

# Cloud Based Voting System With AWS API Gateway and Lambda

*A Project Based Learning Report Submitted in partial fulfilment of the requirements for the  
award of the degree*

*of*

**Bachelor of Technology**

**in The Department of Computer Science & Engineering**

**Cloud Based AI/ML Speciality (22SDCS07A)**

Submitted by

**2210030455: Karra Sri Charan Reddy**

Under the guidance of

**Ms. P. Sree Lakshmi**



Department of Electronics and Communication Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075

FEB - 2025.

# Introduction

## A Cloud-Based Voting System Utilizing AWS API Gateway and AWS Lambda

A Cloud-Based Voting System utilizing AWS API Gateway and AWS Lambda is a secure, scalable, and efficient solution for conducting online elections. This project leverages serverless computing to eliminate the need for traditional infrastructure, reducing operational costs while ensuring high availability and reliability.

The system enables users to cast votes online through a web or mobile application, which interacts with a RESTful API managed by AWS API Gateway. The API routes requests to AWS Lambda functions, which handle authentication, vote processing, and result computation in a secure and stateless manner. To ensure data integrity and security, AWS DynamoDB or Amazon RDS can be used for storing votes, with AWS Cognito providing authentication and user management.

This architecture allows seamless scalability, accommodating high voter traffic without performance degradation. AWS Lambda ensures cost-efficiency by executing functions only when needed, eliminating the need for persistent servers. The integration of AWS Cognito enhances security by enabling multi-factor authentication (MFA) and access control. Additionally, API Gateway provides rate limiting and request validation to prevent fraudulent activities. By leveraging AWS CloudWatch, administrators can monitor real-time system performance and detect anomalies, ensuring a transparent and tamper-proof voting process. This system is well-suited for government, corporate, and community elections, ensuring a fair and efficient electoral process.

## Architecture Overview:

- Frontend (React/Flutter/Web App) → Sends requests to API Gateway.
- API Gateway → Routes HTTP requests to AWS Lambda.
- AWS Lambda (Python) → Processes requests and interacts with DynamoDB.
- DynamoDB → Stores voting data (e.g., users, votes, results).
- Authentication (AWS Cognito) → Handles user login.

## Literature Review/ Application Survey

### 1. API Gateway

AWS **API Gateway** is a fully managed service that allows developers to create, publish, and manage secure APIs at any scale. It helps streamline API management while ensuring high availability, security, and scalability.

#### Use Cases of API Gateway:

##### a) E-Commerce Platforms

- E-commerce applications require **secure and scalable APIs** to handle multiple transactions, including product listings, payments, and user authentication.
- **API Gateway** helps **route requests efficiently** between the frontend, backend, and third-party payment services like Stripe or PayPal.
- It enforces security features such as **OAuth authentication, API rate limiting, and request validation** to prevent fraud.

##### b) Banking and Financial Applications

- Banks and financial institutions use API Gateway to **expose secure endpoints** for online banking, fund transfers, and mobile transactions.
- It integrates with **AWS Lambda and AWS WAF (Web Application Firewall)** to **prevent DDoS attacks and unauthorized access**.
- API Gateway ensures high availability, allowing millions of users to conduct transactions **simultaneously without downtime**.

##### c) IoT Solutions and Smart Devices

- API Gateway is used in IoT applications to route requests between **smart home devices** and cloud services.
- For example, **home automation systems** can send requests via API Gateway to control smart lights, thermostats, and security cameras.
- It enables **low-latency communication**, ensuring real-time responses for IoT commands.

## 2. AWS Lambda

AWS **Lambda** is a serverless computing service that automatically runs code in response to events. It allows developers to execute functions without provisioning or managing servers.

### Use Cases of AWS Lambda:

#### a) Automated Data Processing

- Lambda is widely used in **real-time event processing**, such as stock market analysis, fraud detection, and analytics.
- In **stock trading**, Lambda functions process live stock prices and execute trades when certain conditions are met.
- In **fraud detection**, banks use Lambda to analyze transactions and flag suspicious activities instantly.

#### b) Chatbots and AI Assistants

- Many customer service chatbots use AWS Lambda to **process user queries dynamically** and fetch data from databases.
- AI-driven assistants like **Alexa and Google Assistant** use Lambda functions to analyze voice inputs and respond accordingly.
- Lambda **integrates with Amazon Lex (NLP engine)** to enable conversational AI experiences.

#### c) Video and Image Processing

- AWS Lambda is used in media applications to automate **image resizing, video compression, and media encoding**.
- For example, platforms like **YouTube and Instagram** trigger Lambda functions when users upload videos, automatically optimizing them for different screen sizes.
- Lambda integrates with **Amazon S3 and Rekognition** for **real-time object detection, facial recognition, and content moderation**.

## 3. DynamoDB

AWS **DynamoDB** is a fully managed NoSQL database designed for **high availability, scalability, and low-latency performance**. It is widely used in applications that require real-time access to large datasets.

### Use Cases of DynamoDB:

#### a) Gaming Leaderboards

- Online multiplayer games use DynamoDB to store **real-time scores, rankings, and player statistics**.
- It supports high-throughput **read and write operations**, ensuring that millions of players can access leaderboards instantly.
- Games like **Fortnite and Call of Duty** rely on DynamoDB for **low-latency player data retrieval**.

### **b) Personalized Content Delivery**

- **Streaming platforms like Netflix and Spotify** use DynamoDB to store **user preferences, watch history, and recommendations**.
- The NoSQL structure allows fast and flexible queries, ensuring users receive personalized content recommendations in real-time.
- It enables seamless transitions across devices, where a user can start a movie on one device and continue from the same point on another.

### **c) Log Management and Monitoring**

- Many enterprises use DynamoDB to store **application logs, system monitoring data, and security audit trails**.
- It integrates with **Amazon CloudWatch** to **analyze trends, detect anomalies, and generate alerts** in case of security threats.
- DynamoDB's **automatic scaling** allows storage of **millions of log records per second** without performance degradation.

## **4. AWS Cognito**

AWS **Cognito** provides **user authentication, authorization, and user management** for web and mobile applications. It simplifies identity management by handling login, token generation, and multi-factor authentication (MFA).

### **Use Cases of AWS Cognito:**

#### **a) Secure User Authentication in Mobile Apps**

- Apps like **Uber, Airbnb, and Facebook** use AWS Cognito to manage **user sign-ins and authentication securely**.
- It supports **social logins (Google, Facebook, Apple)** and traditional email/password authentication.
- Cognito integrates with **OAuth 2.0, OpenID Connect, and SAML**, allowing seamless authentication across platforms.

#### **b) Multi-Factor Authentication (MFA) for Enhanced Security**

- AWS Cognito enables **MFA** to enhance security in banking, healthcare, and enterprise applications.
- It allows authentication via **SMS OTP, authenticator apps (Google Authenticator), and biometric authentication**.
- This additional security layer prevents unauthorized access, ensuring that sensitive data is **protected from cyber threats**.

#### **c) Enterprise Identity Management and Single Sign-On (SSO)**

- Large organizations use AWS Cognito for **employee authentication, workforce access control, and single sign-on (SSO) solutions**.
- Cognito integrates with **Microsoft Active Directory, Okta, and AWS IAM**, allowing secure login across corporate applications.
- This reduces the need for multiple passwords, improving **security and employee productivity**.

## References

[1] Amazon API Gateway – Developer Guide, AWS Documentation. Available at:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

[2] AWS Lambda – Developer Guide, AWS Documentation. Available at:

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

[3] Amazon DynamoDB – NoSQL Database Service, AWS Documentation. Available at:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>

[4] Amazon Cognito – Authentication and User Management, AWS Documentation. Available at:

<https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>

[5] Amazon S3 – Object Storage Service, AWS Documentation. Available at:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

[6] AWS Case Study: How Netflix Uses AWS for Scalability. Available at:

<https://aws.amazon.com/solutions/case-studies/netflix/>

[7] AWS Case Study: How Airbnb Manages Authentication with AWS Cognito. Available at:

<https://aws.amazon.com/solutions/case-studies/airbnb/>

[8] API Gateway Use Cases – AWS Whitepaper. Available at: <https://aws.amazon.com/api-gateway/use-cases/>

[9] AWS Financial Services: How Banks Use AWS for Fraud Detection and Security. Available at:

<https://aws.amazon.com/financial-services/>

[10] AWS Serverless Computing: Best Practices for API Gateway and Lambda. Available at:

<https://aws.amazon.com/serverless/>

These references provide in-depth insight

