create topic page.

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

FIFO (first-in, first-out)
• Strictly-preserved message ordering
• Exactly-once message delivery
• Subscription protocols: SQS

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

Name

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

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

Maximum 100 characters.

Encryption - optional
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

Access policy - optional [Info](#)
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS SNS Create topic page.

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

Tags - optional
A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 tags.

Active tracing - optional [Info](#)
Use AWS X-Ray active tracing for this topic to view its traces and service map in Amazon CloudWatch. Additional costs apply.

Use active tracing [Learn more](#)
AWS X-Ray active tracing helps you analyze and debug the end-to-end request paths of your application. When you activate active tracing on this topic, the Amazon SNS console verifies your AWS X-Ray resource policies. If your policies do not have the required permissions, the Amazon SNS console attempts to automatically create a policy that allows the Amazon SNS service to send data to AWS X-Ray. Additional costs apply, check AWS X-Ray pricing for more information.

Don't use active tracing

[Cancel](#) [Create topic](#)

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



➤ Creating SNS Subscriptions:

The screenshot shows the Amazon SNS Topics page. On the left, there's a navigation sidebar with links for Dashboard, Topics, Subscriptions, and Mobile (Push notifications, Text messaging (SMS)). The main area displays a green success message: "Topic CarrerNotifications created successfully. You can create subscriptions and send messages to them from this topic." Below this, the "CarrerNotifications" topic details are shown, including its Name (CarrerNotifications), ARN (arn:aws:sns:us-east-1:366256583005:CarrerNotifications), Display name (CareerAI), and Type (Standard). The topic owner is listed as 366256583005. At the bottom, there are tabs for Subscriptions, Access policy, Data protection policy, Delivery policy (HTTP/S), Delivery status logging, Encryption, and Tags. The Subscriptions tab is selected, showing a table with one row: "Subscriptions (0)". There are buttons for Edit, Delete, Request confirmation, Confirm subscription, and Create subscription. The "Create subscription" button is highlighted in orange. The footer includes links for CloudShell, Feedback, and Console Mobile App, along with standard AWS navigation icons.

The screenshot shows the "Create subscription" page. At the top, there's a blue banner with a "New Feature" message: "Amazon SNS now supports High Throughput FIFO topics. Learn more". The main form is titled "Create subscription" and has three sections: "Details", "Protocol", and "Endpoint". In the "Details" section, the "Topic ARN" field contains "arn:aws:sns:us-east-1:366256583005:CarrerNotifications". In the "Protocol" section, the "The type of endpoint to subscribe" dropdown is set to "Email". In the "Endpoint" section, the "An email address that can receive notifications from Amazon SNS." field contains "221030116@juitsolan.in". A note below the endpoint field says: "After your subscription is created, you must confirm it." In the bottom right corner, there's a section titled "Subscription filter policy - optional" with a note: "This policy filters the messages that a subscriber receives." The footer includes links for CloudShell, Feedback, and Console Mobile App, along with standard AWS navigation icons.



SMARTBRIDGE

The screenshot shows the AWS SNS console interface. In the top navigation bar, the path is: Amazon SNS > Topics > CarrerNotifications > Subscription: 50156da7-ea73-484b-882d-1c1db5889fd0. A green success message box at the top right says "Subscription to CarrerNotifications created successfully. The ARN of the subscription is arn:aws:sns:us-east-1:366256583005:CarrerNotifications:50156da7-ea73-484b-882d-1c1db5889fd0." Below this, the main content area is titled "Subscription: 50156da7-ea73-484b-882d-1c1db5889fd0". It displays the following details:

Details	Status
ARN arn:aws:sns:us-east-1:366256583005:CarrerNotifications:50156da7-ea73-484b-882d-1c1db5889fd0	Pending confirmation
Endpoint 221030116@juitsolan.in	Protocol EMAIL
Topic CarrerNotifications	
Subscription Principal arn:aws:iam::366256583005:root	

Below the details, there are two tabs: "Subscription filter policy" (which is selected) and "Redrive policy (dead-letter queue)". A note below the tabs says: "Subscription filter policy Info This policy filters the messages that a subscriber receives." At the bottom of the page, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information: © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

- **Milestone 8: Testing and Deployment**

- **Activity 8.1:** 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



SMARTBRIDGE

```
[ec2-user@ip-172-31-18-168 ~]$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository
=====
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:
Version 2023.10.20260202:
Run the following command to upgrade to 2023.10.20260202:
dnf upgrade --releasever=2023.10.20260202

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.10.20260202.html
=====
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-18-168 ~]$
```

i-088638fd7729be1eb (CareerCounselor-Server)

PublicIPs: 3.80.67.156 PrivateIPs: 172.31.18.168

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
Complete!
[ec2-user@ip-172-31-18-168 ~]$ sudo dnf install python3-pip -y
```

i-088638fd7729be1eb (CareerCounselor-Server)

PublicIPs: 3.80.67.156 PrivateIPs: 172.31.18.168

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 6°C Clear Search ENG IN 19:43 08-02-2026

aws | [Alt+5] United States (N. Virginia) KS (3662-5658-3005) KS

```
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-18-168 ~]$ sudo dnf install python3-pip -y
Last metadata expiration check: 0:00:31 ago on Sun Feb  8 14:13:06 2026.
Dependencies resolved.

Transaction Summary

Install 2 Packages

Total download size: 1.9 M
Installed size: 11 M
Downloading Packages:
(1/2): libcrypt-compat-4.4.33-7.amzn2023.x86_64.rpm 2.2 MB/s | 92 kB 00:00
(2/2): python3-pip-21.3.1-2.amzn2023.0.15.noarch.rpm 30 MB/s | 1.8 MB 00:00

Total
Running transaction check
Transaction check succeeded.
```



➤ Uploading zip file of AWS_Capstone_Project to cloudshell

A screenshot of the AWS CloudShell interface. The terminal window shows a login message for Amazon Linux 2023 and a timestamp of Sun Feb 8 14:09:21 2026 from 18.206.107.28. Below the terminal, a green notification bar indicates a successful file upload:

```
Last login: Sun Feb 8 14:09:21 2026 from 18.206.107.28
[ec2-user@ip-172-31-18-168 ~]$ ls
AWS_Capstone_Project.zip app_aws.py
[ec2-user@ip-172-31-18-168 ~]$
```

File upload successful
AWS_Capstone_Project.zip was successfully uploaded to the following directory:
/home/cloudshell-user.

A screenshot of the AWS CloudShell interface. The terminal window shows a login message for Amazon Linux 2023 and a timestamp of Sun Feb 8 14:37:14 2026 from 18.206.107.28. Below the terminal, the uploaded zip file is listed:

```
Last login: Sun Feb 8 14:37:14 2026 from 18.206.107.28
[ec2-user@ip-172-31-18-168 ~]$ ls
AWS_Capstone_Project.zip app_aws.py
[ec2-user@ip-172-31-18-168 ~]$
```



SMARTBRIDGE

- Unzipping AWS_Capstone_Project folder

```
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug/_pycache_/_init_.cpython-314.pyc
creating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/INSTALLER
creating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/licenses/
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/licenses/LICENSE.txt
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/METADATA
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/RECORD
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/REQUESTED
inflating: AWS_Capstone_Project/venv/lib/site-packages/werkzeug-3.1.5.dist-info/WHEEL
creating: AWS_Capstone_Project/venv/lib/site-packages/_pycache_/
inflating: AWS_Capstone_Project/venv/lib/site-packages/_pycache_/six.cpython-314.pyc
inflating: AWS_Capstone_Project/venv/pyenv.cfg
creating: AWS_Capstone_Project/venv/Scripts/
inflating: AWS_Capstone_Project/venv/Scripts/activate
inflating: AWS_Capstone_Project/venv/Scripts/activate.bat
inflating: AWS_Capstone_Project/venv/Scripts/activate.fish
inflating: AWS_Capstone_Project/venv/Scripts/Activate.ps1
inflating: AWS_Capstone_Project/venv/Scripts/deactivate.bat
inflating: AWS_Capstone_Project/venv/Scripts/flask.exe
inflating: AWS_Capstone_Project/venv/Scripts/jp.py
inflating: AWS_Capstone_Project/venv/Scripts/pip.exe
inflating: AWS_Capstone_Project/venv/Scripts/pip3.14.exe
inflating: AWS_Capstone_Project/venv/Scripts/pip3.exe
inflating: AWS_Capstone_Project/venv/Scripts/python.exe
inflating: AWS_Capstone_Project/venv/Scripts/pythonw.exe
creating: AWS_Capstone_Project/venv/Scripts/_pycache_/
inflating: AWS_Capstone_Project/venv/Scripts/_pycache_/jp.cpython-314.pyc
[ec2-user@ip-172-31-18-168 ~]$
```

i-088638fd7729be1eb (CareerCounselor-Server)

PublicIPs: 3.80.67.156 PrivateIPs: 172.31.18.168

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

● 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:

```
[ec2-user@ip-172-31-18-168 ~]$ cd AWS_Capstone_Project
[ec2-user@ip-172-31-18-168 AWS_Capstone_Project]$ python3 app_aws.py
/home/ec2-user/.local/lib/python3.9/site-packages/boto3/compat.py:89: PythonDeprecationWarning: Boto3 will no longer support Python 3.9 starting April 29, 2026. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.10 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
* Serving Flask app 'app_aws'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.1:5000)
* Running on http://127.0.0.1:5000
* Running on http://172.31.18.168:5000
Press CTRL+C to quit
```

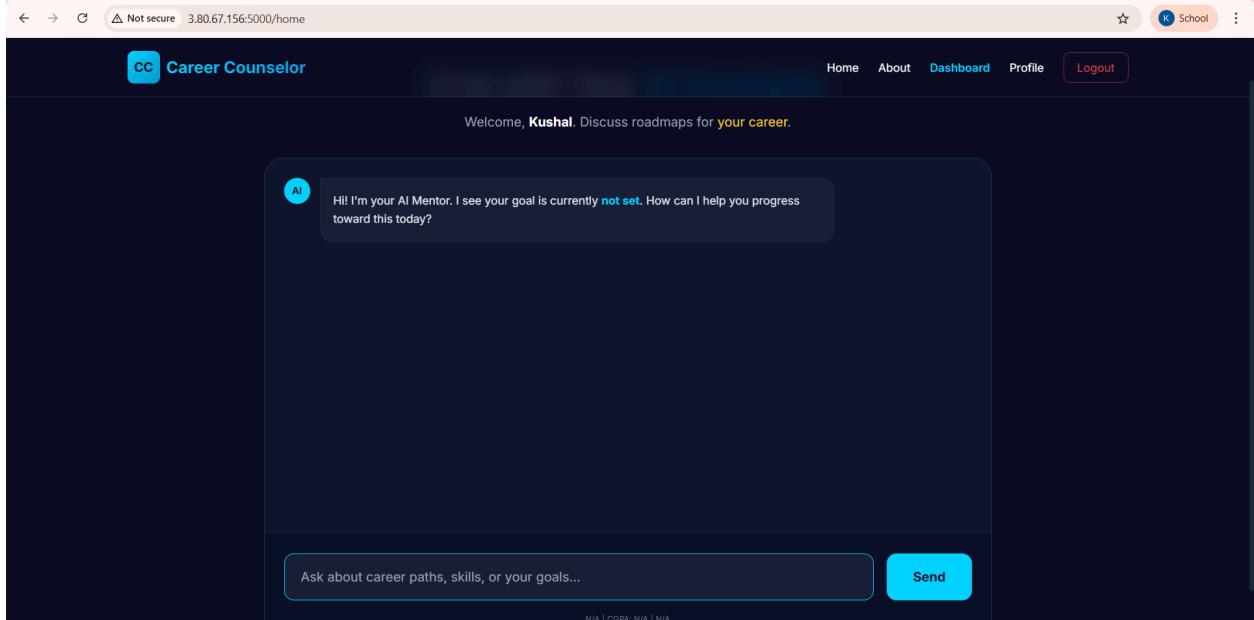
```
i-088638fd7729be1eb (CareerCounselor-Server)
PublicIPs: 3.80.67.156 PrivateIPs: 172.31.18.168
```

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

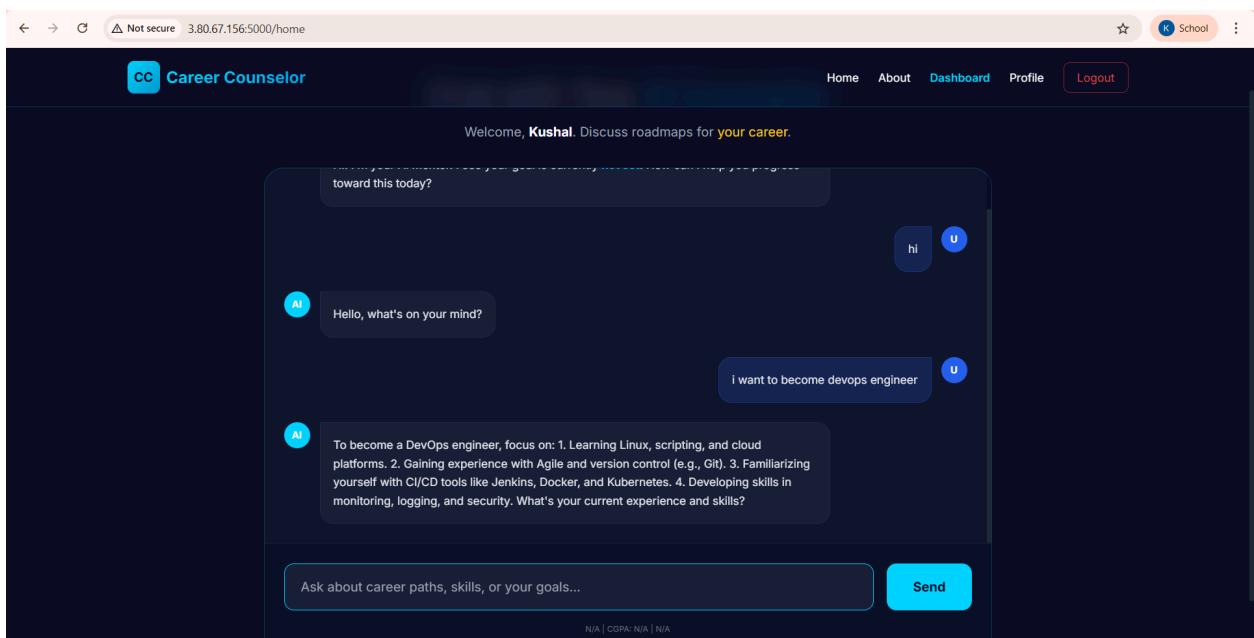
- PublicIPs: **http:3.80.67.156.5000** (Now no longer available)

Activity-2:

- 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:



The screenshot shows a web browser window for 'Career Counselor' at the URL 3.80.67.156:5000/home. The page has a dark theme. At the top, there's a navigation bar with links for Home, About, Dashboard, Profile, and Logout. A user profile icon for 'School' is also visible. The main content area features a welcome message: 'Welcome, Kushal. Discuss roadmaps for your career.' Below this is a large, rounded rectangular input field containing a message from an AI mentor: 'Hi! I'm your AI Mentor. I see your goal is currently **not set**. How can I help you progress toward this today?'. At the bottom of the input field is a 'Send' button.

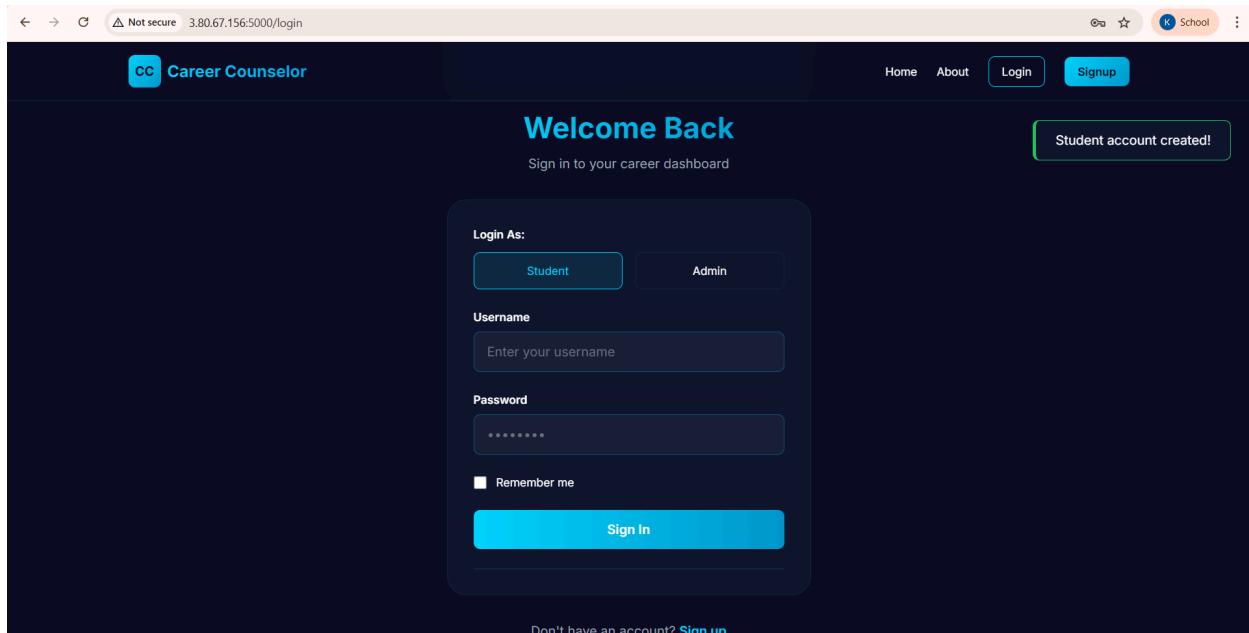


This screenshot shows the same 'Career Counselor' interface after the user has interacted with the AI. The AI message now includes a question: 'toward this today?'. The user has responded with 'hi' and 'u'. The AI has responded with 'Hello, what's on your mind?'. The user has typed 'i want to become devops engineer'. The AI has provided a detailed response: 'To become a DevOps engineer, focus on: 1. Learning Linux, scripting, and cloud platforms. 2. Gaining experience with Agile and version control (e.g., Git). 3. Familiarizing yourself with CI/CD tools like Jenkins, Docker, and Kubernetes. 4. Developing skills in monitoring, logging, and security. What's your current experience and skills?'. At the bottom of the input field is a 'Send' button.



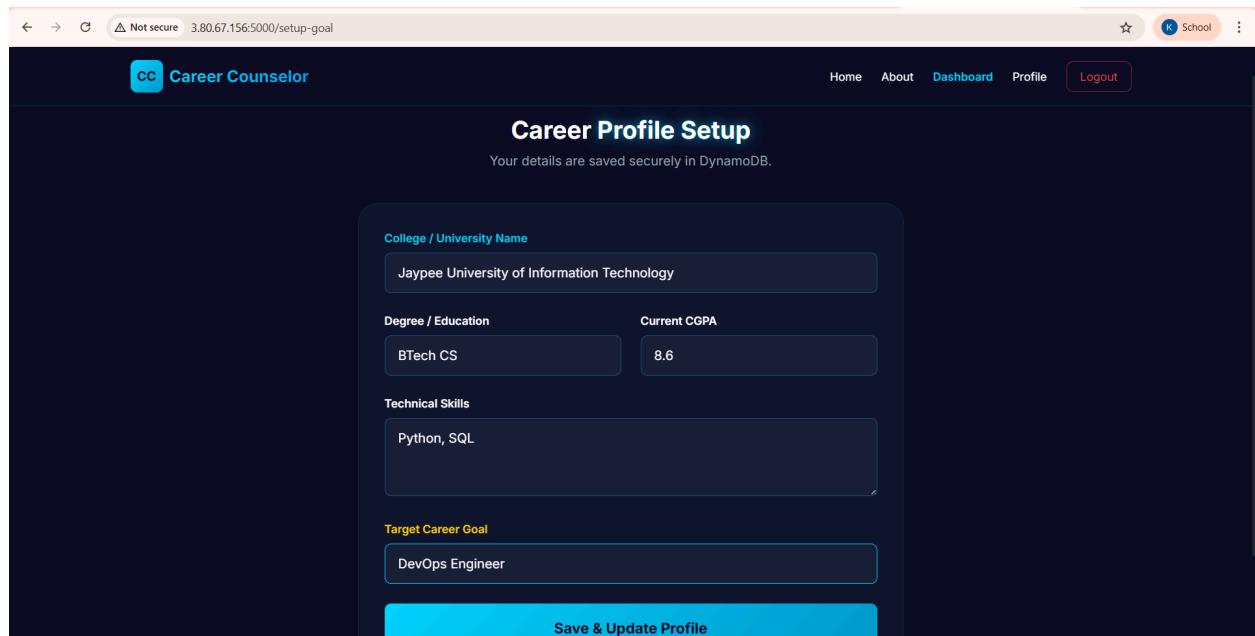
SMARTBRIDGE

- **Login Page:**



The screenshot shows a web browser window with the URL 3.80.67.156:5000/login. The page has a dark blue header with the "Career Counselor" logo. Below the header, a "Welcome Back" message and a "Sign in to your career dashboard" link are displayed. A "Student account created!" message is shown in a green box. The main area contains a login form with fields for "Login As" (Student or Admin), "Username" (placeholder: Enter your username), "Password" (placeholder: *****), a "Remember me" checkbox, and a "Sign In" button. At the bottom, there's a link for users without an account: "Don't have an account? [Sign up](#)".

- **Setup-Goal Page:**



The screenshot shows a web browser window with the URL 3.80.67.156:5000/setup-goal. The page has a dark blue header with the "Career Counselor" logo. Below the header, a "Career Profile Setup" title and a note "Your details are saved securely in DynamoDB." are displayed. The main area contains several input fields: "College / University Name" (Jaypee University of Information Technology), "Degree / Education" (BTech CS), "Current CGPA" (8.6), "Technical Skills" (Python, SQL), and "Target Career Goal" (DevOps Engineer). A large blue "Save & Update Profile" button is at the bottom.



- **Dashboard Page:**

The screenshot shows the 'Career Intelligence Dashboard' for a user named Kushal. The dashboard includes a 'CURRENT STANDING' section with target 'DevOps Engineer', institution 'Jaypee University of IT', and CGPA '8.6'. An 'AI Skill Analysis' section lists 'INDUSTRY ALIGNMENT' with Python, SQL, and Problem-Solving checked, and 'CRITICAL GAPS' with Containerization using Docker, Cloud Computing using AWS or Azure, and Continuous Integration/Continuous Deployment using Jenkins or GitLab CI/CD. The main area is divided into 'PHASE 1: BUILDING FOUNDATIONS' and 'PHASE 2: EXPANDING TECHNICAL SKILLS', each with a list of learning objectives.

The screenshot shows the same 'Career Intelligence Dashboard' for Kushal. A large circular icon with a map symbol is displayed in the center. Below it, text reads: 'Click 'Generate Roadmap' to architect your career path using Llama-3.3.'



SMARTBRIDGE

- **About Page:**

The screenshot shows a web browser window with the URL 3.80.67.156:5000/about. The page has a dark blue header with the "cc" logo and "Career Counselor" text. A navigation bar includes Home, About, Dashboard, Profile, and Logout. The main content area features a title "About the Virtual Career Counselor" and a subtitle "An intelligent system leveraging Generative AI and Cloud Computing to bridge the gap between education and industry." Below this are two rounded rectangular boxes: one for "Generative AI (Groq)" and one for "AWS Cloud Infrastructure". The "Generative AI (Groq)" box contains text about Llama-3.3-70b via the Groq API. The "AWS Cloud Infrastructure" box contains text about persisting user data using AWS DynamoDB. At the bottom, a section titled "Key Functionalities" lists "Intelligent Roadmap Generation" and "Real-time Skill Gap Analysis".

- **Signup Page:**

The screenshot shows a web browser window with the URL 3.80.67.156:5000/signup. The page has a dark blue header with the "cc" logo and "Career Counselor" text. A navigation bar includes Home, About, Login, and Signup. The main content area features a title "Create Account" and a subtitle "Join the AI-powered career platform". A large central form box contains fields for "I want to join as a:" (Student or Admin), "Username" (KK), and "Password" (two asterisks). A "CREATE MY ACCOUNT" button is at the bottom. At the bottom of the page, there is a link "Already have an account? Log in here".



- Admin-Dashboard Page:

A screenshot of a web-based admin dashboard titled "Admin Command Center". The dashboard has a dark theme with several cards and sections. At the top left is the "cc Career Counselor" logo. Top right navigation includes "Home", "About", "Dashboard" (which is highlighted in blue), "Profile", and "Logout".

- TOTAL STUDENTS**: 2
- CLOUD STATUS**: AWS Live

Username	Goal	CGPA
Kushal	DevOps Engineer	8.6

Llama 3.3 System Insights

Top skills students are missing based on AI Gap Analysis:

Skill	Status
SQL & Database Management	High Demand
Cloud Infrastructure (AWS)	Trending



SMARTBRIDGE

- CarrerAI Notifications:

20 of 7,078 < >

 **CareerAI** 8 Feb 2026, 20:34 (2 days ago) 
User KK has performed: Registered as admin

 **CareerAI** 8 Feb 2026, 20:35 (2 days ago) 
User KS has performed: Registered as admin

 **CareerAI** 8 Feb 2026, 20:42 (2 days ago) 
User Kushal has performed: Updated target to DevOps Engineer -- If you wish to stop receiving notifications from this topic, please click or visit the link belo

 **CareerAI** 8 Feb 2026, 20:42 (2 days ago) 
User Kush has performed: Registered as student

 **CareerAI** <no-reply@sns.amazonaws.com> 8 Feb 2026, 20:44 (2 days ago)   
to me ▾
User KN has performed: Registered as admin

SMARTBRIDGE

- **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.