

# **System Design Document**

## **For**

### **Analysis Tool for Commercially Available Cybersecurity AI Tools**

*Nicolas Rodriguez, Jeremiah Webb, Olivia Meholic, Sarah Gleixner, Joseph Alesandrini, Troy  
Neubauer*

<i>Version/Author</i>	<i>Date</i>
<i>V1.0/Group</i>	<i>29SEP2023</i>
<i>V2.0/Group</i>	<i>31OCT2023</i>
<i>V3.0/Group</i>	<i>21NOV2023</i>
<i>V4.0/Group</i>	<i>05FEB2024</i>

# TABLE OF CONTENTS

1	INTRODUCTION .....	3
1.1.1	Purpose and Scope .....	3
1.1.2	Project Executive Summary .....	3
1.1.4	Design Constraints .....	3
1.1.5	Future Contingencies .....	3
1.1.6	Document Organization .....	3
1.1.7	Project References .....	3
1.1.8	Glossary .....	4
2	SYSTEM ARCHITECTURE .....	4
2.1.1	System Hardware Architecture .....	4
2.1.2	System Software Architecture .....	4
2.1.3	Internal Communications Architecture .....	4
3	HUMAN-MACHINE INTERFACE .....	5
3.1.1	Inputs.....	6
3.1.2	Outputs.....	9
4	DETAILED DESIGN .....	13
4.1.1	Hardware Detailed Design .....	13
4.1.2	Software Detailed Design .....	13
4.1.3	Internal Communications Detailed Design .....	15
5	EXTERNAL INTERFACES .....	16
5.1.1	Interface Architecture .....	16
5.1.2	Interface Detailed Design .....	18
6	SYSTEM INTEGRITY CONTROLS .....	19
6.1.1	AWS Log Creation .....	19
6.1.2	AWS Log Storage .....	19
6.1.3	API Security.....	19
6.1.4	Object Storage Security .....	19
6.1.5	Database Security.....	20
6.1.6	User Control in AWS.....	20
6.1.7	User Control in CyberTool .....	20
6.1.8	Application Logging .....	20
6.1.9	Application Log Storage .....	21
6.1.10	Application Security .....	21
6.1.11	HTTPS Encryption.....	21

# SYSTEM DESIGN DOCUMENT

## 1 INTRODUCTION

### 1.1.1 Purpose and Scope

This document is designed to provide clear and concise technical guidance to the development team, ensuring consistency and alignment with the envisioned solution. Furthermore, it aids in facilitating communication among our product owner, enabling reviews, feedback, and ensuring that the software built aligns with the initial design objectives.

### 1.1.2 Project Executive Summary

This is version 1.0 of CyberTool, which is a public website that will enable users to filter through a variety of cybersecurity tools depending on the needs of the user (commercial, personal, etc.). This website will be hosted on an AWS server including all necessary architecture for the product. Among this architecture, there will be a database containing the cybersecurity products and the key values that were determined during research of said products. Recommendations will be generated from the database based on user preferences/requirements. Upon user request, a comparative report will be generated containing prospective tools and/or tools they already have.

### 1.1.3 System Overview



Figure 1: CyberTool Use Case Diagram

### 1.1.4 Design Constraints

The primary constraint on the CyberTool project is the number of tools contained in the DynamoDB database. This is due to the manual effort required by the development team to input these tools directly into the database. The maximum number of CyberTool database entries has been set to 100 to provide the best product given the small development team.

### 1.1.5 Future Contingencies

This system architecture is based upon the AWS hosting platform with its many integrated services. The success of the project depends on the successful flow of data between these integrated services; however, problems could arise due to differences in data types and individual system configurations. These issues can be resolved through a series of overall system inspections and refinements.

### 1.1.6 Document Organization

The Systems Design Document is intended to provide an overview of the system design. This documents reviews both the hardware and software for the System Architecture, its Human Machine Interface, both hardware and software Detailed Design, any External Interfaces being used, and System Integrity Controls implemented throughout.

### 1.1.7 Project References

Amazon Web Services Documentation, from <https://docs.aws.amazon.com/>

### 1.1.8 Glossary

AWS – Amazon Web Services

WAF – Web Application Firewall

SSE – Server-Side Encryption

IAM – Identity and Access Management

OU - Organization Unit

SCP - Service Control Policy

RBAC – Role-based Access Control

CIAM – Customer Identity and Access Management

## 2 SYSTEM ARCHITECTURE

This section describes an overview of the hardware and software architecture for the CyberTool system & subsystems.

### 2.1.1 System Hardware Architecture

The CyberTool system does not rely on specific hardware, however, all services and storage, databases, functions, will be stored on AWS Cloud Infrastructure.

### 2.1.2 System Software Architecture

### 2.1.3 Internal Communications Architecture

Figure 2 illustrates how our website will operate on the AWS server. A user can navigate to the CyberTool website where it will establish a connection to our delivery network CloudFront. Amplify will contain all HTML, CSS, and JavaScript files that will be delivered to the user's computer browser for viewing.

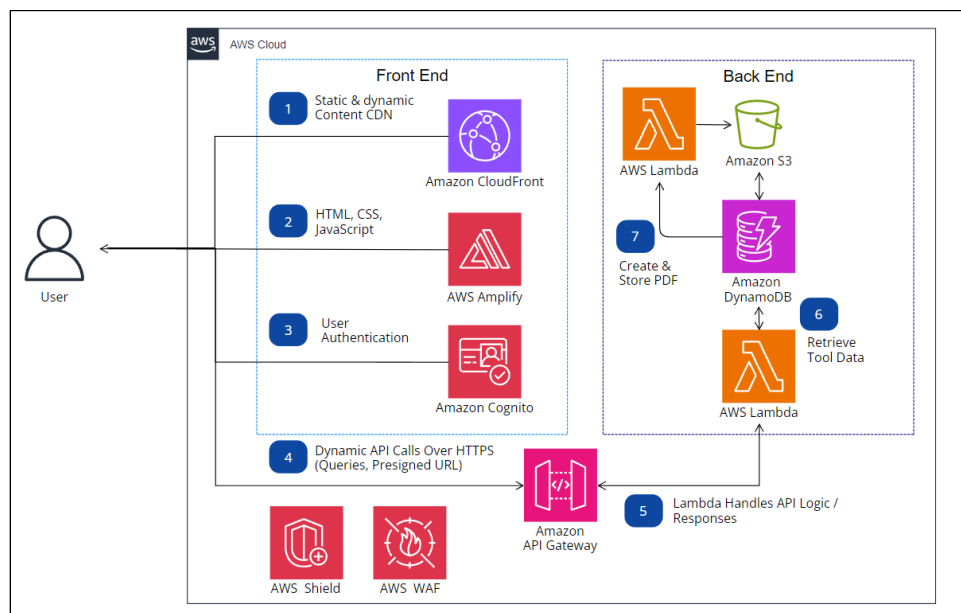


Figure 2: Internal Communications Architecture Overview

Figure 3 illustrates the authentication process using AWS Cognito. Upon a user's attempt to access the website, Cognito will first validate their credentials. Once verified, the user's request is approved, granting them entry to the site's content.

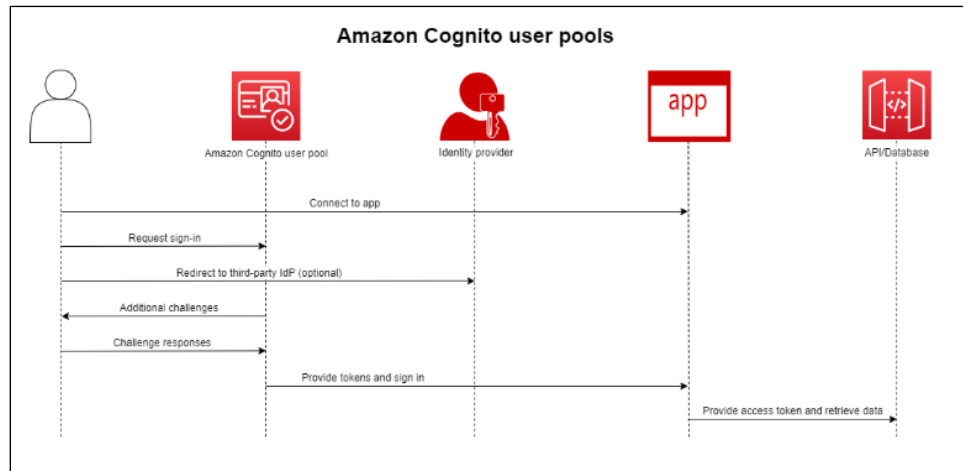


Figure 3: Amazon Cognito User Pool Authentication Data Flow

Note: The diagrams should map to the FRD context diagrams.

### 3 HUMAN-MACHINE INTERFACE

CyberTool is designed to be easily navigated by all users. As a general user, he/she will be able to view all the listed tools within the website and filter through based on user preferences. As an authorized user, he/she will be prompted for a username and password to be authenticated by Amazon Cognito. Authorized users will then be able to input their login information, as well as their preferences for the type of cybersecurity tool they are searching for once authenticated. With this information, our system will output a list of tools that match their preferences along with other recommended products. Authorized users will also have the option to generate a comparative report containing prospective tools and/or tools they already have.

### 3.1.1 Inputs

# CyberTools

## Welcome Back!

**Username:**

**Password:**

[Forgot password?](#)

[Login](#)

Don't have an account?

[Sign Up](#)

Figure 4: Sign In Page

Upon initialization of the system, the user will be prompted for a username and password to be entered in their respective text boxes as shown in Figure 4. Amazon Cognito is used to authenticate the users' credentials if they are already a registered user. If a user has forgotten their password to their username, they may use the "Forgot Password" button to navigate to the appropriate page to reset their password once authenticated as a registered user.

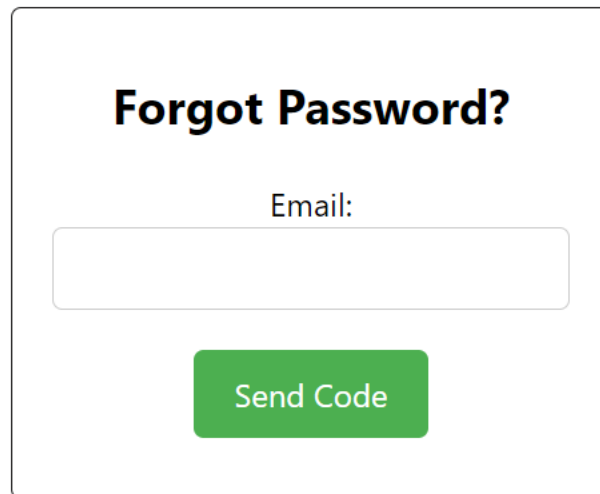
A rectangular form with a white background and a thin black border. At the top, the text "Forgot Password?" is centered in a bold, black font. Below this, the label "Email:" is centered. Underneath the label is a wide, empty text input field. At the bottom center of the form is a green rectangular button with the text "Send Code" in white.

Figure 5: Forgot Password Page

Upon pressing the “Forgot Password” button shown in Figure 4, users are taken to the above Forgot Password page to reset their password. Users will input their registered email address to receive a one-time validation code. Using Amazon Cognito, the user will only receive a validation code if they are authenticated.

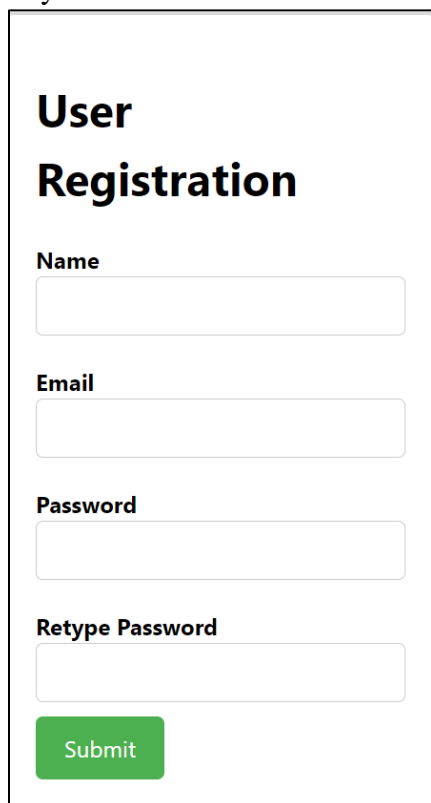
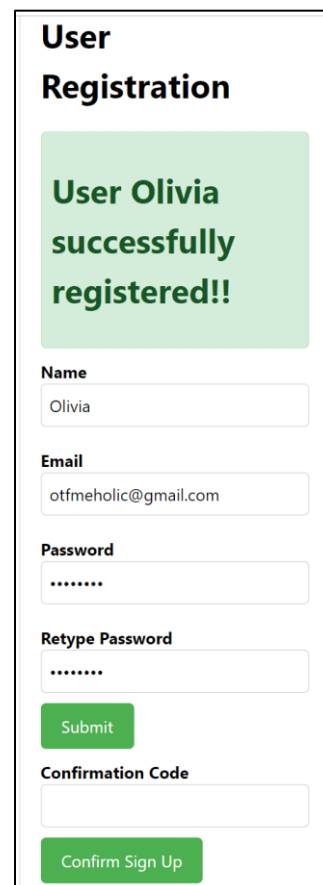
A vertical rectangular form with a white background and a thin black border. The title "User Registration" is at the top left in a bold, black font. Below the title are four labels: "Name", "Email", "Password", and "Retype Password", each followed by a text input field. At the bottom left is a green rectangular button with the text "Submit" in white.A vertical rectangular form with a white background and a thin black border. The title "User Registration" is at the top left in a bold, black font. Below the title is a green rectangular box with the text "User Olivia successfully registered!!" in a bold, dark green font. Below this box are four labels: "Name", "Email", "Password", and "Retype Password", each followed by a text input field. The "Name" field contains the text "Olivia". The "Email" field contains the text "otfmeholic@gmail.com". The "Password" and "Retype Password" fields contain the text ".....". At the bottom left is a green rectangular button with the text "Submit" in white. Below this button is a label "Confirmation Code" followed by a text input field. At the bottom right is a green rectangular button with the text "Confirm Sign Up" in white.

Figure 6: Sign Up Page and Success Message

If a user does not have an account, the sign-in window provides the option to sign up for an account. Upon choosing the “Sign Up” option, users are prompted to provide their full name, email, username, password, and a retype password in their respective text boxes. These data entries are intended to uniquely identify each user in the user database. Each text box has a character limit of 30 characters. If the user provides data that is not already found in our database, they are successfully registered.

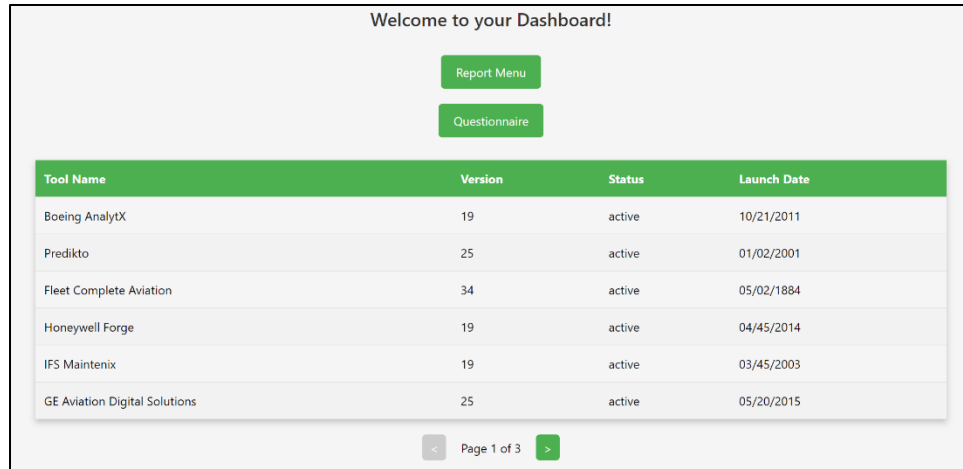


Figure 7 is a screenshot of the CyberTool Dashboard. At the top, it says "Welcome to your Dashboard!". Below this are two green buttons: "Report Menu" and "Questionnaire". The main part of the dashboard is a table with four columns: "Tool Name", "Version", "Status", and "Launch Date". The table lists six tools: Boeing AnalytX, Predikto, Fleet Complete Aviation, Honeywell Forge, IFS Maintenix, and GE Aviation Digital Solutions. At the bottom of the table, there is a pagination control showing "Page 1 of 3" with left and right arrow buttons.

Tool Name	Version	Status	Launch Date
Boeing AnalytX	19	active	10/21/2011
Predikto	25	active	01/02/2001
Fleet Complete Aviation	34	active	05/02/1884
Honeywell Forge	19	active	04/45/2014
IFS Maintenix	19	active	03/45/2003
GE Aviation Digital Solutions	25	active	05/20/2015

Figure 7: CyberTool Dashboard

After successfully signing in, the user will be directed to the dashboard that gives them access to the system’s list of cyber tools. Within this website, users are able to select various filters to query our database based on their preferences.



## Comparative Report Questionnaire

[Dashboard](#)[Report Menu](#)

What specific aerospace-related projects or systems are you involved in, and what kind of data or assets do they handle?

- ☐ Aircraft design and manufacturing
- ☐ Satellite communication systems
- ☐ Air traffic control systems
- ☐ Drone operations

Are there any regulatory compliance requirements (e.g., FAA, NIST, or other standards) that your organization must adhere to regarding cybersecurity?

- ☐ Yes, FAA regulations
- ☐ Yes, NIST standards
- ☐ No specific regulations
- ☐ Other (please specify)

What are the primary threats or vulnerabilities you anticipate facing in the aerospace industry?

- ☐ Insider threats
- ☐ Cyber espionage
- ☐ Malware attacks
- ☐ Supply chain vulnerabilities

Are you aware of any recent cybersecurity incidents or breaches within the aerospace industry that have raised concerns?

- ☐ Yes
- ☐ No
- ☐ Not sure
- ☐ Prefer not to answer

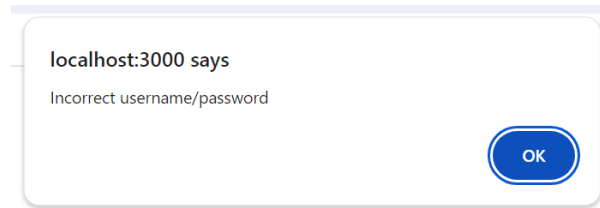
Figure 8: Questionnaire

Upon choosing “Questionnaire” from Figure 7, authorized users are presented with the above questionnaire for them to input preferences to serve as the query on the main database of stored cyber tools. The survey includes explicit questions with answers that will define which attributes the user desires in their cyber tool. Only one answer will be accepted per question.

After completion of the previous survey, users are presented with the above chart displaying options matching their desired preferences for a cybersecurity tool. As an additional input, users are able to filter/sort the list of all recommended tools based on alphabetical order, type (commercial/personal), or date released.

### 3.1.2 Outputs

Displayed below is our current login interface for CyberTool. The page features fields for entering both a username and password. Once these details are provided, users can click the ‘Login’ button to access CyberTool. For those without an account, there’s a ‘Sign Up’ option available for registration.



## CyberTools

### Welcome Back!

**Username:**

**Password:**

[Forgot password?](#)

[Login](#)

Don't have an account?

[Sign Up](#)

Figure 9: Login Page Error Message

Outputs associated with the above login screen include an error message if user credentials are not accepted by Amazon Cognito.

The image shows a web form titled "User Registration". At the top, there is a pink rectangular box with the text "Please enter all the fields" in a dark red font. Below this, the form contains four input fields: "Name" with the text "Oliviaaaaaaaaaaaaaaaaaaaaaa", "Email" with the text "meholico@my.erau.edu", "Password" with masked characters (dots), and "Retype Password" also with masked characters. At the bottom of the form is a green "Submit" button.

Figure 10: Sign Up Page Error Message

Outputs for the above Sign-Up display include an error message if the user attempts to include more than 25 characters in any text entry. An additional error message is displayed if the desired username is already assigned to another user in the database.

The above display functions as a system output after the completion of the cyber tools survey. This output/summary screen includes all tools that match user preferences from the previous survey. (need visual for this report)

# Report Dashboard

Dashboard Refresh

File Name	Date Created	Download
IAMANOTHERTEST	11/1/2023	Download
IAMANOTHERTEST	11/1/2023	Download
IAMANOTHERTEST	11/1/2023	Download

Figure 11: Report Dashboard

As a result of choosing “Report Menu” from the previous summary output, the above display is shown to the user. This display shows all past comparative reports generated by that unique user. Upon choosing “Download” in the above display, users can view past summary outputs based on past user preferences.

Welcome to your Dashboard!

Report Menu Questionnaire

Tool Name	Version	Status	Launch Date
Boeing AnalytX	19	active	10/21/2011
Predikto	25	active	01/02/2001
Fleet Complete Aviation	34	active	05/02/1884
Honeywell Forge	19	active	04/45/2014
IFS Maintenix	19	active	03/45/2003
GE Aviation Digital Solutions	25	active	05/20/2015

< Page 1 of 3 >

Figure 12: CyberTool Dashboard

In addition, the image above shows the initial layout we are currently working on for listing the various tools accessible to users. Each tool entry is defined by its name, version, status, and launch

date. Many of these categories will be modified in the future. Additionally, we have started developing a filtering tool.

## 4 DETAILED DESIGN

**Front-End Design:** The front-end prioritizes user accessibility, incorporating a search bar and tailored filter options. Its design adapts seamlessly across devices. Users can create accounts for personalized experiences, saving preferences, and generating reports. Further optimization may include user feedback features and user guides.

**Back-End Design:** The entirety of the back-end is supported by AWS. DynamoDB is a scalable, NoSQL database that was used to store cyber tools for querying. The sorting system is based on a primary key of the tool function and a secondary sorting key of a unique identifier string. These two keys, including a list of defined attributes, allow for the system to navigate and locate tools in the table. DynamoDB allows the developers to add, update, delete, and compare items in the table. API Gateway is used to connect DynamoDB to CloudWatch and allows the front-end to communicate with the back-end. API Gateway uses endpoints that interpret the user preferences (attained from questionnaire) and translate this to a query of various sorting keys and attributes in DynamoDB. Amazon S3 is integrated with DynamoDB to create and store PDF results for the user based on the query results.

**Research:** We maintain a continually updated database featuring the latest in cybersecurity tools, each subjected to rigorous verification. User feedback informs our improvements. Emphasis is given to highlighting trending tools, potential collaborations, and user education through research.

### 4.1.1 Hardware Detailed Design

CyberTool is strictly hosted on AWS infrastructure and servers. For more insight into what technologies are used, please look at the AWS website. No hardware is handled directly by the team.

### 4.1.2 Software Detailed Design

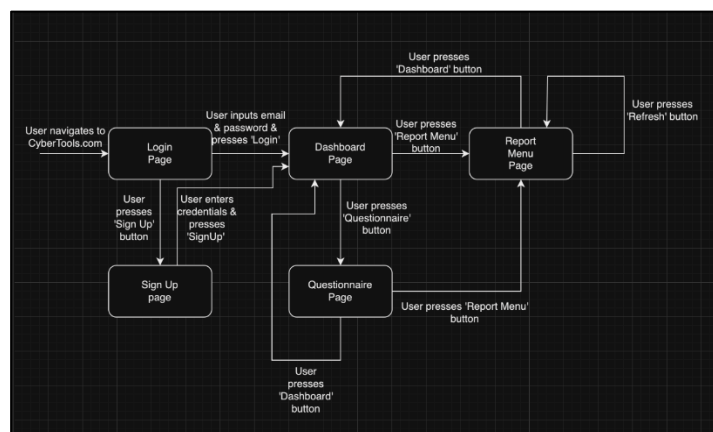


Figure 13: Front-end diagram

Figure 13 illustrates the front-end process using AWS Amplify & AWS React. When the user navigates to CyberTools.com, the login page will be presented to the user. Once verified, the user will be directed to the dashboard page. If the user does not have an account, they can enter their credentials to sign up. Once finished, the user is directed to the dashboard page. Once on the dashboard page, they can press the ‘Questionnaire’ button or the ‘Report Menu’ button. If the questionnaire is clicked, the user can choose to go back to the dashboard to the report menu page. If the report menu page is clicked, the user can go back to the dashboard page or click the refresh button, which essentially reloads the page.

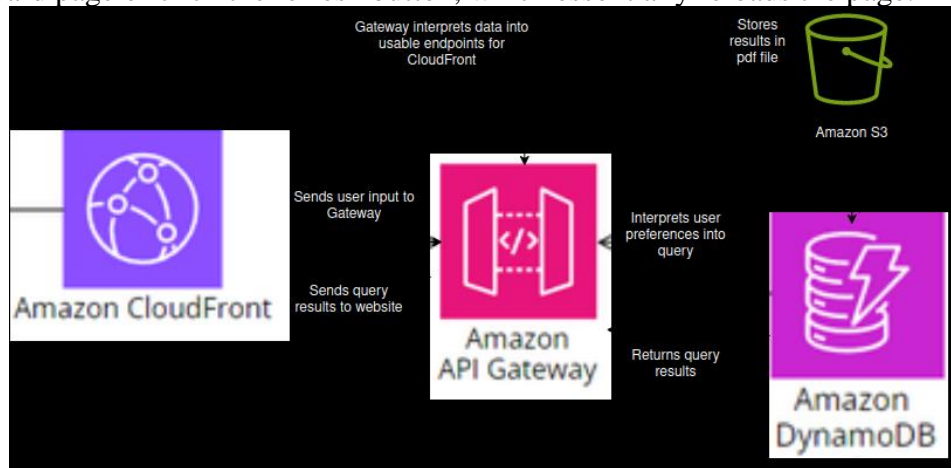


Figure 14: Back-end Diagram

Figure 14 depicts the framework of the database and how it communicates with the front end. Once the user has completed their questionnaire, the data is routed to AWS API Gateway where the request is interpreted as a Lambda function. This JavaScript method serves as the query for DynamoDB. The results are interpreted into a Json file that is routed back to the website using Gateway. Amazon S3 also stores the results as a PDF file in the event the user decides they want this information.

### 4.1.3 Internal Communications Detailed Design

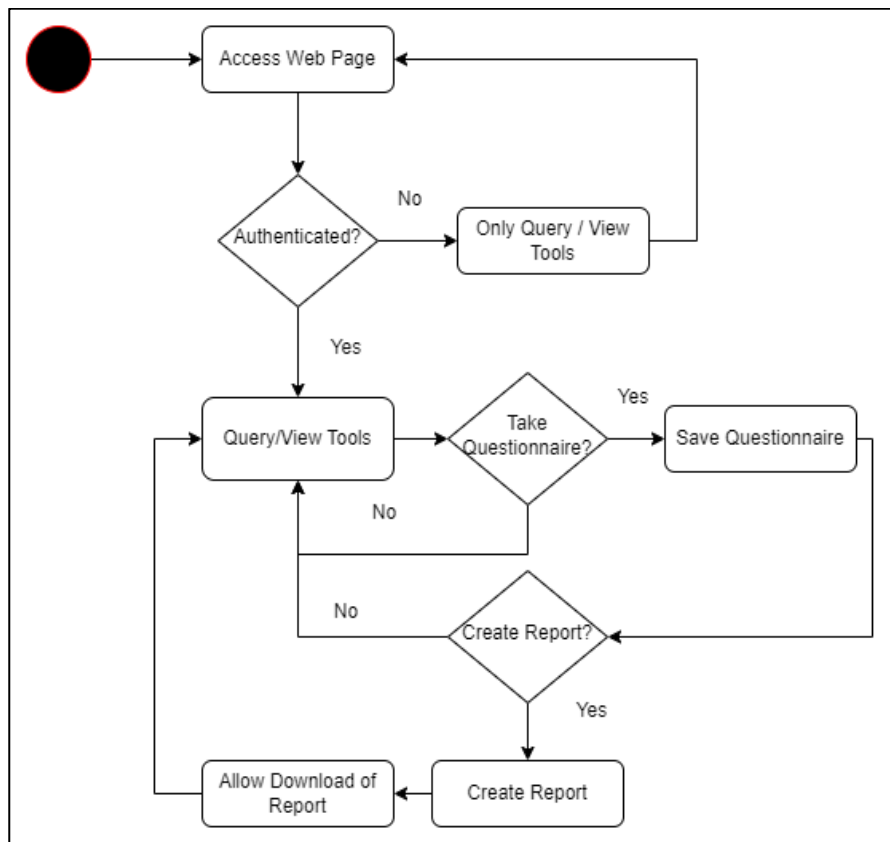


Figure 15: State Transition Diagram for CyberTool Website Users

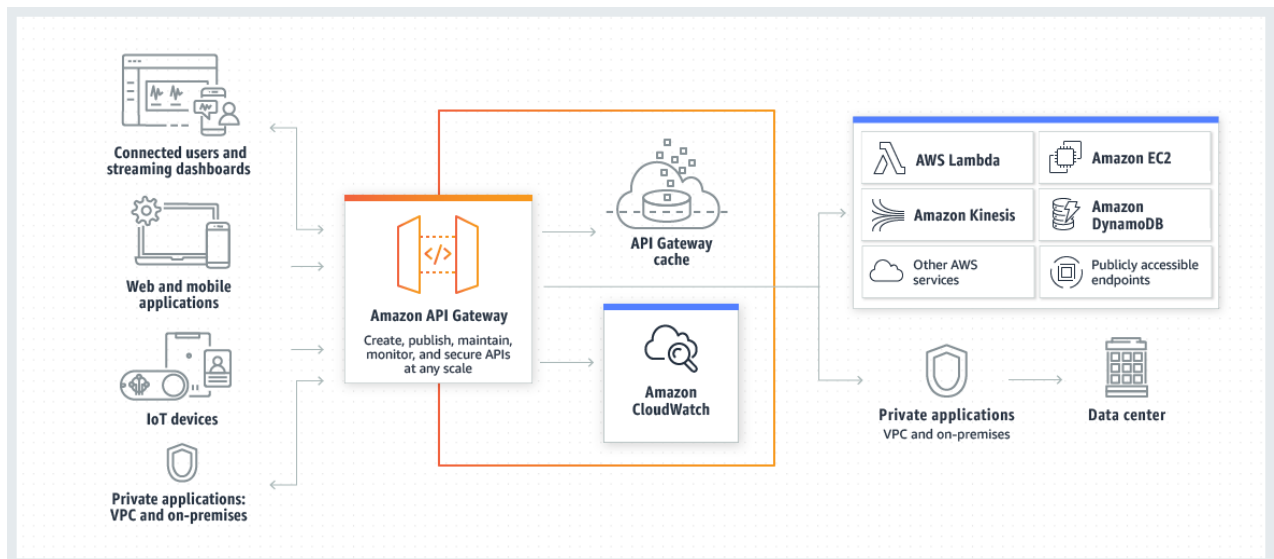


Figure 16: Amazon API Gateway architecture

Amazon API Gateway is used to facilitate communication between different AWS services by allowing for the development of a RESTful API.

API Gateway makes use of AWS Lambda, another AWS-provided service which allows for code to be run on the serverless web application. Together with Lambda, API Gateway will act as a proxy between the front-end web application managed through Amplify and the back-end resources and data stored in DynamoDB.

For both developers and users, API Gateway uses URL endpoints to invoke different HTTP methods to access or modify resources. These methods and resources are detailed in section 5.1.1.1.

## 5 EXTERNAL INTERFACES

CyberTool does not utilize data from external interfaces to operate. All systems such as Amazon Cognito, AWS Amplify, DynamoDB, and other AWS services are currently used to securely host our website. Users will only have to use browsers as specified to interact with the web application.

### 5.1.1 Interface Architecture

The primary internal communication for the CyberTool system will be via a web browser as specified.

#### 5.1.1.1 API Architecture

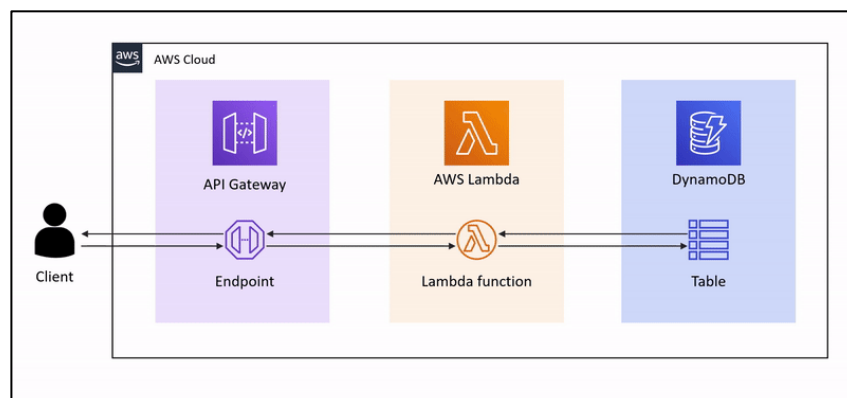


Figure 17: CyberTool Interface API Proxy Architecture

#### HTTP methods

The API will support the following HTTP methods:

GET: retrieve data

POST: create data

PUT: update data

DELETE: delete data

#### Request bodies & request headers

Request headers contain metadata and instructions for the API regarding the request. Necessary headers will include:

*Content-Type*: specifies the format of the data in the request body (this will almost always be JSON).



*Authorization:* contains credentials for authenticating the client making the request.

*Accept:* specifies the preferred media type for the response.

Request bodies contain the actual data regarding the request if any further data is needed. For example, a POST request with a JSON content type will contain the key-value attributes for a new item formatted in JSON in the request body.

### Query parameters

Endpoints can include query parameters to provide additional information for the request. Query parameters are appended to the endpoint URL.

This is important for filtering queries. For example, the request

GET /tools?category=log\_analysis&year=2004

should only retrieve tools categorized as Log Analysis tools that launched in 2004.

Query parameters shall accept strings, integers, or Boolean values (true/false) as data types depending on the attribute being requested.

### Resource endpoint examples

POST beginCreateReportForUser - Create a new Report for user

POST fetchDashboard - Get a list of tools for user for dashboard presentation

POST getReportListForUser - Get a list of reports for user that can be downloaded

POST getSingularToolData - Get a tool's data for a webpage

Post getPsignedURL - Get a url to download a reports URL.

#### 5.1.1.2 Report Generation & Architecture

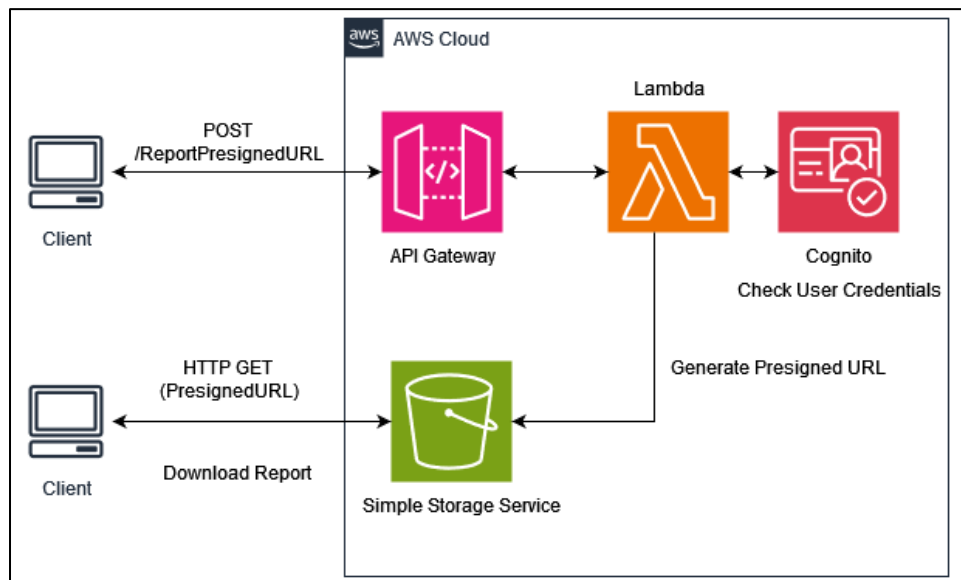


Figure 18: GetPsignedURL Interface for Getting an Object from S3.

By using API Gateway as an interface and proxy for resources on AWS, the user will send the API user credentials & a specific UUID for the report. Using Lambda & Cognito to authenticate the user's credentials and ensure they have personal access to these files on the cloud, the API will send back a URL, containing a JWT that contains security credentials for a limited time, in this

case 1 week, to access the specified report file. This report file will be accessed via the browser at it will be a PDF file type.

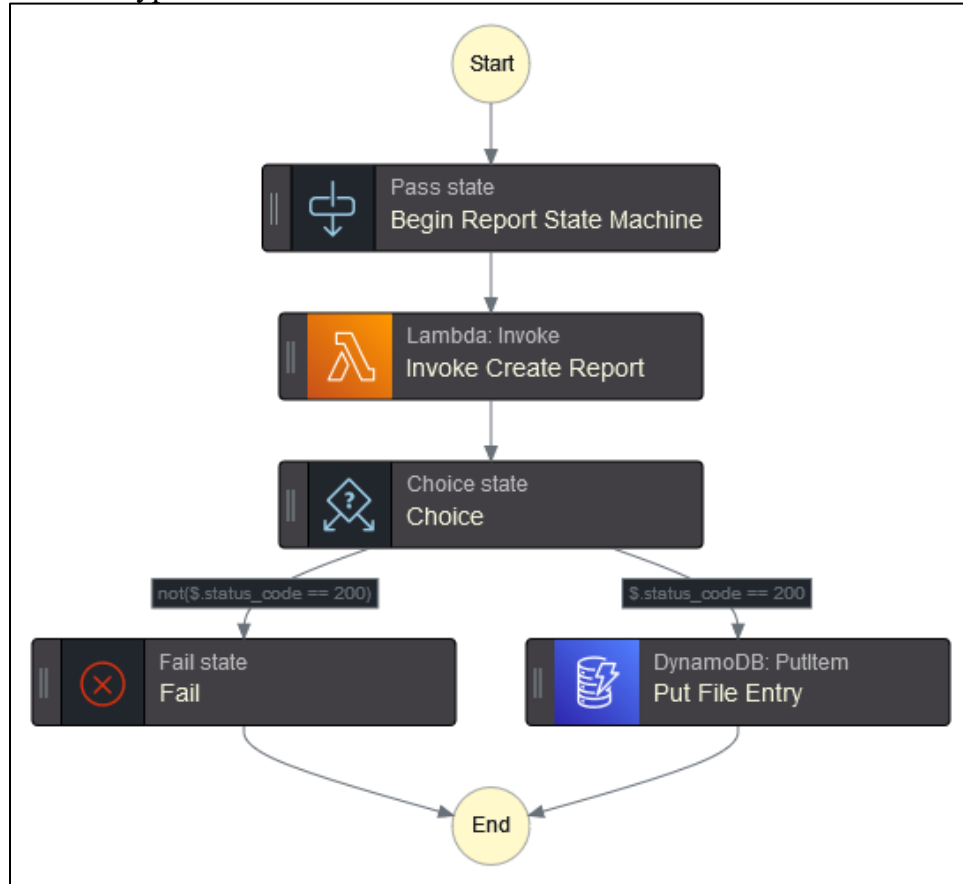


Figure 19: Report Step Function Diagram

Figure 19 represents the steps and states the “CreateReportStepFunction” API endpoint will be. Once a user has completed the questionnaire, this endpoint will receive and serialize the questions & answers, and generate a report based on the answers & other entered data, such as already used cyber tools. Within the Lambda function, the report will be physically created to a pdf, and stored within S3. At any time if the report generation fails, a server failure 500 error tells the user what error and when it happened. At success of generation, the file’s location in s3 is saved in DynamoDB and can be queryable by the user in the ListReport endpoint.

## 5.1.2 Interface Detailed Design

### 5.1.2.1 Data Format Requirements

All data passed from front-end React Website to API Gateway must be done via JSON Document I/O.

### 5.1.2.2 Security Interaction Front/Back-End

All website connections are done with HTTPS protocol, SSL/TLS.

### 5.1.2.3 Error Handling

Client-Server errors will be reported to users via JavaScript Alerts.  
Creation failures & errors will be reported to users via emails.

## **6 SYSTEM INTEGRITY CONTROLS**

### **6.1.1 AWS Log Creation**

#### **6.1.1.1 Amazon CloudWatch**

Function: Central log management service for collecting, storing, and monitoring log data from a wide range of AWS resources and applications.

Organization: Log data is organized into log groups and log streams, facilitating efficient log management.

#### **6.1.1.2 AWS CloudTrail**

Function: Provides a detailed history of API calls made on an AWS account, offering insights into who made the calls and the actions performed.

Importance: Crucial for security and compliance monitoring, as it records all API activities.

### **6.1.2 AWS Log Storage**

#### **6.1.2.1 Amazon S3 (Simple Storage Service)**

Function: Amazon S3 simple object storage. Necessary for long-term log retention, storing both CloudWatch and CloudTrail Logs.

Retention: Amazon S3 allows for the configuration of retention policies, providing flexibility in meeting storage duration requirements.

### **6.1.3 API Security**

Using AWS Firewall Manager, employ AWS WAF to safeguard against common web exploits and attacks, complemented by request and response validation for input sanitization. Set up rate limiting and throttling to protect against abuse and Distributed Denial of Service (DDoS) attacks using AWS Shield. For API authentication, Amazon Cognito User Pools will act as access control to API.

### **6.1.4 Object Storage Security**

#### **6.1.4.1 Bucket Policies and Access Control Lists (ACLs)**

Bucket policies and ACLs control access to S3 buckets and objects, allowing you to grant or deny permissions to specific AWS accounts or users.

#### **6.1.4.2 Bucket and Object-Level Encryption**

Implement (SSE) to encrypt data at rest within S3. AWS provides three SSE options: SSE-S3, SSE-KMS, and SSE-C, each with distinct advantages.

#### **6.1.4.3 Access Logging**

Enable access logging on S3 buckets to maintain records of all requests made to objects, which is invaluable for auditing and monitoring access.

#### **6.1.4.4 Versioning**

Enable versioning to preserve all versions of an object, protecting against accidental overwrites or deletions and aiding in data recovery.

#### **6.1.4.5 Pre-Signed URLs and Cookies**

Use pre-signed URLs or cookies to grant temporary access to S3 objects, for time-limited access to private content.

#### **6.1.4.6 Monitoring and Alerts**

Amazon CloudWatch alarms and AWS CloudTrail to monitor and log S3 bucket activity, setting up alerts for suspicious or unauthorized actions.

### **6.1.5 Database Security**

#### **6.1.5.1 Data Encryption, Encryption at Rest**

Implement server-side encryption (SSE) to protect data at rest. SSE ensures that data stored in DynamoDB remains encrypted and secure.

#### **6.1.5.2 Access Control**

Leverage IAM policies and fine-grained access control to limit access to DynamoDB tables and items to authorized entities only.

#### **6.1.5.3 Auditing and Monitoring**

Implement Amazon CloudWatch to monitor and audit DynamoDB queries & transactions.

### **6.1.6 User Control in AWS**

#### **6.1.6.1 AWS Organizations – Centralized Account Management**

Centralized Platform for management and organization of AWS accounts, this enables users of the main AWS account to be grouped in OUs. The advantage of using OUs is to add fine-grained access control policies through SCPs. This ensures the consistency of security and compliance policies across the organization. Furthermore, through the use of AWS Organizations, RBACs can be leveraged to define granular permissions on the user level. RBACs can explicitly grant and deny specific resources or actions within each user account.

### **6.1.7 User Control in CyberTool**

User control, including authorization and authentication, will be done via Amazon Cognito. Amazon Cognito User Pools operates as a CIAM that offers secure identity storage and authenticator that can support login with social identity providers and SAML or OIDC-based identity providers. Users can register within CyberTool to access specific pages to the webapp.

### **6.1.8 Application Logging**

(See 6.1) AWS X-Ray can provide insights into the health and operation of services and resources in live use. Integrating X-ray with CloudWatch Logs can enable CloudWatch alarms to resolve issues based on trace X-ray traced data.

### **6.1.9 Application Log Storage**

(See 6.2)

### **6.1.10 Application Security**

All client-webapp communication will be encrypted via HTTPS. Cognito endpoints for authorization will be encrypted via HTTPS.

### **6.1.11 HTTPS Encryption**

Using Amazon CloudFront as a low latency content delivery network, the base requirement for client to server communication is SSL/TLS security certificate with TLS 1.3. The security certificate must be in X.509 format. The security certificate will be from the AWS Certificate Manager.