

Building Infrastructure using Cloud Interface

Sneha Mahajan, Riya Tattu, Ashish Bhoir,
Rizwana Shaikh.

Abstract: Today's Generation is very advanced when it comes to technologies. Everything can be achieved easily if right technology is used in correct manner and at right place. The most used technologies used in industries are Machine learning, Cloud Computing, Robotics, Data science and IOT etc. The on-demand supply of IT resources over the internet with pay-per-use fees is known as cloud computing. As an alternative to acquiring and maintaining computer gear and software, you can pay to use a cloud computing service. The cloud computing provides various services which helps the developers to develop projects and functionalities with ease. They provide various operating systems and different networks and servers to be used by paying cost of it. There are three famous and mostly widely used cloud services: Infrastructure as a service-"Infrastructure as a Service (IaaS)," another name for cloud infrastructure services, provides computer infrastructure as a service, often a platform virtualization environment. Software as a service. Amazon Web Services, Google Cloud Platform, and Microsoft Azure all offer these cloud services. This project is also based on cloud services that is creating a website through which a user can easily start and stop their instances available on the google platforms. Here we are working with two types of cloud platforms, Amazon web services and Google cloud platform. These instances can be easily start and stopped according to the project working on by users individually or can directly start or stop the project itself which will contain their respective instances.

I. INTRODUCTION

The term "cloud services" refers to a wide range of services that are made readily available to customers and organizations online. These services were developed to provide easy access to resources and programs without requiring a user's own hardware or infrastructure.

Whether they are aware of it or not, most employees use cloud services during the workday for everything from checking email to collaborating on documents. Cloud service management is solely the responsibility of cloud computing vendors and service providers. Apps are made available to customers via the servers of the providers; thus, a company is not required to host them on its own internal servers. This calls for computation autoscaling, which essentially comes in two flavours. Both schedule-based and metric-based

autoscaling Measure-based autoscaling, which is an autoscaling action that is performed when a performance metric reaches a certain threshold, is the first kind on which we are concentrating in this project. In this we are creating an infrastructure that will help the user to operate their instances all from one place. They will be able to start and stop their instances that are present on cloud platforms like AWS and GCP. This brings great convenience and ease at work as it avoids hopping on each time back and forth to start and stop the instances on both the platforms.

1.2 MOTIVATION

This project's primary goal is to address the problem of manually starting the system through a respective infrastructure team member, which entails six to seven setups and thus takes more time and is not an efficient method. Since all of the virtual machines in this configuration are on cloud platforms like Amazon Web Services and Google Cloud Platform, each of which has its own fees, occasionally leaving the system on (without turning off the virtual setup) would result in an overcharge. Therefore, we made the decision to create an effective interface that any non-technical team member (authorized student) could utilize to instantly start and stop the setup. Simple UI design will ensure that utilization is at its highest level. Management of the same is also an important point to be taken into consideration so admin and user both should be effectively able to use this website to manage all project instances without any confusion by stating and designing all the factors precisely

1.3 PROBLEM STATEMENT

The current system manually controls cloud start and stop functionalities for different operating systems and services as needed. A non-technical individual finds it challenging to do this and must rely on another person, which can occasionally delay things and be annoying. For the same purpose, a straightforward GUI must be developed, allowing users to simply start and stop services with a single button click. Additionally, a record of the incident must be kept.

1.4 Objectives

- Development of a straightforward GUI.
- To construct instances, combine all the services used into a single list.
- Starting and terminating services with a single button click
- To handle all user responsibilities and to assign roles of specific instances to be worked on
- Managing all these instances with respect to project and using its service by both admin and user

1.5 Organization of the Report

The report is divided in specific sections. The abstract of the project is mentioned in the beginning of the report. We have also included the report. We have also included the abbreviations used in the report. Coming to the further three sections, the first section contains introduction of the project, motivation to choose this topic, problem statement and objectives. The next section consists of a survey of the existing systems in a tabular form. We have also included the limitations of the existing systems. Further section gives details regarding our proposed system which includes the framework of the project along with algorithms followed. Flowcharts are also added for better understanding. Software details and results of projects are displayed. The report is concluded with future work. Lastly, we have mentioned a few references regarding our topic.

2.1 Survey of Existing System

G.Ranjith (2018)	<i>Metering services of the cloud is deployed for the users to obtain knowledge about their resource consumption and to cross verify their consumption as well as the details about consumption amount they have made at the particular time period. As per the current situation customer prefer to purchase infrastructure from different vendors to avoid vendor lock-in problem and backup, security purpose. Proposed model focus on preparing a consolidated report comprises of deeper usage details irrespective of vendor models to achieve better satisfaction from the client side.</i>
-------------------------	--

<i>Author</i>	<i>Description</i>
P.Nagendra Babu	<i>According to the study report, cloud computing is a model for providing a convenient on-demand network. Servers, software, data center space, and networking hardware are all totally outsourced services that the client purchases rather than being purchased outright. Open-source Cloud application Eucalyptus that works with EC2 (allows to set up local cloud infra prior to buying services).</i>
Joel Scheuner, (2020)	<i>Function-as-a-Service (FaaS) is a new serverless cloud computing hardware provisioning model that enables developers to essentially be completely abstracted from hardware concerns, according to a published paper titled Function-as-a-Service (FaaS) Infrastructures and Implications for Software Developers. Three different FaaS providers were examined (AWS Lambda, Google Cloud Functions and Microsoft Azure Functions)</i>

2.2 Limitation Existing system or research gap

The work on Cloud computing highlights the absence of a comprehensive scheduling mechanism for cloud computing, which hinders the provision of tailored quality of service (QoS) levels for different clients. In this, aptly titled Management of API Gateway Based on Micro-service Architecture, the authors point out the risk of complexity in having all API rules concentrated in one location without a fixed scale. Other papers also identify several restraints in cloud computing, such as metering services that are limited in their ability to provide detailed.

3. Proposed System

3.1 Introduction

In this UI model the user will login through his/her credentials also by specifying the instance id and region name then when user will click on start button, internally JavaScript on click function will work which will call through API gateway which will connect lambda function through web servers. This will start the specific service instance created and then the user can work with it respectively. After the completion of work the user will click on stop button which in turn will stop the current working services/instances. When the user successfully login to the system he/she will be directed to main dashboard which will present all the current working project instances the user has been allotted. The admin has all the rights to allocate these instances link to the users through his login credentials and can add users also can delete the instances link from any user. Here user can directly stop and start a particular instance or can also stop and start the project which includes all the instances the user is working on in that project, this is very effective for both user to save time and admin as well to maintain and manage these records.

We used two concepts to build this project: AWS Lambda, a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you, and terraform, a cloud service that enables infrastructure automation for provisioning, compliance, and management of any cloud, datacenter, and service. An application programming interface (API) gateway is software that accepts a request from a user of an application, routes it to one or more backend services, gathers the necessary data, and then delivers it to the user in a single, combined package. Additionally, it offers analytics, several levels of threat defense, and other security for the application.

3.2 ARCHITECTURE/ FRAMEWORK

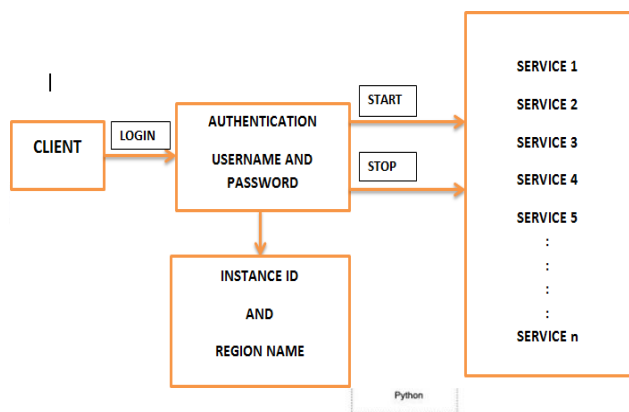


Fig. 1. Flow Diagram



Fig. 2 GCP working diagram

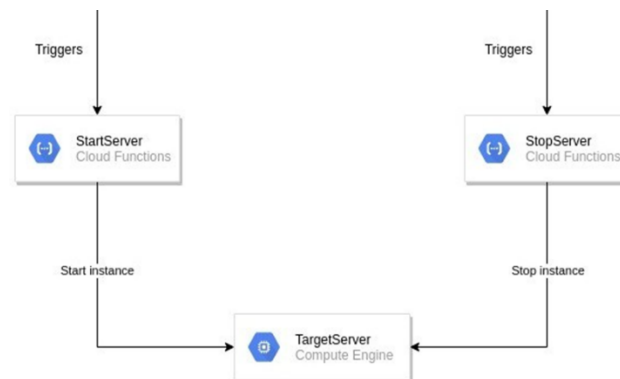


Fig. 3 working diagram

common process in both services is that we have passed the instance ID and region name according to the required system (virtual machines)

- 1) AWS- to start and stop virtual machines we have used lambda function which is in python language (boto3 library), then lambda function through http request is accessed and for that we have used API gateway which is coded in python.
- 2) GCP-The Google API Client Library for Python is designed for Python client-application developers, then we have also used cloud function, interface is simple (API trigger)

3.3 METHODOLOGY AND FLOWCHART

The methodology of the system involves the following flow of how the whole system process takes place.

1. In order to create or oversee instances, users commence by inputting their valid login credentials, which, upon successful authentication, direct them to the main dashboard. These logins encompass both administrators and employees, with the corresponding IDs managed by administrators.

Published by :

<http://www.ijert.org>

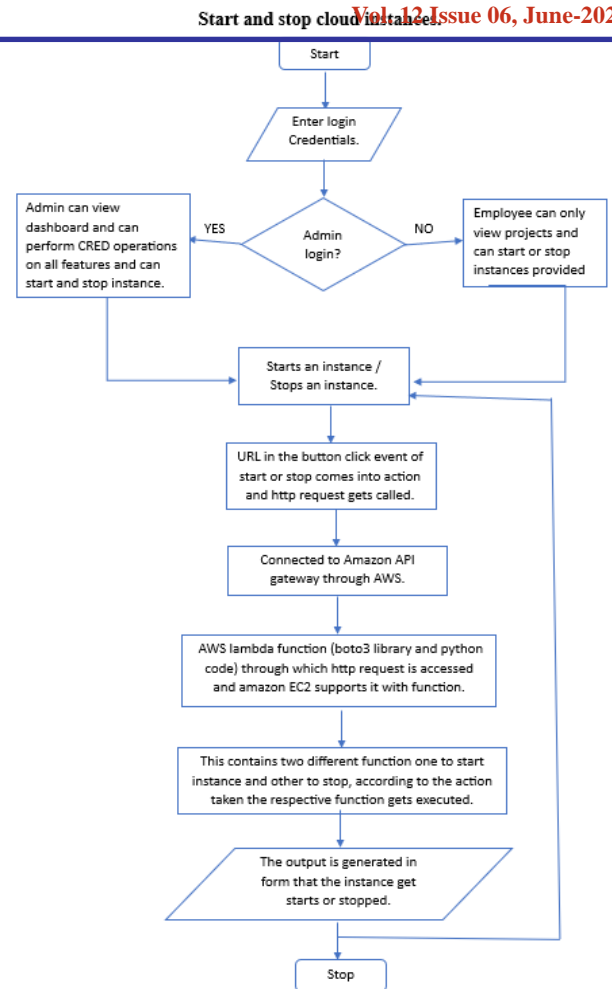
2. If an administrator logs in, they are granted the ability to view and modify all functions, including all CRED operations related to projects, instances, and user IDs. Furthermore, they can allocate specific users to main projects and their associated instances based on their respective roles and responsibilities.

3. Upon an employee's login, they gain access exclusively to the projects they are currently engaged in and are presented with the option to initiate or terminate instances.

4. When a user or administrator commences or halts a particular instance, the backend triggers the respective start or stop function by utilizing the URL specified within the cloud service. This cloud service URL is connected to Amazon Web Services (AWS) through the Amazon API Gateway service. AWS Lambda functions, which execute code in response to events and autonomously manage the necessary computing resources, are employed for this purpose. The code is developed using the Python language (utilizing the boto3 library) and is accessed through an HTTP request to the Lambda function. The start or stop action is performed by leveraging Amazon EC2, a service intended for long-term, stable operations such as website hosting and high-performance computing.

5. Similarly, upon initiation or termination of a particular instance by a user or administrator, the start button's click event triggers the action specified within the URL, which is connected to the cloud service URL associated with Google Cloud Platform (GCP). The Google API Client Library, designed for Python client-application developers, is utilized for this purpose. Additionally, cloud functions, which are triggered by API calls, are employed. This process mirrors the functionality of AWS, executing the specified function upon triggering, resulting in the initiation or termination of the instance.

6. This system manages the operation of a collection of instances belonging to a specific project and consolidates them into a unified project. Starting and stopping instances within this project can be accomplished using a single button.



3.4 EXPERIMENTS AND RESULTS

Some of the screenshots of the implementation of the system has been displayed in this section.

The proposed system, “Build Cloud infrastructure using UI” is capable of managing multiple AWS and GCP instances by starting and stopping the instances. Also, they can manage the users access and capable of assigning roles based on requirements.



Fig. 4. Login Page

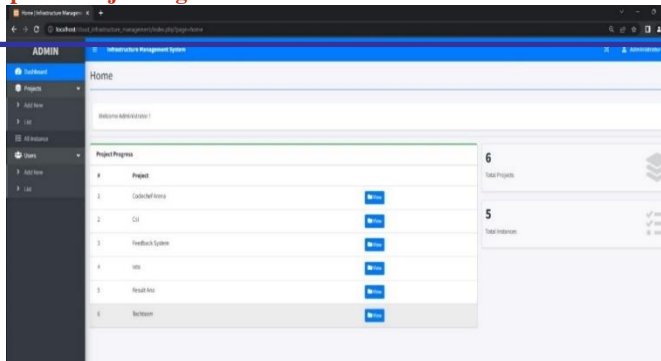


Fig. 5. Admin Panel



Fig. 9. Edit instance option for admin

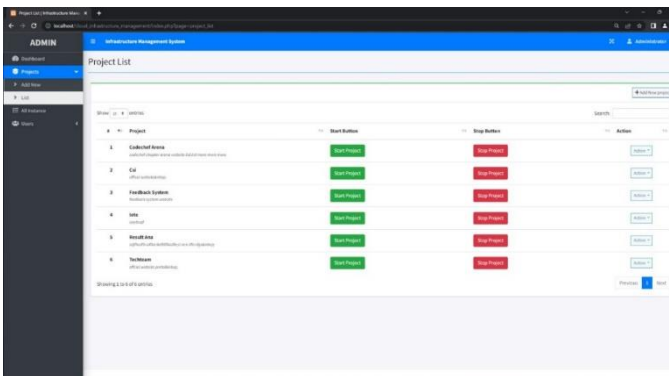


Fig. 6. List of all Projects

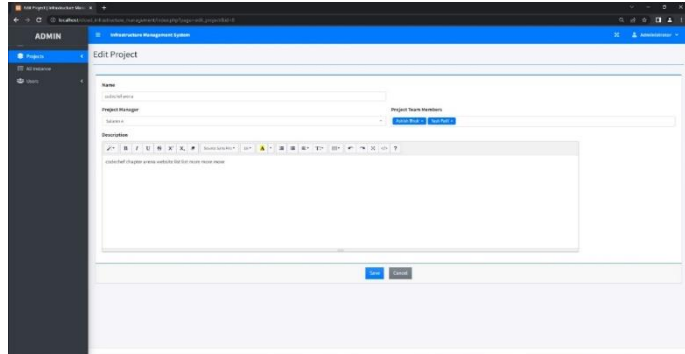


Fig. 10. Project edit options for admin with CRED Operations

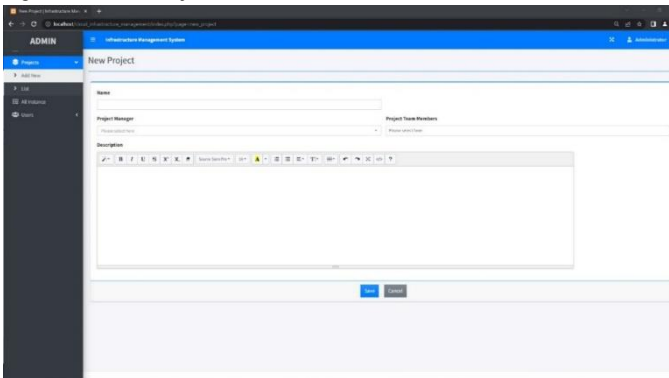


Fig. 7. Create new Project

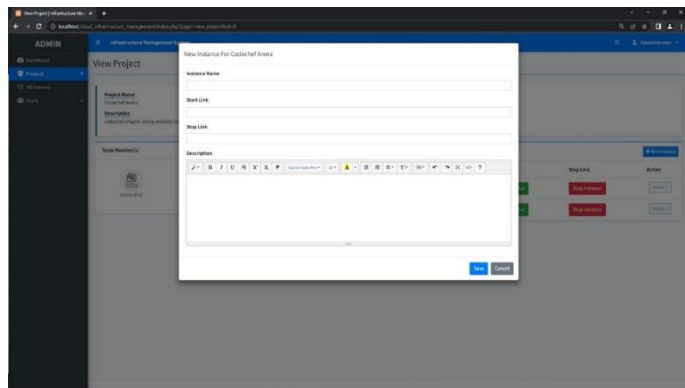


Fig. 11. Create new instance under project option for admin.

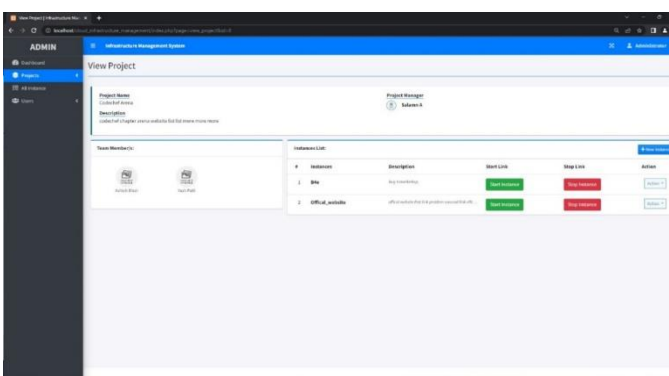


Fig. 8. View project details with individual button to start and stop instance with project member manager description details for admin to edit delete add option

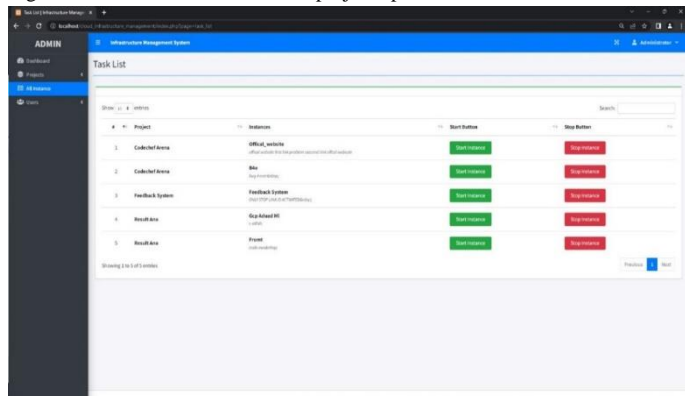


Fig. 12. List of all instances

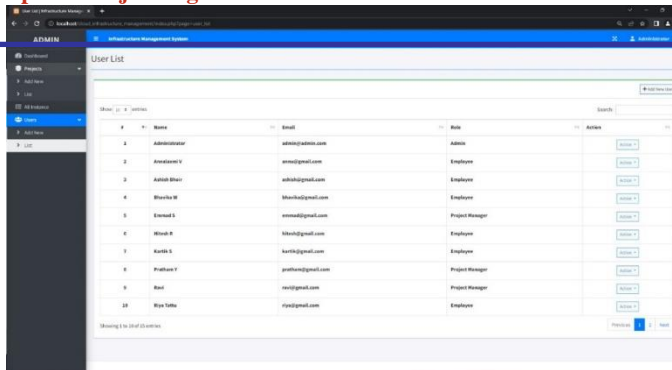


Fig. 13. List of all members including team manager and team member with their role and add, update, delete and access data for admin.

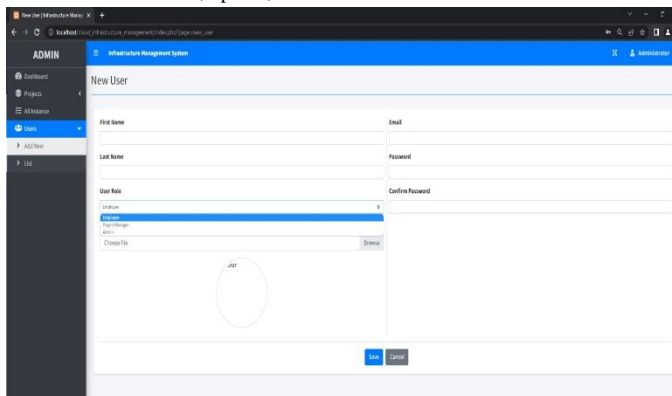


Fig. 14. Create new user with their role.

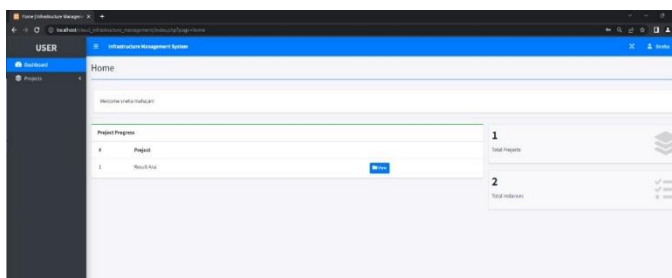


Fig. 15. User dashboard

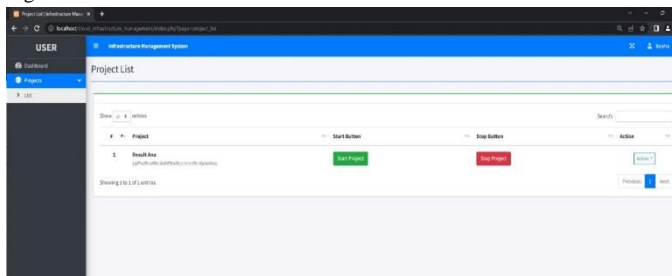


Fig. 16. User's assigned project list with restricted action to the project according to their roles and responsibilities

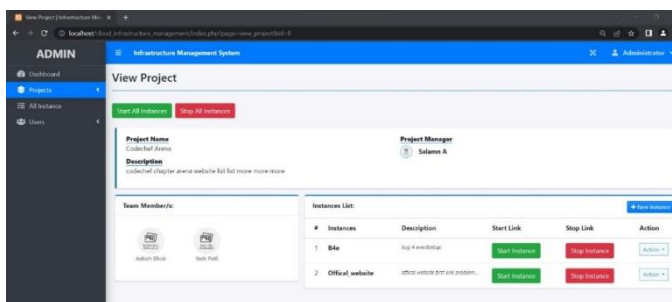


Fig. 17. Details of project (view form user panel)

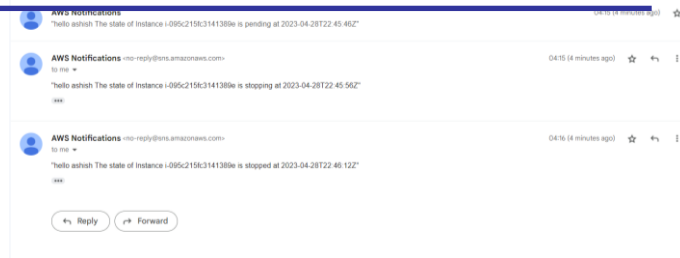


Fig. 18. Email Notification once an instance is created

3.5 FUTURE WORK

Cloud computing technology is one of the latest technologies that is widely used for development of projects. This brings a lot of convenience and improves the functionality of the project. In this project, we have developed a web application that is used to manage the operation of a set of instances of a particular project. This merges all the set of instances.

The instance can be started and stopped by using a single button. The developed application has been studied detail, Reviewing various research papers and blogs. The flow of the project is understood using data flow diagrams. All the statements and studies have been stated with proper references. Overall, it reduces the effort and time of the clients. This system fulfills all the requirements of the clients adequately. Along with that the services on both platforms i.e., on AWS and GCP can be merged for a project. Multiple projects can be displayed and operated using this functionality at the same time.

3.6 CONCLUSION

The project is developed with the help of the latest technology. The project has successfully met all the requirements of the clients accurately. The project can be expanded further developing into an advanced system for increasing the speed as well adding more functionalities making the website more efficient. The existing system has a lot of scope to be expanded. The developed project has limited functionalities for admin and user. The possible additions that can be made to the system are: A record can be displayed for admin where details of the user will be present who has initiated the service and the time when they started and ended it. The status of starting or stopping an instance can be displayed on the page when clicked on the button. These are a few specifications and along with this the system is flexible to acquire the requirements.

REFERENCES

- [1] A Handy Guide to Using Agile Methodology in Testing: Processes, Best Practices & Tools. (2022).
- [2] Abrahamsson, P. ((2017 September 25)). Agile Software development Methods.
- [3] Daeyong Jung, S. C. (2011). An Efficient Checkpointing Scheme Using Price History of Spot Instances in Cloud Computing

Published by :

<http://www.ijert.org>

Environment. Proceedings of the 8th IFIP international conference on Network and parallel computing.

- [4] Dharmapuri. (2020, May). Cloud computing review. International Research Journal of Innovations in Engineering and Technology, Vol. 4, Iss. 5,66-69.
- [5] Francisco Cocozza, G. L. (march 2015). Cloud Management Platform Selection: A Case Study in a University Setting. Sixth International Conference on Cloud Computing, GRIDs, and Virtualization At: Nice, France, 978-1-61208-388-9.
- [6] G.Ranjith, M. (2018, January). A unified approach for effective use of cloud metering service. International-Journal-of-Pure-and-Applied-Mathematics.
- [7] J T ZHAO, S. Y. (2018, September). Management of API Gateway Based on Micro- service Architecture. Journal of physics conference series.
- [8] Joel Scheuner, P. L. (2020, December). Function-as-a-Service performance evaluation : A multivocal literature review. journal of systems and software, Volume 170.
- [9] Joel Scheuner, P. L. (2020, December). Function-as-a-Service performance evaluation: A multivocal literature review. journal of systems and software, volume 170.
- [10] P.Nagendra Babu, M. K. (n.d.). Infrastructure-as-a-service.
- [11] Why are we moving to agile software development . (2019, April)
- [12] <https://orionadvisortech.com/blog/why-were-moving-to-agile-software-development/?print=print>