

DESIGN AND IMPLEMENTATION OF SERVER LESS WEB APPLICATIONS UTILIZING API GATEWAY

Dr.Koduru.Gouthami¹, Dr.K. Sunitha², Dr.U.Suneetha³ , M.Chandana⁴

^{1,4} Department of ECE, Malineni Lakshmaiah women's Engineering, Pulladigunta, Guntur, Andhrapradesh-522017.

² Department of ECE, Malineni Lakshmaiah women's Engineering, Pulladigunta, Guntur, Andhrapradesh-522017

³ *Department of Electronics, Sri Krishna Devaraya University, Anathapur, Andhra Pradesh-515003*

Abstract. A revolutionary development that has made it possible for enterprises to streamline the difficulties involved in providing, growing, and managing infrastructure is server less computing. The growing popularity of server-less solutions makes it difficult to choose the best designs from a plethora of professional suggestions. Through a multi vocal literature survey, this study classifies server-less patterns and highlights the advantages and difficulties associated with coordination, aggregation, event management, availability, communication, and validation, among other areas. We offer a case study using AWS Lambda, Amazon Dynamo DB, and other services to analyze real-time data streams for a ride-sharing company in order to illustrate the usefulness of server less computing. The report highlights the benefits of server-less computing and highlights how it can relieve developers of server-related worries, enabling them to concentrate on developing dependable and scalable applications. This work advances knowledge about server less computing function in the larger framework of cloud technologies and how it's applied in practical situations.

Keywords: Server less computing, Multi vocal, Cloud technologies, Scalable application

1. INTRODUCTION (11 PT, CAPITAL BOLD FACE, CENTERED)

The Using API Gateway to create a server less web application is a contemporary method of creating scalable and economical apps. With this configuration, you don't have to manage any servers because AWS API Gateway serves as the entry point for the backend services of your application. An overview of creating such an application is provided here. Developers may build and publish content more easily with the help of the fully managed AWS API Gateway service. At whatever size, keep an eye on, safeguard, and maintain APIs. Serving as an entry point for your backend services, it

takes care of all the work required to receive and process hundreds of thousands of API requests at once. Prior to putting into practice

2. METHODS FOR THE DEVELOPMENT:

Add-ons like Flexible Scaling, No Server Management, High Availability, and No Idle Capacity are provided to developers. To enable user interaction with the web application, we plan to develop a RESTful API through the use of Amazon API Gateway. These APIs can be used to start AWS Lambda Events that automate storage or even send data to queues for processing. Using Amazon Cognito for user authentication and AWS Amplify for HTML, CSS, and JavaScript hosting, you may host a web application on their virtual server. To implement the Lambda functions and API Gateway settings. Performance problems can be found and fixed after deployment with the use of monitoring and debugging tools such as CloudWatch. Last but not least, continuous optimization and scaling initiatives are necessary to guarantee the application stays effective, economical, and able to manage a range of workloads.

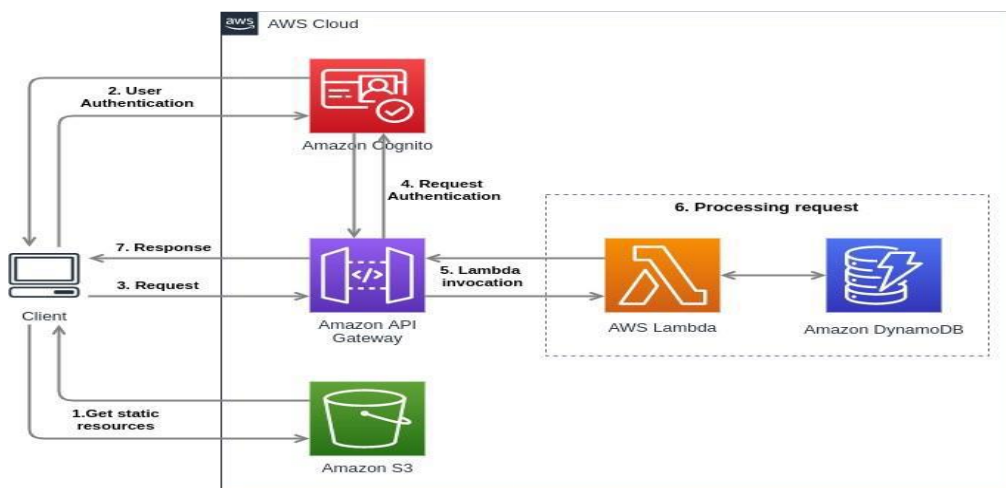
2.1. Input Design

When using API Gateway to create a server-less web application, there are a number of crucial factors to take into account. Determining the API endpoints, which outline the user interface of the program, is the first stage. After that, since each Lambda function in AWS Lambda comprises specific endpoint functionalities, developing Lambda functions is crucial. Thus, configuring the API Gateway is essential to building the API's framework, which consists of the routes, HTTP methods, and authentication schemes. When Lambda and API Gateway are combined, the backend functionality and the designated endpoints are seamlessly coupled. Authentication and authorization processes must be set up correctly, utilizing AWS services like Cognito or IAM, in order to grant safe access. Prior to deployment, the functionality of the API is confirmed by extensive testing using Postman software. CloudFormation or AWS services are examples of infrastructure-as-code solutions used in the actual deployment process.

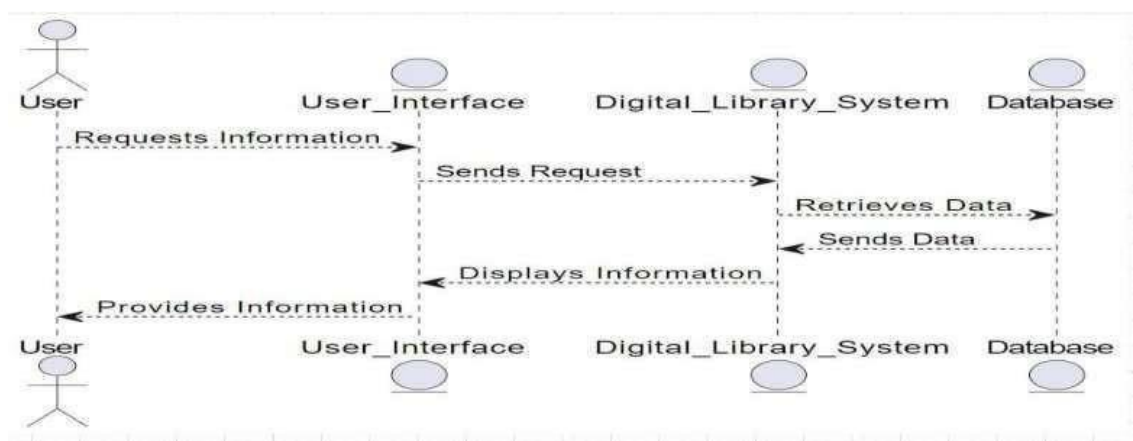
2.2. Output Design

Determining how data is shown to users or other systems within a software program or system is known as output design. It includes data, messages, reports, and any other type of output that needs to be formatted, organized, and delivered. A clear, efficient, and user-friendly output design takes into account the audience, purpose, content, and context of use in order to effectively communicate information. This covers creating user interfaces for reports, alerts, notifications, error messages, and any other output that the system produces. Usability, readability, accessibility, and user pleasure are all improved by effective output design. To guarantee efficient communication and interaction, it should take into account a variety of user preferences, device capabilities, and communication routes. Additionally, by presenting information in a meaningful and actionable way, output design should support decision-making, problem-solving, and action-taking.

2.3. System Architecture

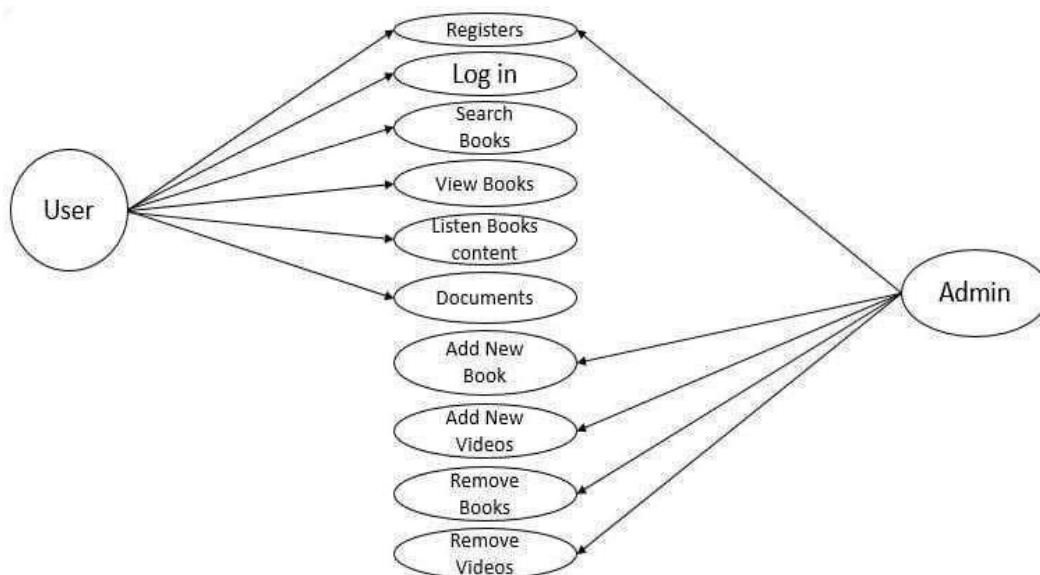


2.4. Data Flow Diagram



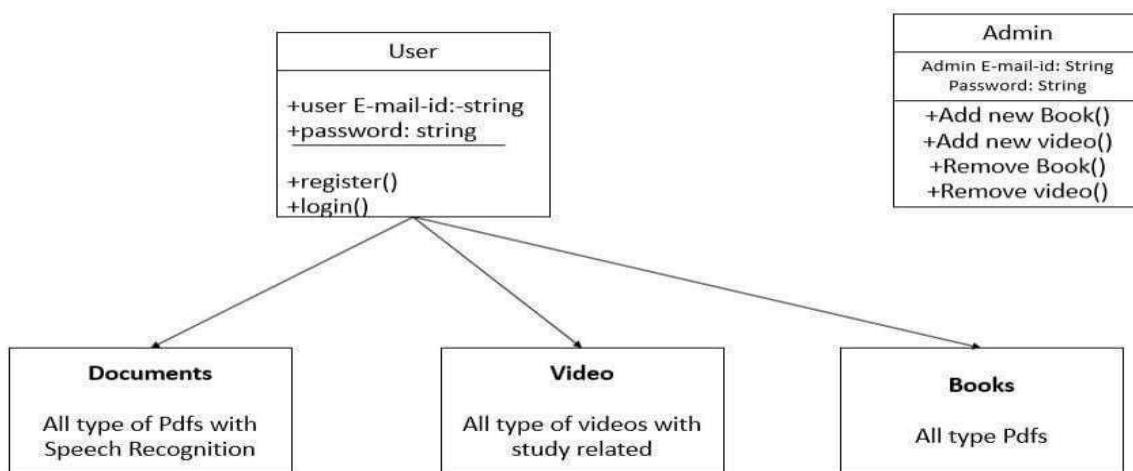
2.4 Use Case Diagrams

A server-less web application's use case diagram would show the several ways in which actors and the system interact. Users interacting with the application and other systems or services that the application interfaces with might both be considered actors in this scenario. Use cases would stand in for particular elements or functionalities of the program, like content delivery, data processing, and user authentication.



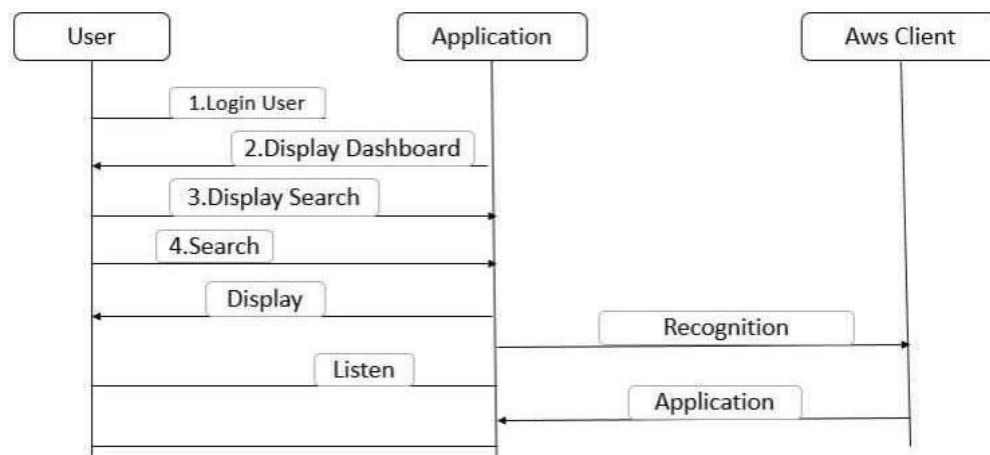
2.5 Class Diagram:

A class diagram, as used in software engineering, is a sort of static structural diagram in the Unified Modeling Language (UML) that illustrates a system's classes, attributes, operations (or methods), and interactions between the classes. It clarifies which class



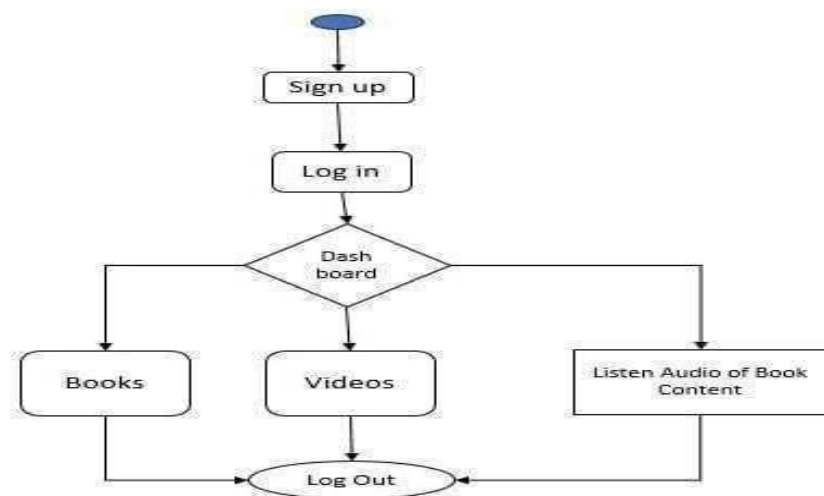
2.6 Sequence Diagram:

In the Unified Modeling Language (UML), a sequence diagram is a type of interaction diagram that illustrates the relationships and sequence in which processes operate with one another. It is a Message Sequence Chart construct. Event diagrams, event situations, and timing diagrams are other names for sequence diagrams.



2.6 Activity Diagram

Workflows of sequential activities and actions with provision for choice, iteration, and concurrency are represented graphically using activity diagrams. Activity diagrams in the Unified Modeling Language can be used to explain the step-by-step operational and business workflows of system components. An activity diagram displays the total control flow



3. RESULTS AND DISCUSSION (12 pt, CAPITAL BOLD FACE, CENTERED)

Module Description

3.1. User:

The first person can register. He needed a working user email to complete the registration process and for future correspondence. After the user has been activated by the admin, they can log in to our system. Based on our dataset's column matching, the user can add their own dataset. The format of the data must be float for the algorithm to

execute. Here, we used the dataset of traffic signs for cars. Users utilizing our Flask application can additionally contribute fresh data to the already-existing dataset. The user can click Classification on the webpage to see the accuracy and loss of the data calculated using algorithms.

3.2. Books:

The purpose of the Books Management System module is to make it easier for books to be arranged, cataloged, and found in a library or bookstore. It offers extensive capabilities for handling all elements of book-related duties, such as inventory control, circulation, cataloging, and customer relations.

3.3. Video:

Videos are pictures that have been captured visually, frequently with music added. They are a well-liked type of multimedia content that can be utilized for marketing, education, communication, and entertainment. Videos can be short clips or full-length films, covering a broad range of subjects and genres. They can also differ in length, content, and structure. Usually, they are watched on a variety of media, including social media, websites, streaming services, and television. Technology has made it easier and more accessible to create, share, and watch videos, which is why they are becoming a necessary component of modern communication and entertainment.

3.4. Audio

An extra narration track called audio description (AD) is mainly meant for blind and visually impaired people who watch visual media, such as movies, television shows, and other multimedia presentations. It gives a spoken explanation of visual elements that are difficult to make out from the primary audio track alone, like actions, places, costumes, facial expressions, and scene transitions.

RESULTS

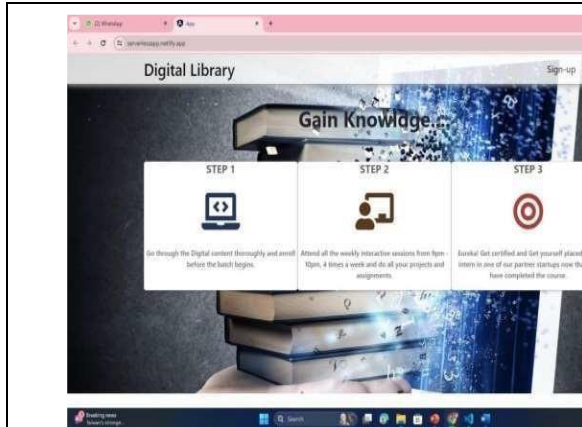


Figure 1 Home page

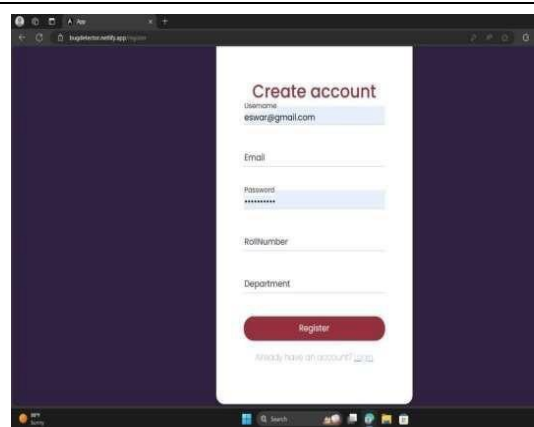


Figure 2 Registration

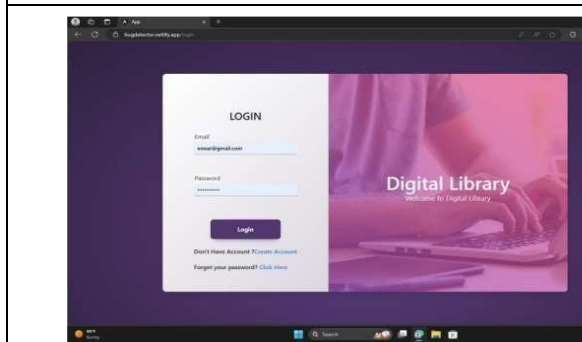


Figure 3 Admin login page

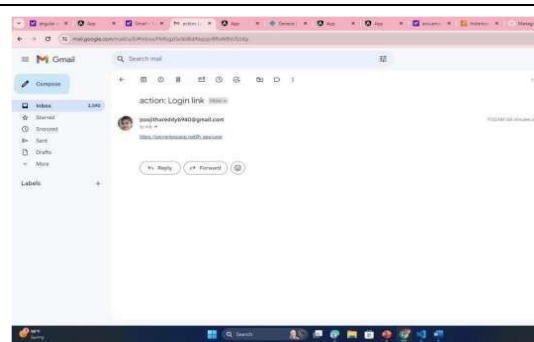


Figure 4 Mail link page

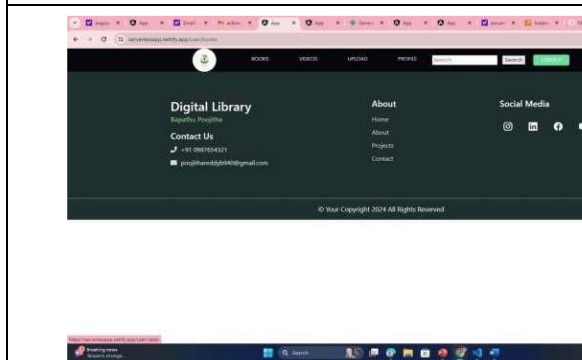


Figure 5 Dash board

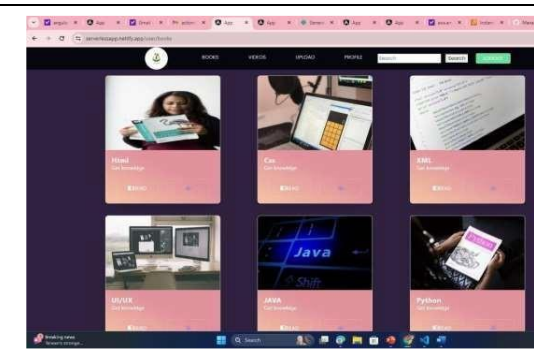


Figure 6 Books Page

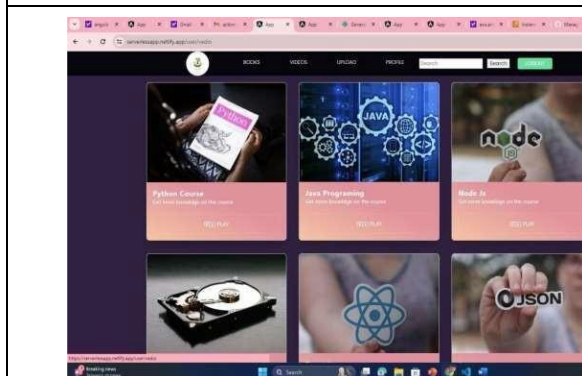


Figure 7 Video Page

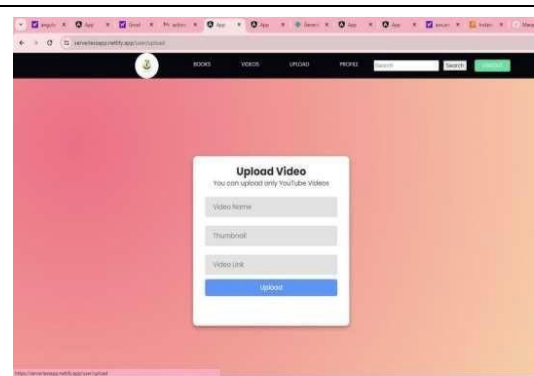
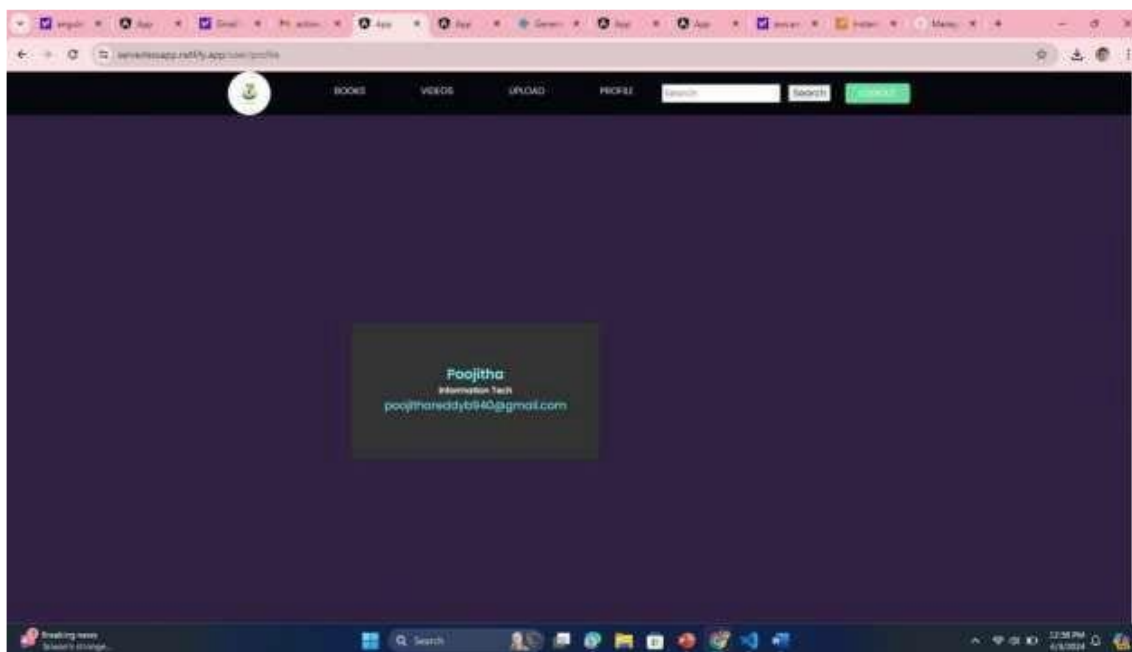


Figure 8 Up load Page



4. CONCLUSIONS (12 pt, CAPITAL BOLD FACE, CENTERED)

A revolutionary paradigm known as server less computing has surfaced, providing enterprises with a way to simplify infrastructure management. The growing popularity of server-less architectures demands that solutions be carefully chosen from a variety of professional suggestions. Through a thorough assessment of the literature, this research classifies server-less patterns and clarifies issues related to coordination, aggregation, event management, availability, communication, and validation. Through a case study involving AWS Lambda and Amazon Dynamo DB, the actual implementation of server less computing is illustrated, demonstrating its ability to free developers from server-related issues and allow the creation of scalable and dependable applications

REFERENCES

1. Sewak and S. Singh, "Winning in the Era of Serverl ess Computing and Function as a Service," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, 2018, pp. 1-5
2. Lynn, P. Rosati, A. Lejeune and V. EMEA aroha, "A Preliminary Review of En temporise Server less Cloud Computing (Function-as-a-Service) Platforms," 2017 IEEE International Conference on Cloud Computing Technology and Science (Clouds), Hong Kong, 2017, pp. 162-169.
3. Swedha and T. Dubey," Analysis of Web Authentication Methods Using Amazon Web Services," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-6.
4. Lloyd, S. Ramesh, S. Chhatrapati, L. Ly and S. Pallickara," Serverless Computing: An Investigation of Factors Influencing Microservice Performance," 2018 IEEE International Conference on Cloud Engineering, Orlando, FL, 2018, pp. 159-169.
5. Al-Ali et al.," Making Serverless Computing More Serverless," 2018 IEEE 11th International Conference on Cloud Computing, San Francisco, CA, 2018, pp. 456-459.

6. Kritikos and P. Skrzypek, "A Review of Server less Frameworks," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion, Zurich, 2018, pp. 161-168.

7. Kotas, T. Naughton and N. Imam, "A comparison of Amazon Web Services and Microsoft Azure cloud platforms for high performance computing," 2018 International Conference on Consumer Electronics, Las Vegas, NV, 2018, pp. 1-4. [7] Yoon, A. Gavrilovska, K. Schwan and J. Donahue, "Interactive Use of Cloud .