A Project Report on

GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

SUBMITTED TO OSMANIA UNIVERSITY FOR THE PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS

BY
TUMU SRIKANTH
(130522862091)

UNDER THE GUIDANCE OF
MRS. ANNAPURNA VEGE
Assistant Professor

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

AVANTHI PG COLLEGE

(AFFILIATED TO OSMANIA UNIVERSITY & RECOGNIZED BY AICTE)
MOOSARAMBAGH, DILSUKHNAGAR, HYDERABAD-500036

2022-2024

MASTER OF COMPUTER APPLICATIONS
AVANTHI PG COLLEGE

(AFFILIATED TO OSMANIA UNIVERSITY&RECOGNIZED BY AICTE)

MOOSARAMBAGH, DILSHUKNAGAR, HYDERABAD-500036.

CERTIFICATE

THIS IS TO CERTIFY THAT THE PROJECT **GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS** IS A BONAFIDE WORK CARRIED OUT BY **TUMU SRIKANTH (130522862091)** IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF **MASTER OF COMPUTER APPLICATIONS** FROM AVANTHI PG COLLEGE, AFFILIATED TO OSMANIA UNIVERSITY, HYDERABAD, UNDER GUIDANCE AND SUPERVISION.

INTERNAL GUIDE	HEAD OF THE DEPARTMENT
MRS. ANNAPURNA VEGE	**MR. P. NAVEEN CHANDRA**
ASST. PROFESSOR,	ASST. PROFESSOR,
AVANTHI PG COLLEGE.	AVANTHI PG COLLEGE.

Date:________________

CERTIFICATE

This is to certify that the project work **GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS** at **Manac Infotech** has been successfully carried out by **TUMU SRIKANTH**, a student of MCA II year bearing Hall Ticket No. **1305-22-862-091** in partial fulfillment to award then degree of MASTER OF COMPUTER APPLICATIONS in Osmania University for the academic year 2022 - 2024.

Project Guide Head of Dept External examiner

MANAC
Infotech (P) Limited

Ref: MANAC/PROJ/23-24/D2009
JULY 03, 2024

PROJECT COMPLETION CERTIFICATE

This is to certify that **Mr. TUMU SRIKANTH (1305-22-862-091)** has successfully completed the Project on **"GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS"** in Partial fulfillment of the **MCA (Master of Computer Applications)** course during the period **APRIL 01, 2024 TO JULY 31, 2024.**

SOFTWARE: PYTHON

We wish him/her good luck in all future endeavors.

Project Manager

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and indebtedness to my project guide **MRS. ANNAPURNA VEGE** for her valuable suggestions and interest throughout the course of this project.

I am also thankful to our principal **Dr.S.Apparao** and **Mr.P.Naveen Chandra**, Head of Department, Master of Computer Applications, Avanthi P.G College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our course.

My sincere thanks to my project coordinator **Mr. P. Naveen Chandra** for given valuable suggestions to fulfilment of this project and I convey my thanks to the lab staff for allowing me to use the required equipment whenever needed.

I sincerely acknowledge and thank all those who have directly / in-directly given their support in completion of this Project work.

TUMU SRIKANTH

(130522862091)

DECLARATION

This is certify that the project report on **GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS** is a record of bonafide work done by me in the department of Master of Computer Applications, Avanthi P.G College, Osmania University. The reports are based on the project work done entirely by me and not copied from any other source.

The results embodied in this project report have not been submitted to any other institute or institute for the award of any degree or diploma to the best of my knowledge and belief.

TUMU SRIKANTH

(130522862091)

INDEX

LIST OF FIGURES

GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

ABSTRACT

Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone .To alleviate this problem, in this paper we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. The cloud monitors contain contracts used to automatically verify the implementation. We use Django web framework to implement cloud monitor and OpenStack to validate our implementation.

CHAPTER 1
INTRODUCTION

In many companies, private clouds are considered to be an important element of data center transformations. Private clouds are dedicated cloud environments created for the internal use by a single organization . According to the Cloud Survey 2017 ,private clouds are adopted by 72% of the cloud users, while the hybrid cloud adoption (both public and private) accounts for 67%. The companies, adopting private clouds, vary in size from 500 to more than 2000 employees. Therefore, designing and developing secure private cloud environments for such a large number of users constitutes a major engineering challenge. Usually, cloud computing services offer REST APIs (REpresentational State Transfer Application Programming Interface) to their consumers. REST APIs, e.g., AWS, Windows Azure , OpenStack , define software interfaces allowing for the use of their resources in various ways. The REST architectural style exposes each piece of information with a URI, which results in a large number of URIs that can access the system. Data breach and loss of critical data are among the top cloud security threats . The large number of URIs further complicates the task of the security experts, who should ensure that each URI, providing access to their system, is safeguarded to avoid data breaches or privilege escalation attacks. Since the source code of the Open Source clouds is often developed in a collaborative manner, it is a subject of frequent updates. The updates might introduce or remove a variety of features and hence, violate the security properties of the previous releases. It makes it rather unfeasible to manually check correctness of the APIs access control implementation and calls for enhanced monitoring mechanisms. In this paper, we present a cloud monitoring framework that supports a semi-automated approach to monitoring a private cloud implementation with respect to its conformance to the functional requirements and API access control policy. Our work uses UML (Unified Modeling Language) models with OCL (Object Constraint Language) to specify the behavioral interface with security constraints for the cloud implementation. The behavioral interface of the REST API provides an information regarding the methods that can be invoked on it and pre- and post-conditions of the methods. In the current practice, the pre- and post-conditions are usually given as the textual descriptions associated with the API methods. In our work, we rely on the Design by Contract (DbC) framework , which allows us to define security and functional requirements as verifiable contracts. Our methodology enables creating a (stateful) wrapper that emulates the usage scenarios and defines security-enriched behavioural contracts to monitor cloud. Moreover, the proposed approach also facilitates the requirements traceability by ensuring the propagation of the security specifications into the code. This also allows the security experts to observe the coverage of the security requirements during the testing phase. The approach is implemented as a semi-automatic code generation tool in Django – a Python web framework – and validated using OpenStack as a case study. OpenStack is an open source cloud computing framework providing IaaS (Infrastructure as a Service) . The validation using OpenStack has shown promising results and motivates us to continue the tool development described in this paper. The paper is organized as follows: section II motivates our work. Section III gives an overview of our cloud monitoring framework. In section IV, we present our design approach to modelling stateful REST services. The contract generation mechanism is described in section V. Section VI presents the tool architecture and our work with monitoring OpenStack. The related work and the conclusion are presented in sections VII and VIII correspondingly.

CHAPTER 2
LITERATURE SURVEY

TITLE: Model driven security for web services.

ABSTRACT: Model driven architecture is an approach to increase the quality of complex software systems based on creating high level system models that represent systems at different abstract levels and automatically generating system architectures from the models. We show how this paradigm can be applied to what we call model driven security for Web services. In our approach, a designer builds an interface model for the Web services along with security requirements using the object constraint language (OCL) and role based access control (RBAC) and then generates from these specifications a complete configured security infrastructure in the form of Extended Access Control Markup Language (XACML) policy files. Our approach can be used to improve productivity during the development of secure Web services and quality of resulting systems.

TITLE: Run-time generation, transformation, and verification of access control models for self-protection.

ABSTRACT: A self-adaptive system uses runtime models to adapt its ar-chitecture to the changing requirements and contexts. How-ever, there is no one-to-one mapping between the require-ments in the problem space and the architectural elements in the solution space. Instead, one refined requirement may crosscut multiple architectural elements, and its realization in volves complex behavioral or structural interactions manifested as architectural design decisions. In this paper we pro-pose to combine two kinds of self-adaptations: requirements-driven self-adaptation, which captures requirements as goal models to reason about the best plan within the problem space, and architecture-based self-adaptation, which cap-tures architectural design decisions as decision trees to search for the best design for the desired requirements within the contextualized solution space. Following these adaptations, component-based architecture models are reconfigured using incremental and generative model transformations. Com-pared with requirements-driven or architecture-based approaches, the case study using an online shopping bench-mark shows promise that our approach can further improve the effectiveness of adaptation (e.g. system throughput in this case study) and offer more adaptation flexibility.

TITLE: Towards development of secure systems using umlsec.

ABSTRACT: We show how UML (the industry standard in object-oriented modelling) can be used to express security requirements during system development. Using the extension mechanisms provided by UML, we incorporate standard concepts from formal methods regarding multi-level secure systems and security protocols. These definitions evaluate diagrams of various kinds and indicate possible vulnerabilities. On the theoretical side, this work exemplifies use of the extension mechanisms of UML and of a (simplified) formal semantics for it. A more practical aim is to enable developers (that may not be security specialists) to make use of established knowledge on security engineering through the means of a widely used notation.

TITLE: Cloud computing the business perspective

ABSTRACT: The evolution of cloud computing over the past few years is potentially one of the major advances in the history of computing. However, if cloud computing is to achieve its potential, there needs to be a clear understanding of the various issues involved, both from the perspectives of the providers and the consumers of the technology. While a lot of research is currently taking place in the technology itself, there is an equally urgent need for understanding the business-related issues surrounding cloud computing. In this article, we identify the strengths, weaknesses, opportunities and threats for the cloud computing industry. We then identify the various issues that will affect the different stakeholders of cloud computing. We also issue a set of recommendations for the practitioners who will provide and manage this technology. For IS researchers, we outline the different areas of research that need attention so that we are in a position to advice the industry in the years to come. Finally, we outline some of the key issues facing governmental agencies who, due to the unique nature of the technology, will have to become intimately involved in the regulation of cloud computing.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In many companies, private clouds are considered to be an important element of data center transformations. Private clouds are dedicated cloud environments created for the internal use by a single organization. Therefore, designing and developing secure private cloud environments for such a large number of users constitutes a major engineering challenge. Usually, cloud computing services offer REST APIs (REpresentational State Transfer Application Programming Interface) to their consumers. The REST architectural style exposes each piece of information with a URI, which results in a large number of URIs that can access the system.

DISADVANTAGES:

- Data breach and loss of critical data are among the top cloud security threats.
- The large number of URIs further complicates the task of the security experts, who should ensure that each URI, providing access to their system, is safeguarded to avoid data breaches or privilege escalation attacks.
- Since the source code of the Open Source clouds is often developed in a collaborative manner, it is a subject of frequent updates. The updates might introduce or remove a variety of features and hence, violate the security properties of the previous releases.

3.2 PROPOSED SYSTEM

We present a cloud monitoring framework that supports a semi-automated approach to monitoring a private cloud implementation with respect to its conformance to the functional requirements and API access control policy. Our work uses UML (Unified Modeling Language) models with OCL (Object Constraint Language) to specify the behavioral interface with security constraints for the cloud implementation. The behavioral interface of the REST API provides an information regarding the methods that can be invoked on it and pre- and post-conditions of the methods. In the current practice, the pre- and post-conditions are usually given as the textual descriptions associated with the API methods. In our work, we rely on the Design by Contract (DbC) framework, which allows us to define security and functional requirements as verifiable contracts.

ADVANATGES:

- Our methodology enables creating a (stateful) wrapper that emulates the usage scenarios and defines security-enriched behavioural contracts to monitor cloud.

- The proposed approach also facilitates the requirements traceability by ensuring the propagation of the security specifications into the code. This also allows the security experts to observe the coverage of the security requirements during the testing phase.
- The approach is implemented as a semi-automatic code generation tool in Django a Python web framework.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Functional requirements are specifications that define the specific behavior or functions of a system, software, or product. They describe what the system should do, its features, and how it should perform under certain conditions. In the context of software development, functional requirements serve as a blueprint for the developers, guiding them in building a system that meets the needs and expectations of the users and stakeholders.

- USER
- ADMIN

4.2 NON- FUNCTIONAL REQUIREMENTS

4.2.1 HARDWARE REQUIREMENTS

➢ Processor	-	Pentium –IV
➢ RAM	-	4 GB (min)
➢ Hard Disk	-	20 GB
➢ Key Board	-	Standard Windows Keyboard
➢ Mouse	-	Two or Three Button Mouse
➢ Monitor	-	SVGA

4.2.2 SOFTWARE REQUIREMENTS

❖ Coding Language	:	Python.
❖ Operating system	:	Windows 7 Ultimate.
❖ Front-End	:	Python.
❖ Back-End	:	Django-ORM
❖ Designing	:	Html, CSS, JavaScript.
❖ Data Base	:	MySQL.

CHAPTER 5
SYSTEM STUDY

5.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

5.2 FEASIBILITY ANALYSIS

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

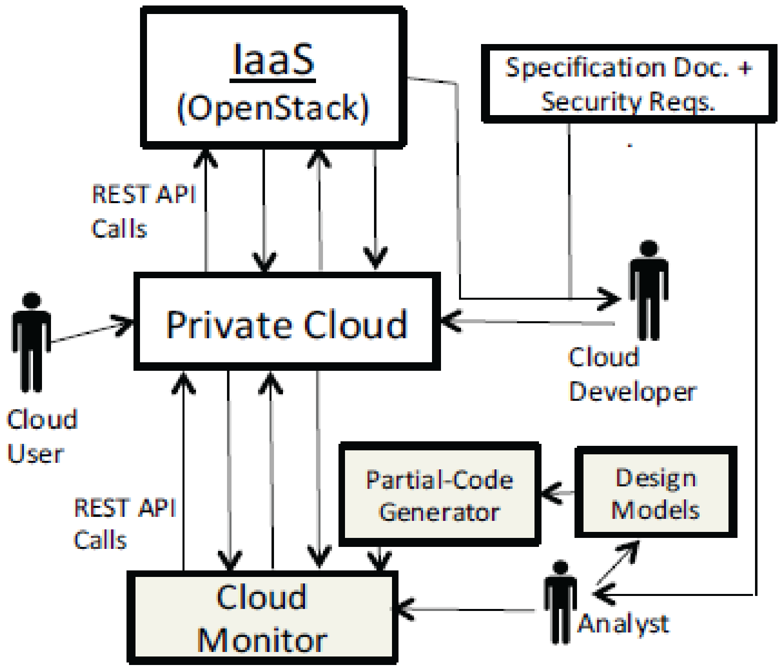
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 6
SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE



6.2 UML DIAGRAM'S:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

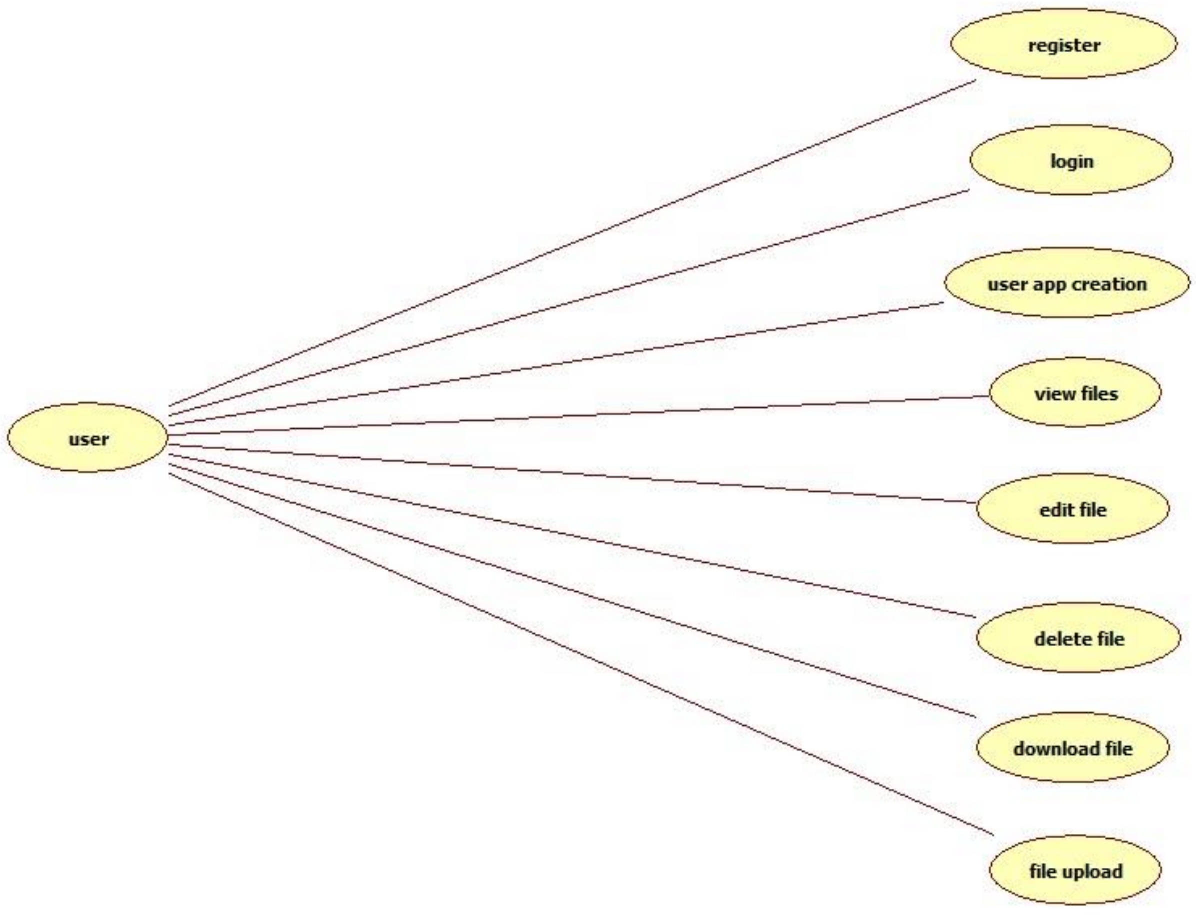
GOALS:

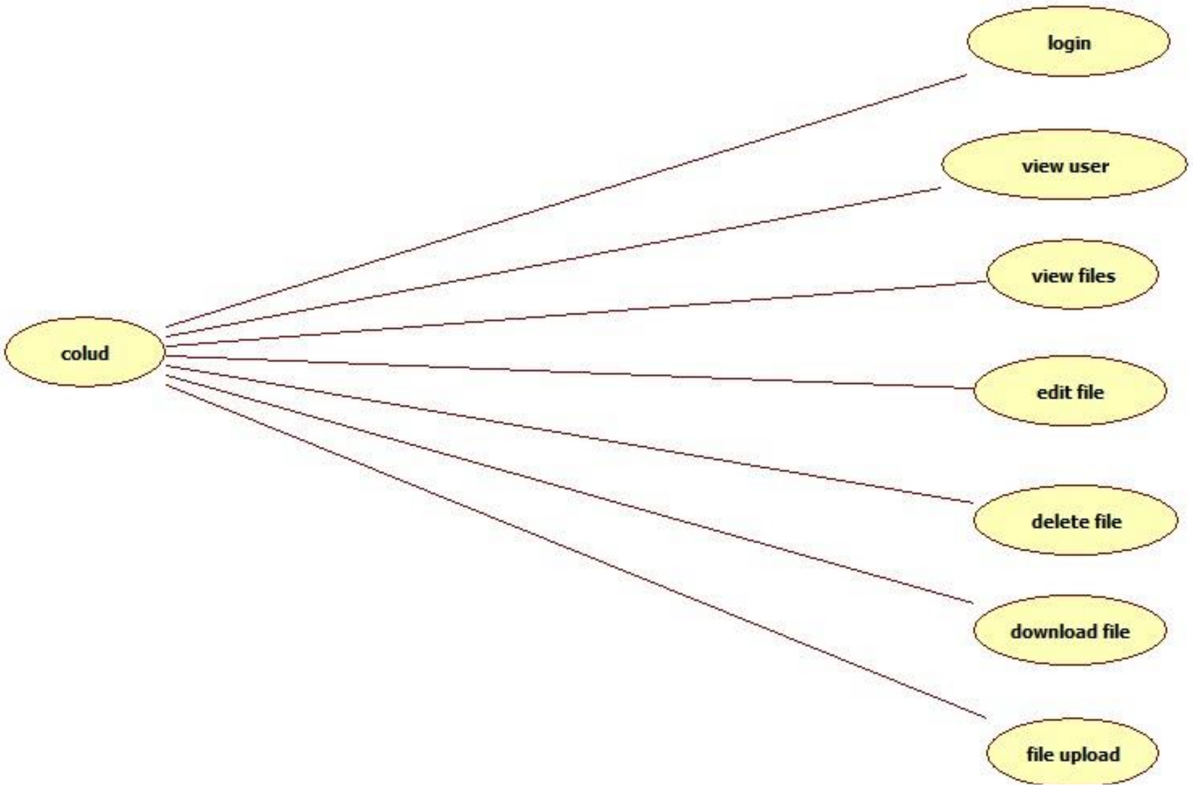
The Primary goals in the design of the UML are as follows:

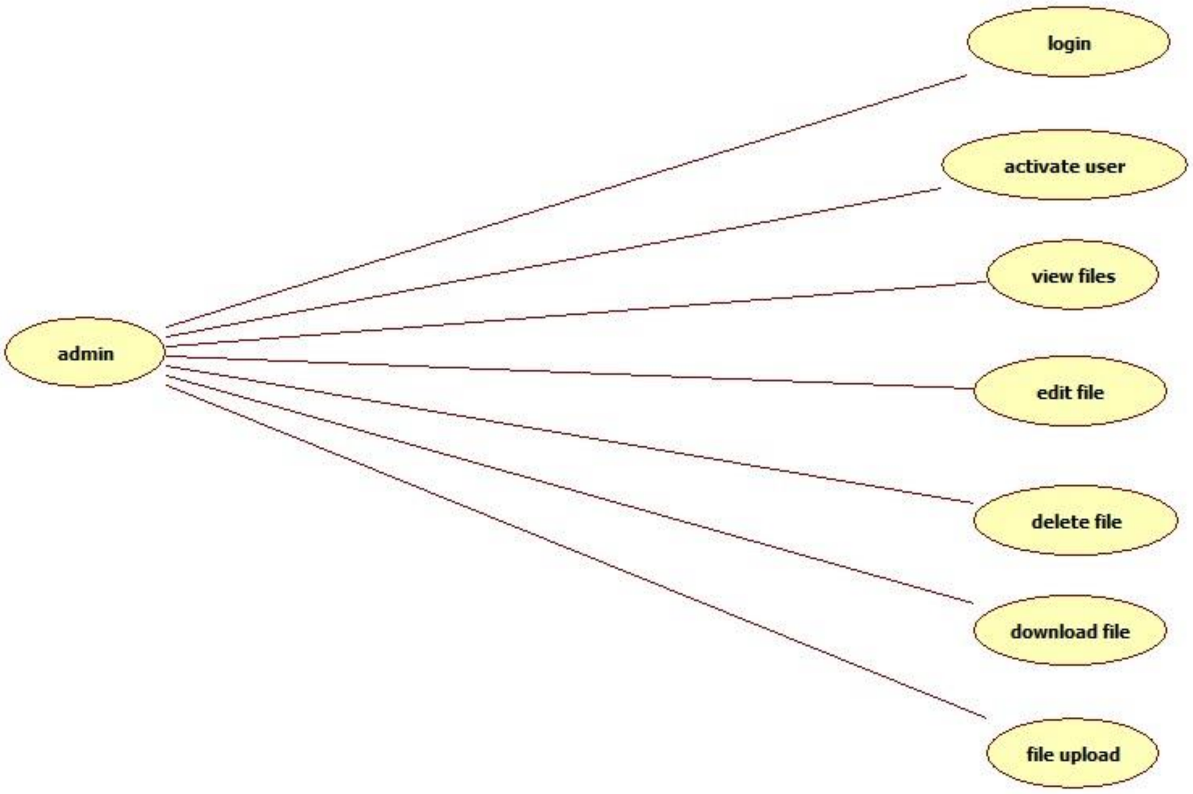
1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

6.2.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

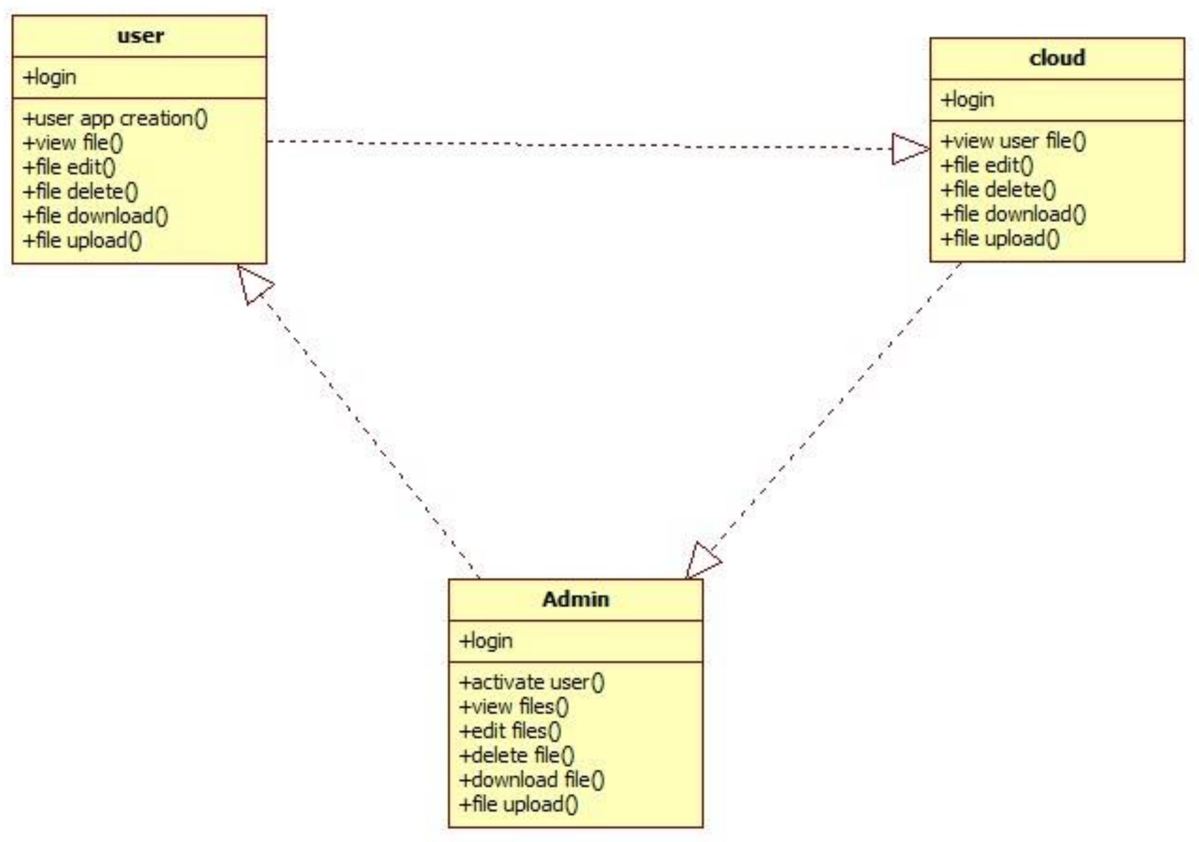



colud
login
view user
view files
edit file
delete file
download file
file upload


admin
login
activate user
view files
edit file
delete file
download file
file upload

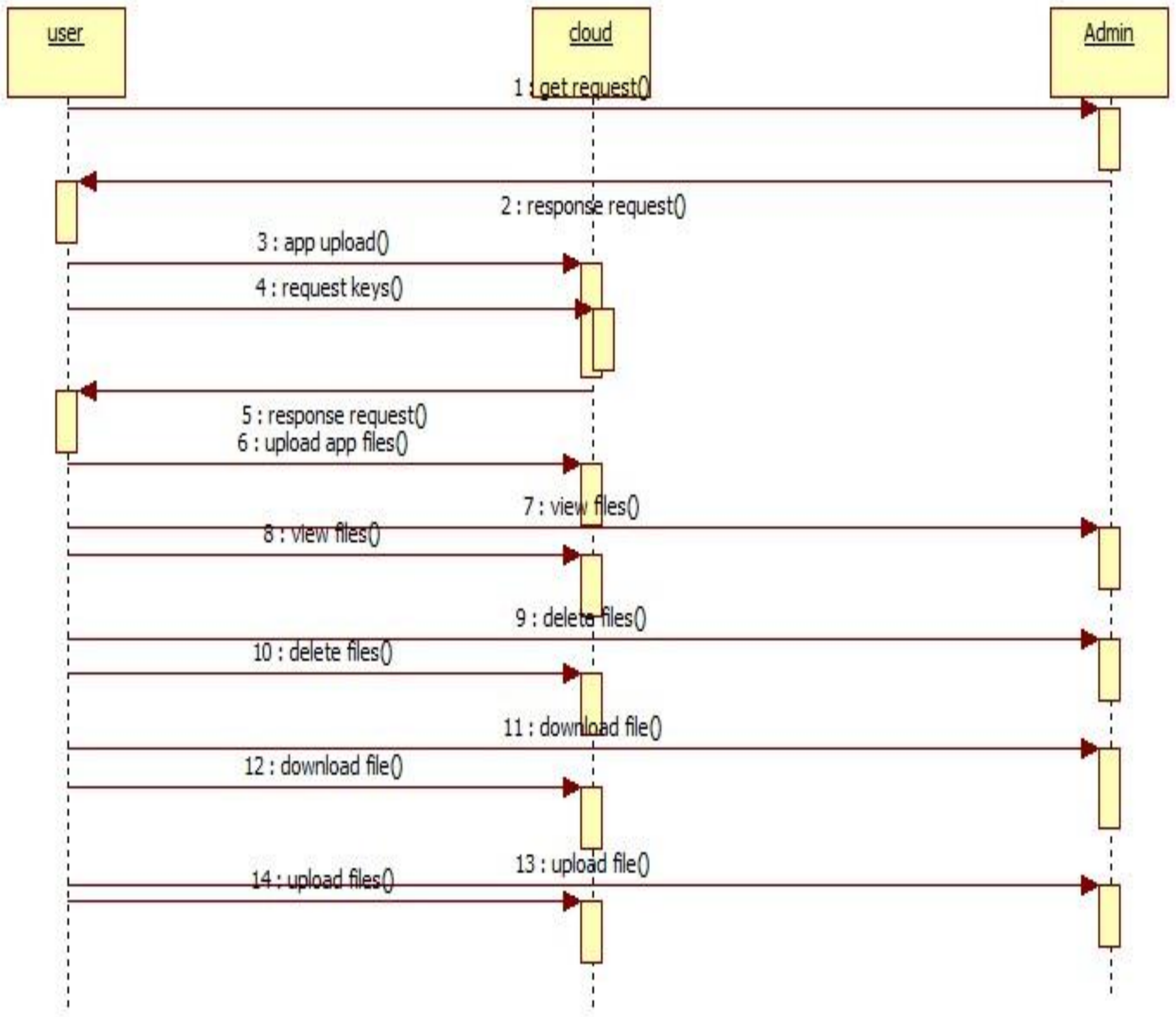
6.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



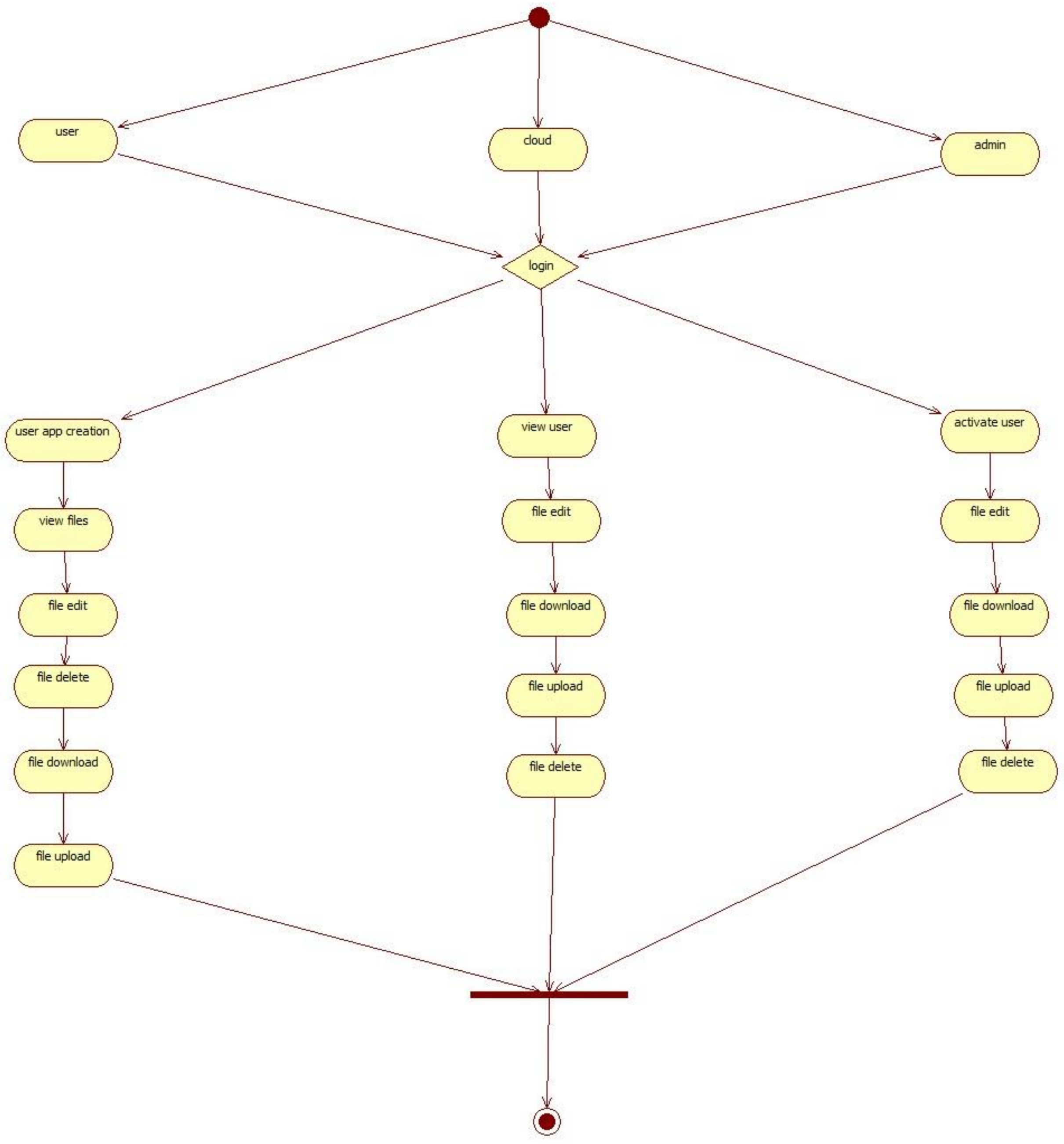
6.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



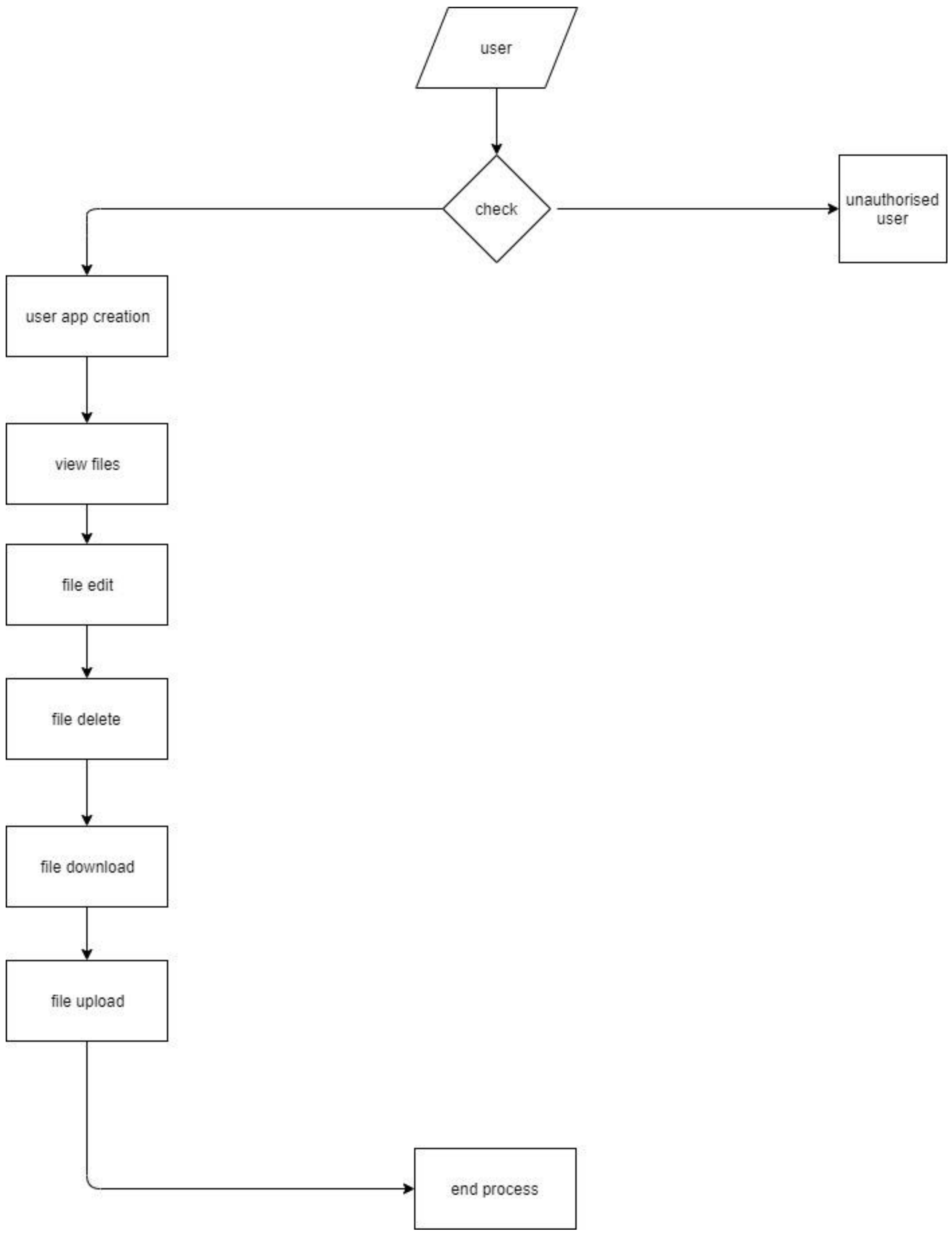
6.2.4 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams canbe used to describe the business and operational step-by-step workflows of components in a system. Anactivity diagram shows the overall flow of control.

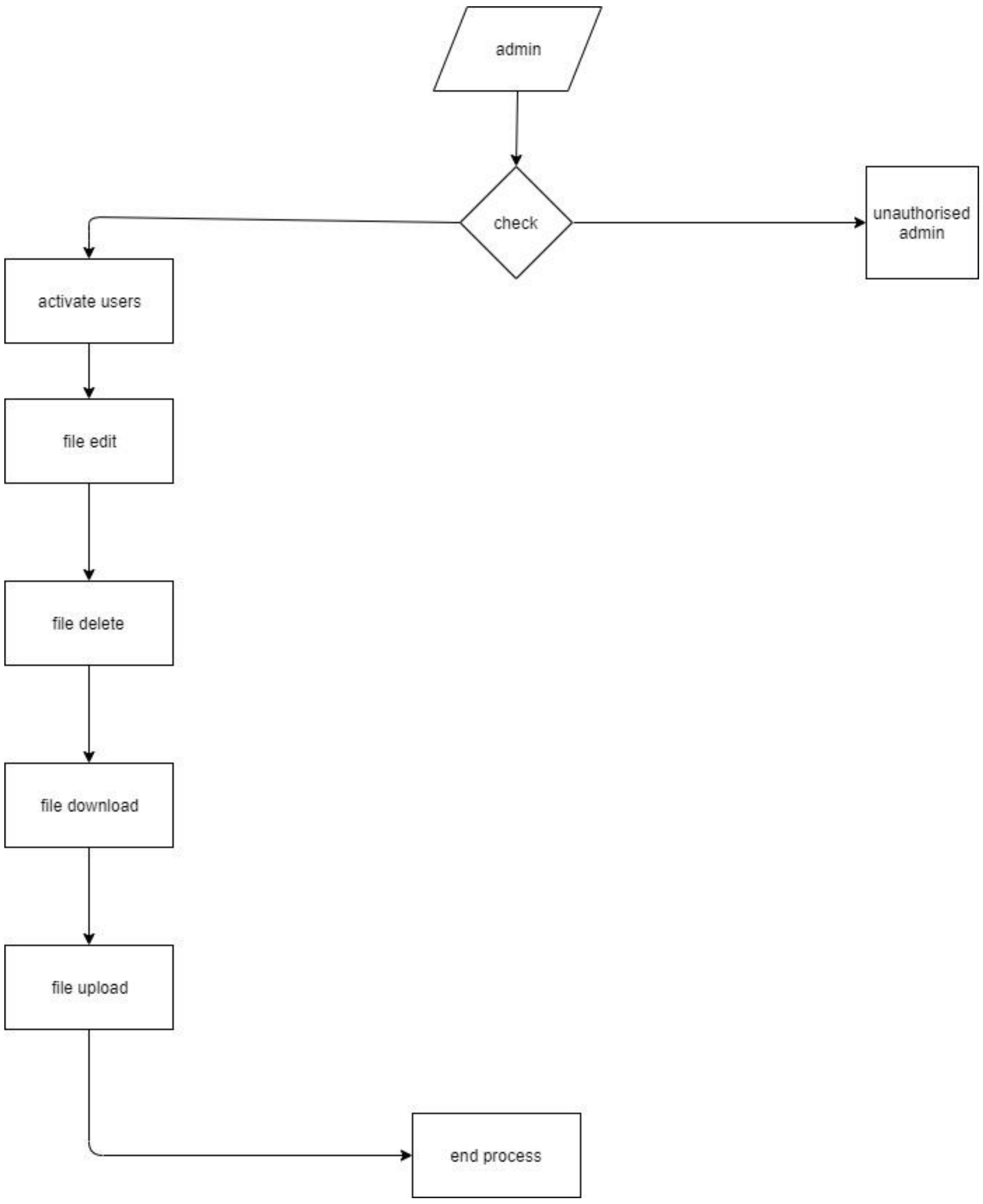


6.2.5 DATA FLOW DIAGRAM

USER:



ADMIN:


admin
check
unauthorised admin
activate users
file edit
file delete
file download
file upload
end process

CHAPTER-7
INPUT AND OUTPUT DESIGN

7.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

7.1.1 OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

7.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER 8
IMPLEMENTATION

8.1 MODULES:

- User
- Cloud
- Admin
- Machine Learning

8.2 MODULE DESCRIPTION

User

It defines the access rights of the cloud users. A volume can be created, if the it has not exceeded its quota of the permitted volumes and a user Authorization is an important security concern in cloud computing environments. a POST request from the authorized user on the volumes resource would create a new volume. a DELETE request on the volume resource by an authorized user would delete the volume. if the user of the service is authorized to do so, and the volume is not attached to any instance. It aims at regulating an access of the users to system resources.

Cloud

The cloud monitors contain contracts used to automatically verify the implementation. A cloud developer uses IaaS to develop a private cloud for her/his organization that would be used by different cloud users within the organization. In some cases, this private cloud may be implemented by a group of developers working collaboratively on different machines. We use Django web framework to implement cloud monitor and OpenStack to validate our implementation.

Admin

the cloud administrator using Keystone and users or user groups are assigned the roles in these projects. It defines the access rights of the cloud users in the project. A volume can be created, if the project has not exceeded its quota of the permitted volumes and a user is authorized to create a volume in the project. Similarly, a volume can be deleted, if the user of the service is authorized to do so, and the volume is not attached to any instance, i.e., its status is not in-use.

Machine learning

Machine learning refers to the computer’s acquisition of a kind of ability to make predictive judgments and make the best decisions by analyzing and learning a large number of existing data. The representation algorithms include deep learning, artificial neural network, decision tree, enhancement algorithm and so on. The key way for computers to acquire artificial intelligence is machine learning. Nowadays, machine learning plays an important role in various fields of artificial intelligence. Whether in aspects of internet search, biometric identification, auto driving, Mars robot, or in American presidential election, military decision assistants and so on, basically, as long as there is a need for data analysis, machine learning can be used to play a role.

CHAPTER 9
SOFTWARE ENVIRONMENT

9.1 What is Python :

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python: -

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages :

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.
The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML-

- Emotion analysis
- Sentiment analysis

- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".**
And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.
But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.
So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning :-

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in

Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit

Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.
The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

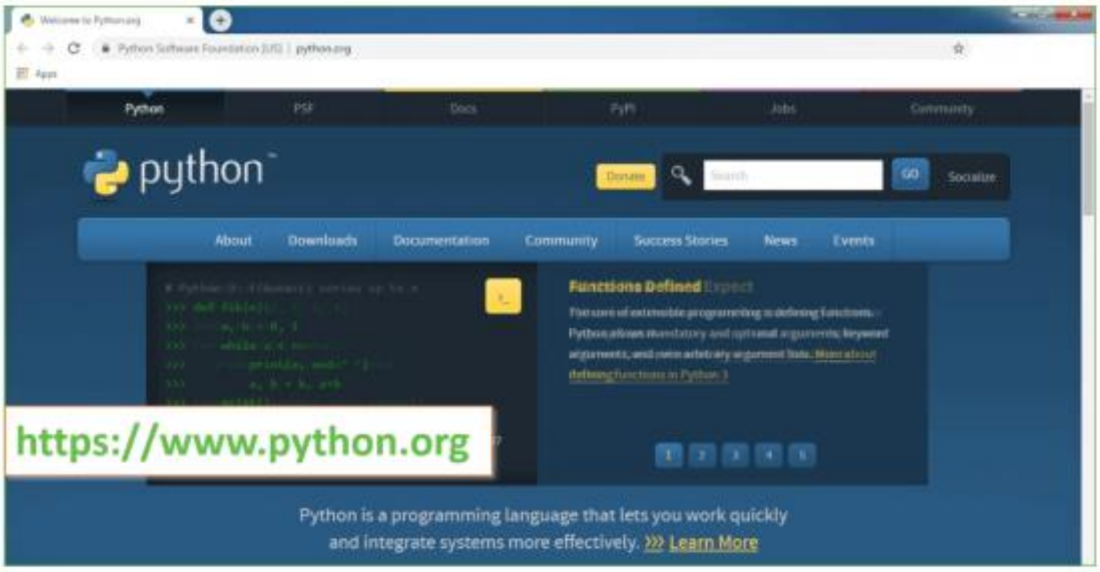
How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.
Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

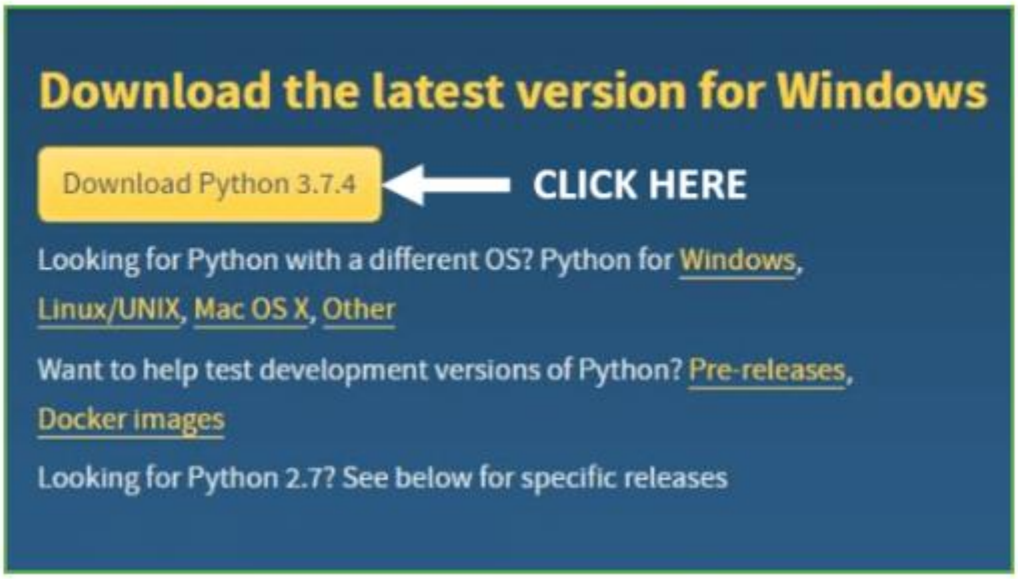
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

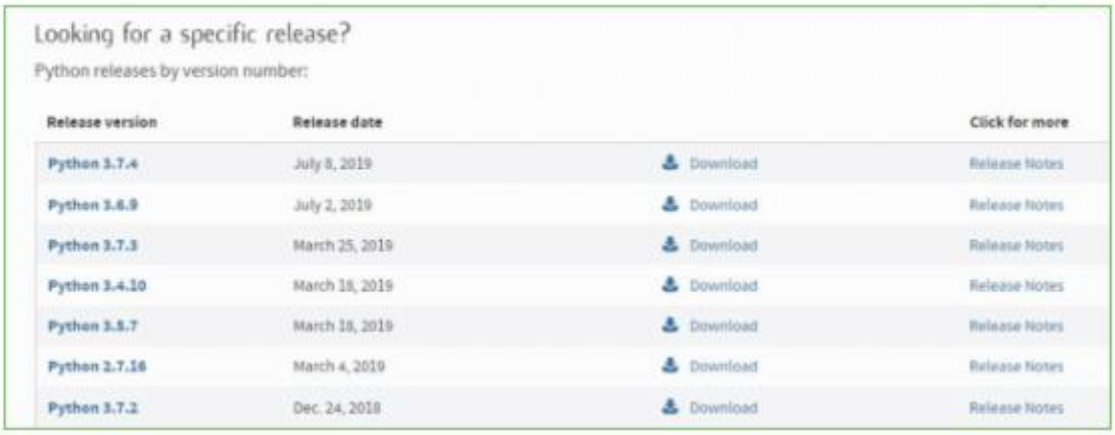


Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.


Download the latest version for Windows
Download Python 3.7.4
CLICK HERE
Looking for Python with a different OS? Python for Windows,
Linux/UNIX, Mac OS X, Other
Want to help test development versions of Python? Pre-releases,
Docker images
Looking for Python 2.7? See below for specific releases

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



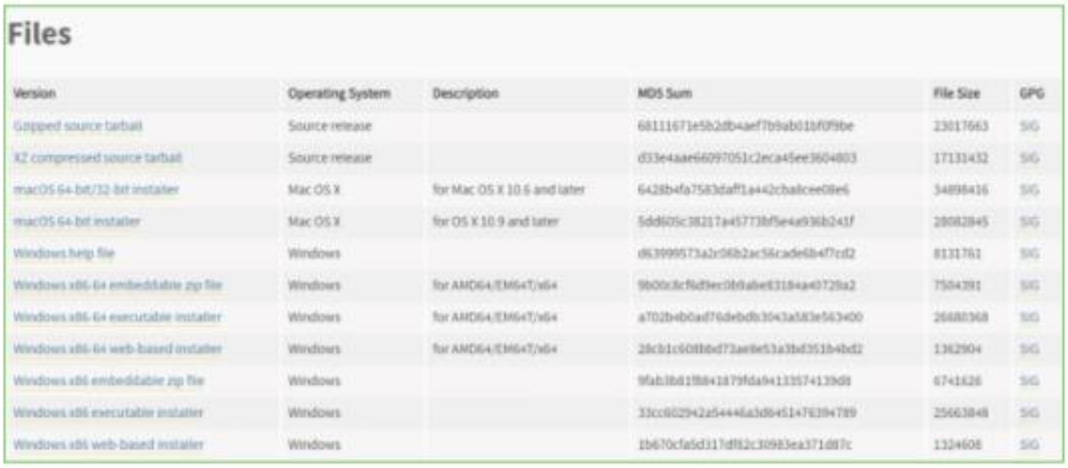
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbalcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

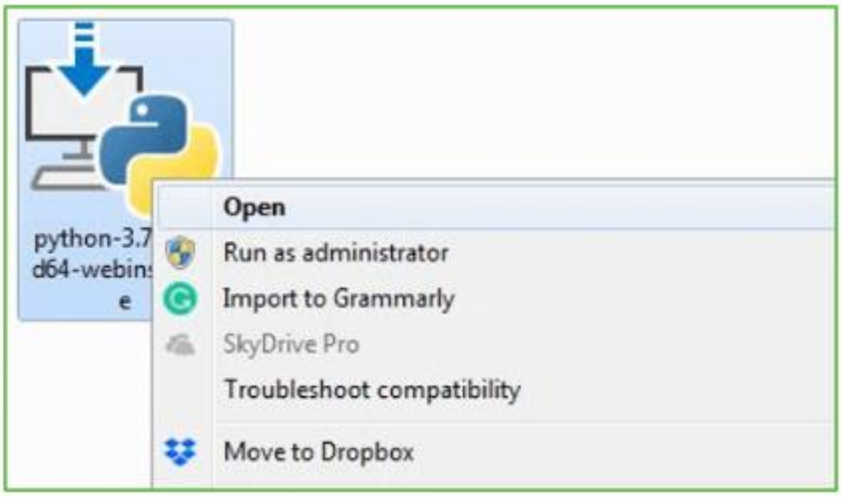
• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer. Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

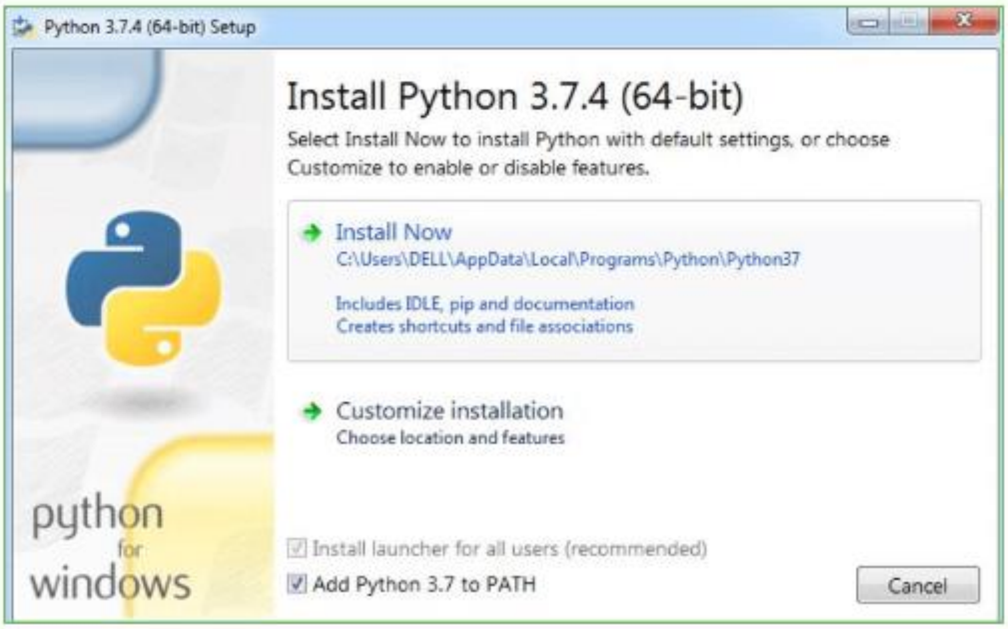
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

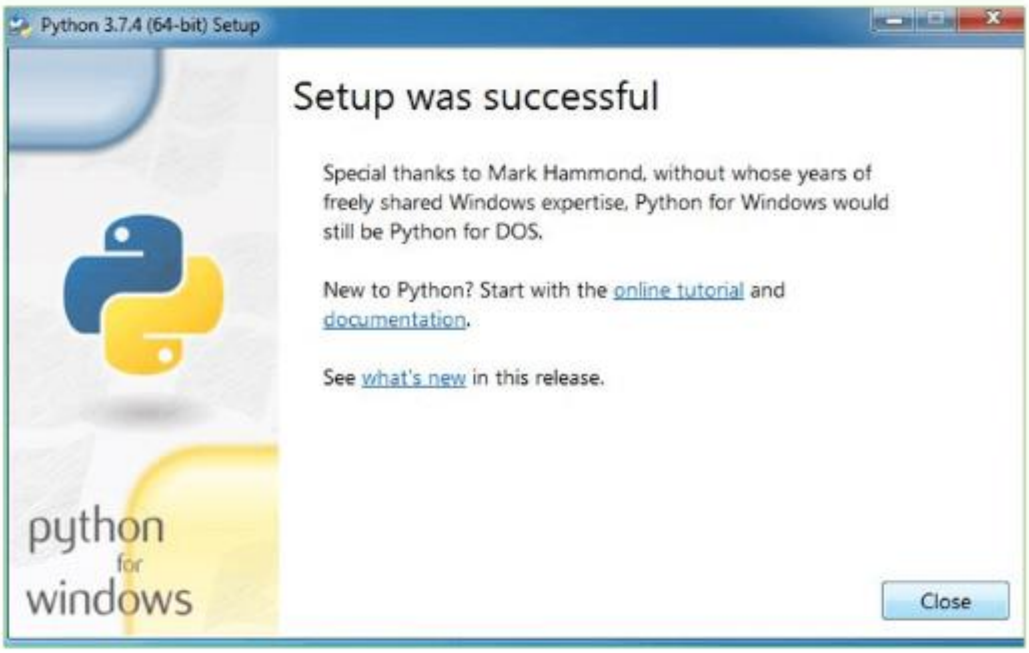
Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Step 3: Click on Install NOW After the installation is successful. Click on Close.


Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

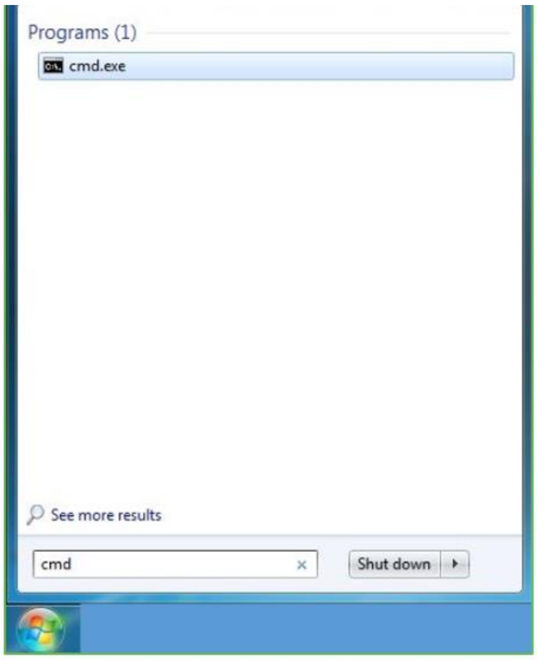
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

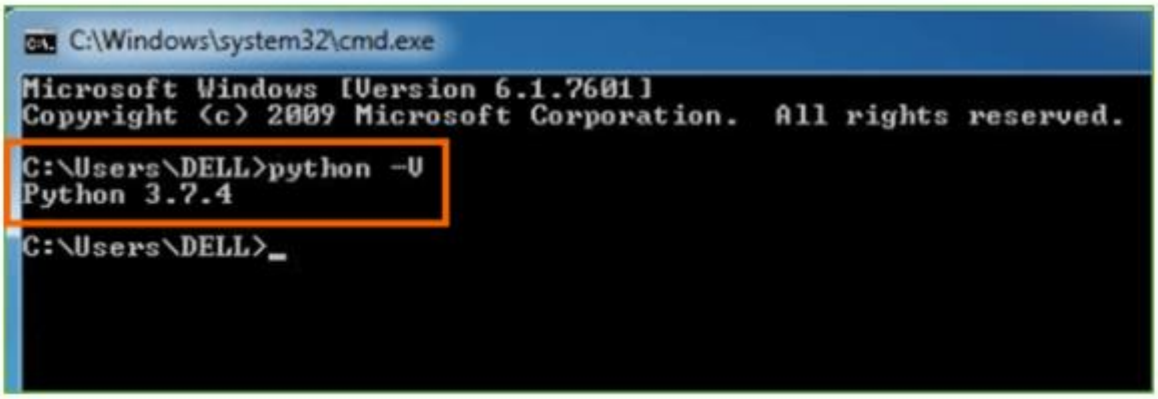
Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

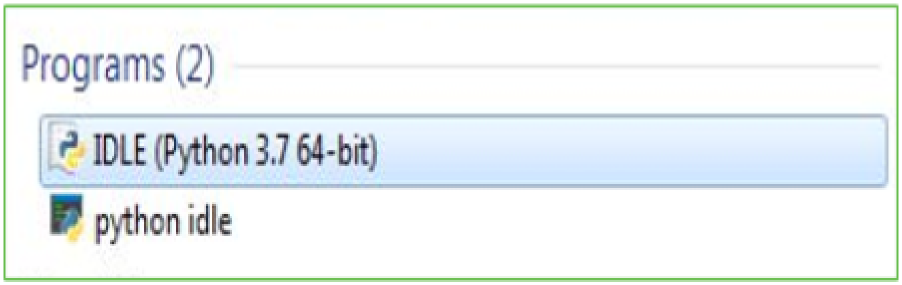
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

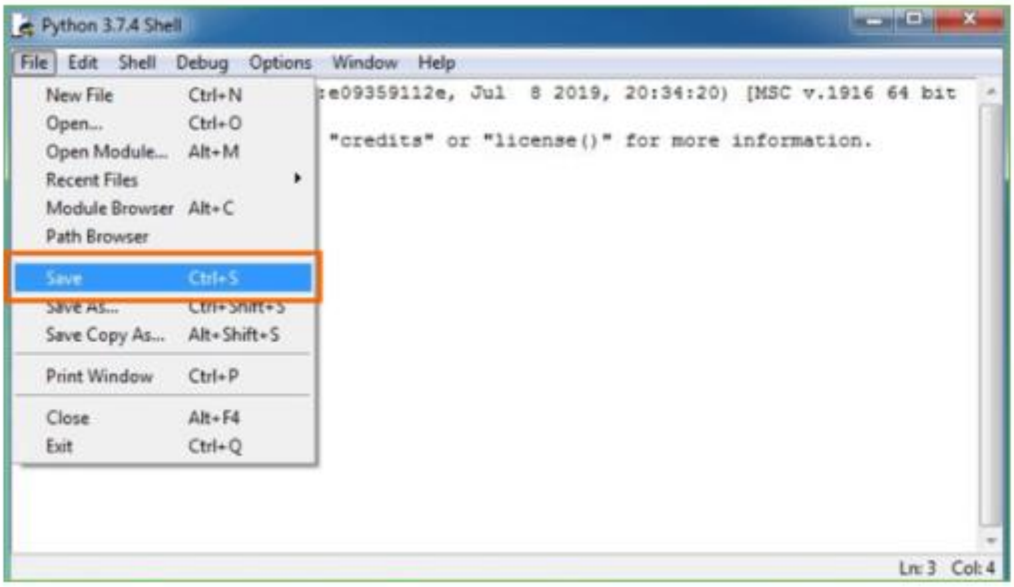
Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit

"credits" or "license()" for more information.
```

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

9.2 SOURCE CODE

urls.py

```
from django.contrib import admin

from django.urls import path

from django.conf import settings

from django.conf.urls.static import static

from users import views

from users.views import
index,userlogin,adminlogin,cloudlogin,userregister,storeregistration,logout,userlogincheck,usercr
eateapp,appcreaterequest,useruploadfile,snippet_detail

from admins.views import adminlogincheck,adminactivateusers,activatewaitedusers

from clouds.views import activateuserapp,cloudlogincheck,clouduserappactivations

from .views import resturl,downloadfile,deletefile,uploadfile


urlpatterns = [

    path('admin/', admin.site.urls),

    path('',index,name='index'),

    path(r'accounts', views.AccountAPIView.as_view(), name='account-list'),

    path(r'contacts', views.ContactAPIView.as_view(), name='contact-list'),

    path(r'activities', views.ActivityAPIView.as_view(), name='activity-list'),

    path(r'activitystatuses', views.ActivityStatusAPIView.as_view(), name='activity-status-list'),

    path(r'contactsources', views.ContactSourceAPIView.as_view(), name='contact-source-list'),

    path(r'contactstatuses', views.ContactStatusAPIView.as_view(), name='contact-status-list'),

    path(r'logout',logout,name='logout'),


    path(r'adminlogincheck',adminlogincheck,name='adminlogincheck'),

    path(r'adminactivateusers',adminactivateusers,name='adminactivateusers'),

    path(r'activatewaitedusers/<id>/$',activatewaitedusers,name='activatewaitedusers'),
    path(r'userlogin',userlogin,name='userlogin'),
    path(r'adminlogin', adminlogin, name='adminlogin'),
    path(r'cloudlogin', cloudlogin, name='cloudlogin'),
    path(r'userregister', userregister, name='userregister'),
    path(r'storeregistration',storeregistration,name='storeregistration'),
    path(r'userlogincheck', userlogincheck, name='userlogincheck'),
    path(r'usercreateapp',usercreateapp,name='usercreateapp'),
    path(r'appcreaterequest',appcreaterequest,name='appcreaterequest'),
    path(r'useruploadfile/<appname>/$',useruploadfile,name='useruploadfile'),
    path(r'^snippet_detail/$',snippet_detail,name='snippet_detail'),

    path(r'resturl/<id>',resturl,name='resturl'),
    path(r'downloadfile/<id>',downloadfile,name='downloadfile'),
    path(r'deletefile/<id>',deletefile,name='deletefile'),
    path(r'uploadfile',uploadfile,name='uploadfile'),



    path(r'activateuserapp',activateuserapp,name='activateuserapp'),
    path(r'cloudlogincheck',cloudlogincheck,name='cloudlogincheck'),
    path(r'clouduserappactivations/<appname>/$',clouduserappactivations,
name='clouduserappactivations'),




]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

Cloud Side views.py

```
from django.shortcuts import render,HttpResponse
from rest_framework.views import APIView
from rest_framework.decorators import api_view
from rest_framework import generics
from users.models import UserFileModel
import os
from django.conf import settings
from django.contrib import messages
from rest_framework import status
from rest_framework.response import Response
import os

from django.http import HttpResponse, Http404
from users.models import UserAppCreatModel

@api_view(['GET', 'PUT', 'DELETE','POST'])
def resturl(request,id):
    role = request.session['role']
    print('ROle is ',role)
    if request.method == 'GET':
        if role=='user':
            dict = {}
            data = UserFileModel.objects.get(id=id)
            filepath = data.userfile
            file = str(filepath).split("/")
            rd = open(os.path.join(settings.MEDIA_ROOT+'/media/', file[1]),'r',encoding='UTF-
8',errors='ignore')
            filedata = rd.read()
            dict.update({'id':id,'filename':file[1],'seckey':data.secretkey,'fdata':filedata})
            return render(request,'users/editfilesdata.html',dict)
        elif role=='admin':
            print('Admin resturl works fine')
            dict = {}
            data = UserFileModel.objects.get(id=id)
            filepath = data.userfile
            file = str(filepath).split("/")
            rd = open(os.path.join(settings.MEDIA_ROOT + '/media/', file[1]), 'r', encoding='UTF-8',
errors='ignore')
            filedata = rd.read()
            dict.update({'id': id, 'filename': file[1], 'seckey': data.secretkey, 'fdata': filedata})
            return render(request, 'admin/admineditfilesdata.html', dict)

        elif role=='cloud':
            return Response(status=status.HTTP_405_METHOD_NOT_ALLOWED)
        else:
            print("Invalid URL")
    elif request.method =='POST':
        fileid = request.POST.get('fileid')
        filename = request.POST.get('filename')
        filedata = request.POST.get('filedata')
        with open(settings.MEDIA_ROOT+'/media' +'/'+filename, 'w+', encoding='UTF-8') as f:
            f.write(filedata)
        return Response(status=status.HTTP_200_OK)
        print('POST  Request Executed')
    print('User ID ',role,'File ID ',id)
    return HttpResponse('Am work fine')

def downloadfile(request,id):
    data = UserFileModel.objects.get(id=id)
    filepath = data.userfile
    # x1 = os.path.join(settings.MEDIA_ROOT+"//"+filepath)
    # print('X1 path = ',filepath)
    fppath = str(filepath).split("/")
    file_path = os.path.join(settings.MEDIA_ROOT+'/media/', fppath[1])
    if os.path.exists(file_path):
        with open(file_path, 'rb') as fh:
            response = HttpResponse(fh.read(), content_type="application/vnd.ms-excel")
            response['Content-Disposition'] = 'inline; filename=' + os.path.basename(file_path)
            return response
    raise Http404
@api_view(('GET',))
def deletefile(request,id):
    role = request.session['role']
    if role == 'user':
        data = UserFileModel.objects.get(id=id)
        data.delete()
        ##filepath = data.userfile
        #fpath = filepath #settings.MEDIA_ROOT+'/'+filepath
        #print('Removing FIle path is ',fpath)
        #os.remove(fpath)
        return Response(status=status.HTTP_200_OK)
    elif role =='admin':
        return Response(status = status.HTTP_405_METHOD_NOT_ALLOWED)
    elif role =='cloud':
        data = UserFileModel.objects.get(id=id)
        data.delete()
        return Response(status = status.HTTP_200_OK)

@api_view(('GET',))
def uploadfile(request):
    role = request.session['role']
    if role =='user':
        usremail = request.session['email']
        dict = UserAppCreatModel.objects.filter(email=usremail)
        return  render(request,'users/uploadfile.html',{'objects':dict})
    elif role =='admin':
        return Response(status = status.HTTP_405_METHOD_NOT_ALLOWED)

    elif role =='cloud':
        return Response(status = status.HTTP_405_METHOD_NOT_ALLOWED)
```

models.py

```
from django.shortcuts import render,HttpResponse
from django.contrib import messages
from users.models import UserAppCreatModel
import string
import random

# Create your views here.

def cloudlogincheck(request):
    if request.method == "POST":
        usid = request.POST.get('name')
        pswd = request.POST.get('password')
        print("User ID is = ", usid)
        if usid == 'cloud' and pswd == 'cloud':
            request.session['role'] = 'cloud'
            return render(request, 'clouds/cloudhome.html')
        else:
            messages.success(request, 'Invalid Login Details')
    return render(request,'cloudlogin.html',{})

def activateuserapp(request):
    dict = UserAppCreatModel.objects.all()
    return render(request,'clouds/userappactivation.html',{'objects':dict})


def clouduserappactivations(request,appname):
    accessKey = genAccessToken(10)
    secretKey = genSecretKey(32)
    print('App Name = ', appname,' Access Key ',accessKey,' Secret Key ',secretKey)

UserAppCreatModel.objects.filter(appname=appname).update(accesskey=accessKey,secretkey=
secretKey)
    dict = UserAppCreatModel.objects.all()
    return render(request, 'clouds/userappactivation.html', {'objects': dict})


def genAccessToken(stringLength=10):
    letters = string.ascii_lowercase
    return ''.join(random.choice(letters) for i in range(stringLength))


def genSecretKey(stringLength=32):
    """Generate a random string of letters and digits """
    lettersAndDigits = string.ascii_letters + string.digits
    return ''.join(random.choice(lettersAndDigits) for i in range(stringLength))
```

User side Views.py

```
from django.db import models
from django.contrib.auth.models import User
import os


INDCHOICES = (
    ('FINANCE', 'FINANCE'),
    ('HEALTHCARE', 'HEALTHCARE'),
    ('INSURANCE', 'INSURANCE'),
    ('LEGAL', 'LEGAL'),
    ('MANUFACTURING', 'MANUFACTURING'),
    ('PUBLISHING', 'PUBLISHING'),
    ('REAL ESTATE', 'REAL ESTATE'),
    ('SOFTWARE', 'SOFTWARE'),
)


class Account(models.Model):
    name = models.CharField("Name of Account", "name", max_length=64)
    email = models.EmailField(blank = True, null = True)
    phone = models.CharField(max_length=20, blank = True, null = True)
    industry = models.CharField("Industry Type", max_length=255, choices=INDCHOICES,
blank=True, null=True)
    website = models.URLField("Website", blank=True, null=True)
    description = models.TextField(blank=True, null=True)
    createdBy = models.ForeignKey(User, related_name='account_created_by',
on_delete=models.CASCADE)
    createdAt = models.DateTimeField("Created At", auto_now_add=True)
    isActive = models.BooleanField(default=False)


    def __str__(self):
        return self.name


class ContactSource(models.Model):
    status = models.CharField("Contact Source", max_length=20)


    def __str__(self):
        return self.status


class ContactStatus(models.Model):
    status = models.CharField("Contact Status", max_length=20)


    def __str__(self):
        return self.status


class Contact(models.Model):
    first_name = models.CharField("First name", max_length=255, blank = True, null = True)
    last_name = models.CharField("Last name", max_length=255, blank = True, null = True)
    account = models.ForeignKey(Account, related_name='lead_account_contacts',
on_delete=models.CASCADE, blank=True, null=True)
    email = models.EmailField()
    phone = models.CharField(max_length=20, blank = True, null = True)
    address = models.TextField(blank=True, null=True)
    description = models.TextField(blank=True, null=True)
    createdBy = models.ForeignKey(User, related_name='contact_created_by',
on_delete=models.CASCADE)
    createdAt = models.DateTimeField("Created At", auto_now_add=True)
    isActive = models.BooleanField(default=False)


    def __str__(self):
        return self.first_name


class ActivityStatus(models.Model):
    status = models.CharField("Activity Status", max_length=20)


    def __str__(self):
        return self.status


class Activity(models.Model):
    description = models.TextField(blank=True, null=True)
    createdAt = models.DateTimeField("Created At", auto_now_add=True)
    contact = models.ForeignKey(Contact, on_delete=models.CASCADE, blank=True,
null=True)

    def __str__(self):
        return self.description

class CloudUsersModel(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=200)
    email = models.CharField(max_length=100,unique=True)
    password = models.CharField(max_length=100)
    mobile = models.CharField(max_length=100)
    address = models.TextField(max_length=100)
    city = models.CharField(max_length=100)
    state = models.CharField(max_length=100)
    status = models.CharField(max_length=100,default='waiting')

    def __str__(self):
        return self.email

    class Meta:
        db_table = "registrations"

class UserAppCreatModel(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=200)
    email = models.CharField(max_length=200)
    appname = models.CharField(max_length=200,unique=True)
    accesskey = models.CharField(max_length=200,default='waiting')
    secretkey = models.CharField(max_length=200,default='waiting')
    def __str__(self):
        return self.appname
    class Meta:
        db_table = "userapps"

class UserFileModel(models.Model):
    id = models.AutoField(primary_key=True)
    name        = models.CharField(max_length=200)
    email       = models.CharField(max_length=200)
    appname          = models.CharField(max_length=200)
    accesskey              = models.CharField(max_length=200)
    secretkey              = models.CharField(max_length=200)
    filename               = models.CharField(max_length=200)
    userfile           = models.FileField(upload_to='media/')

    def __str__(self):
        return os.path.basename(self.userfile.name)
    class Meta:
        db_table = "userfiles"

    def delete(self, *args, **kwargs):
        self.userfile.delete()
        super().delete(*args, **kwargs)
```

CHAPTER 10
RESULTS/DISCUSSION

10.1 SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.1.1 TEST CASES

Test case1 for Login form:

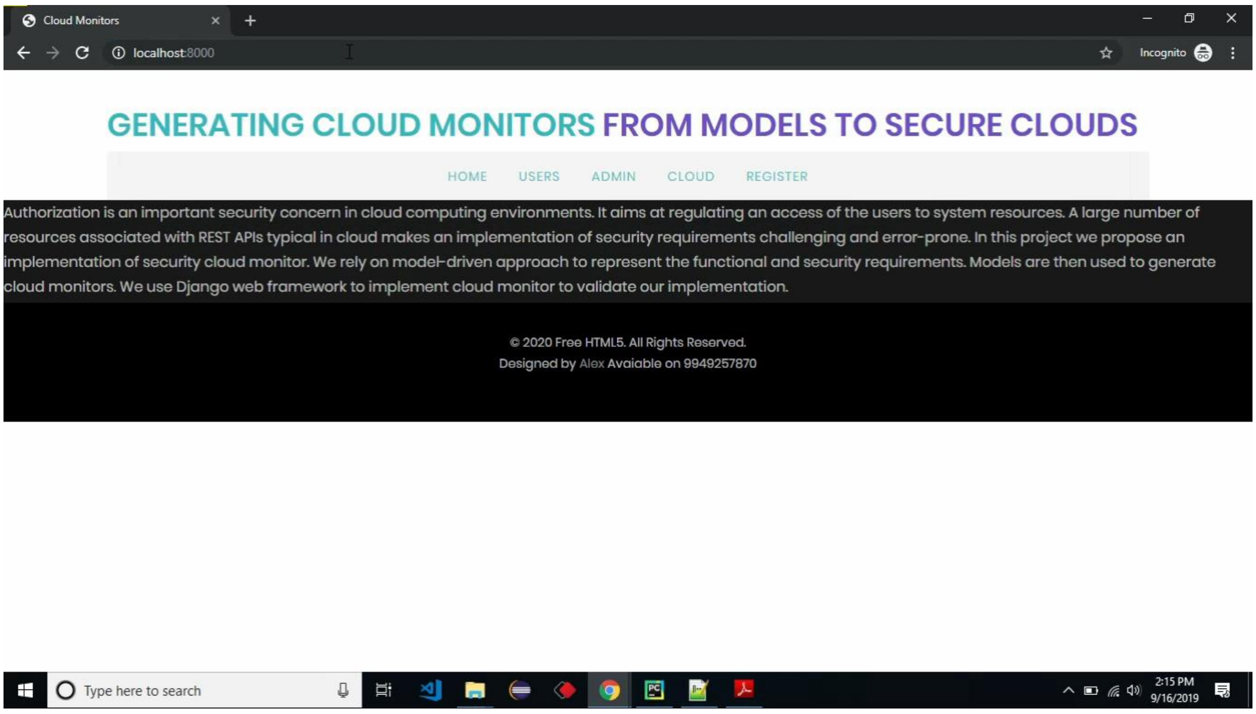
FUNCTION:	**LOGIN**
EXPECTED RESULTS:	Should Validate the user and check his existence in database
ACTUAL RESULTS:	Validate the user and checking the user against the database
LOW PRIORITY	**No**
HIGH PRIORITY	**Yes**

Test case2:

Test case for User Registration form:

FUNCTION	USER REGISTRATION
EXPECTED RESULTS:	Should check if all the fields are filled by the user and saving the user to database.
ACTUAL RESULTS:	Checking whether all the fields are filled by user or not through validations and saving user.
LOW PRIORITY	No
HIGH PRIORITY	Yes

10.2 SCREENSHOTS

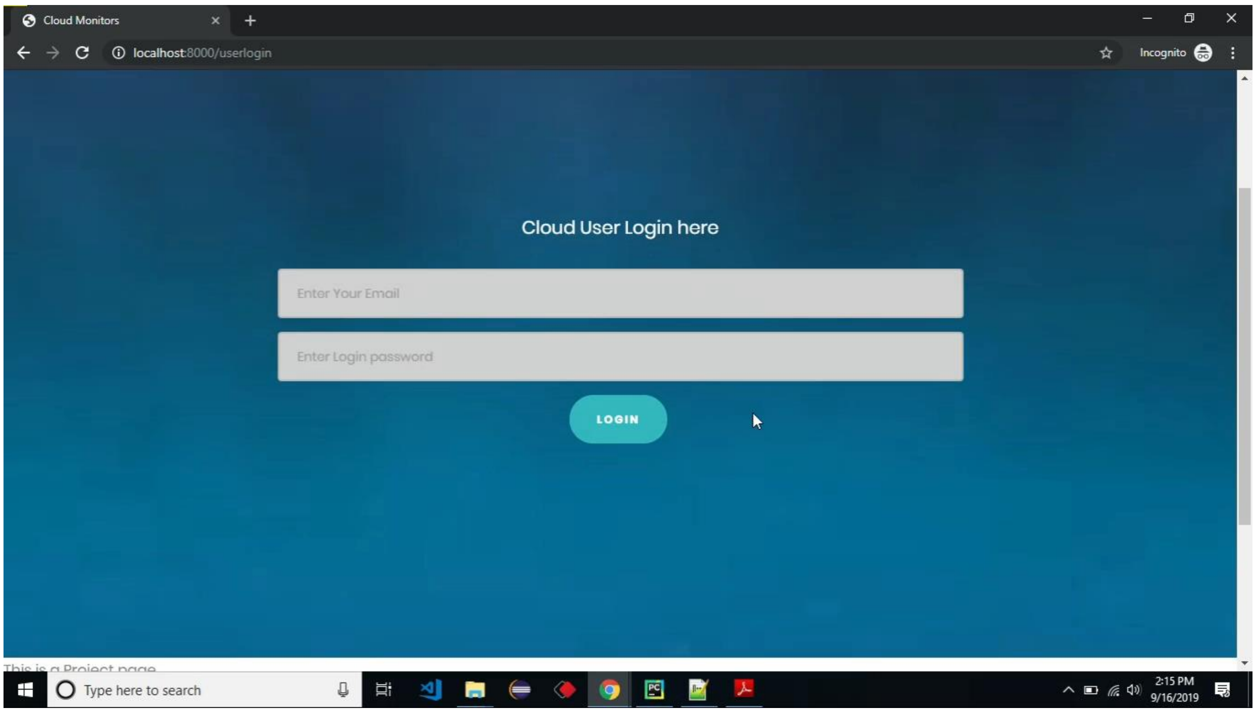


GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

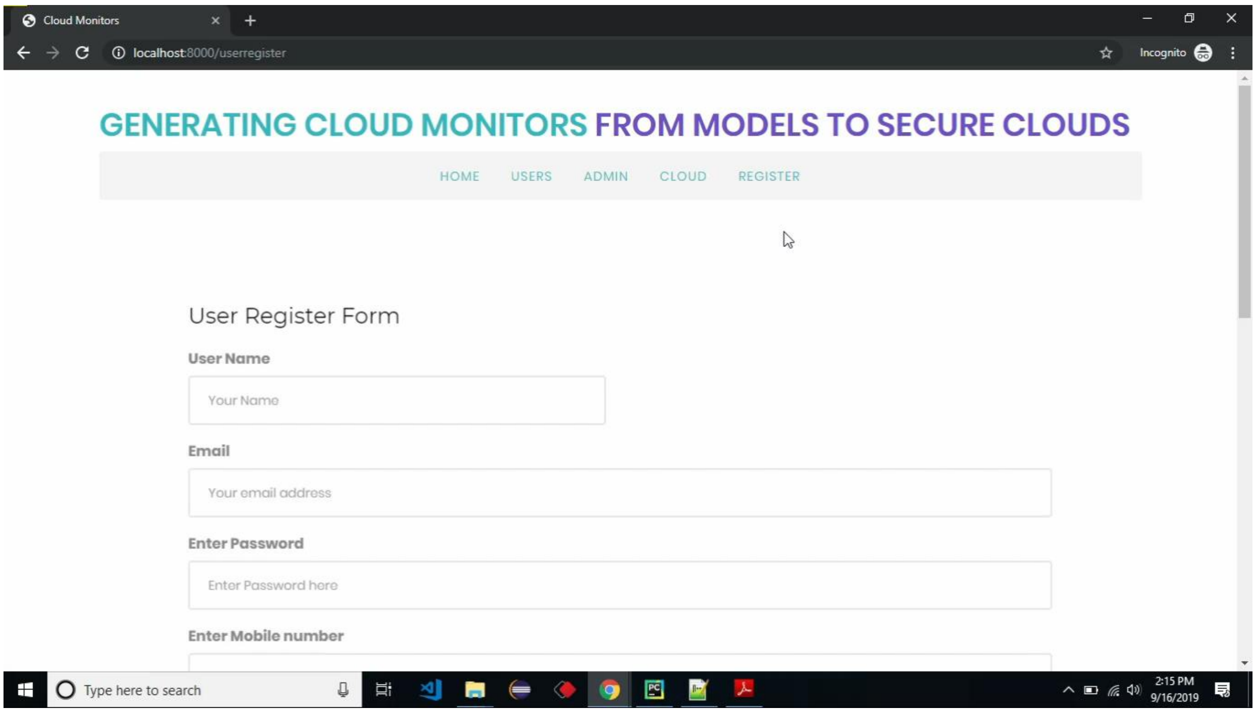
Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. In this project we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. We use Django web framework to implement cloud monitor to validate our implementation.

© 2020 Free HTML5. All Rights Reserved.
Designed by Alex Avaiable on 9949257870

User login



User register


GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

HOME USERS ADMIN CLOUD REGISTER

User Register Form

User Name

Your Name

Email

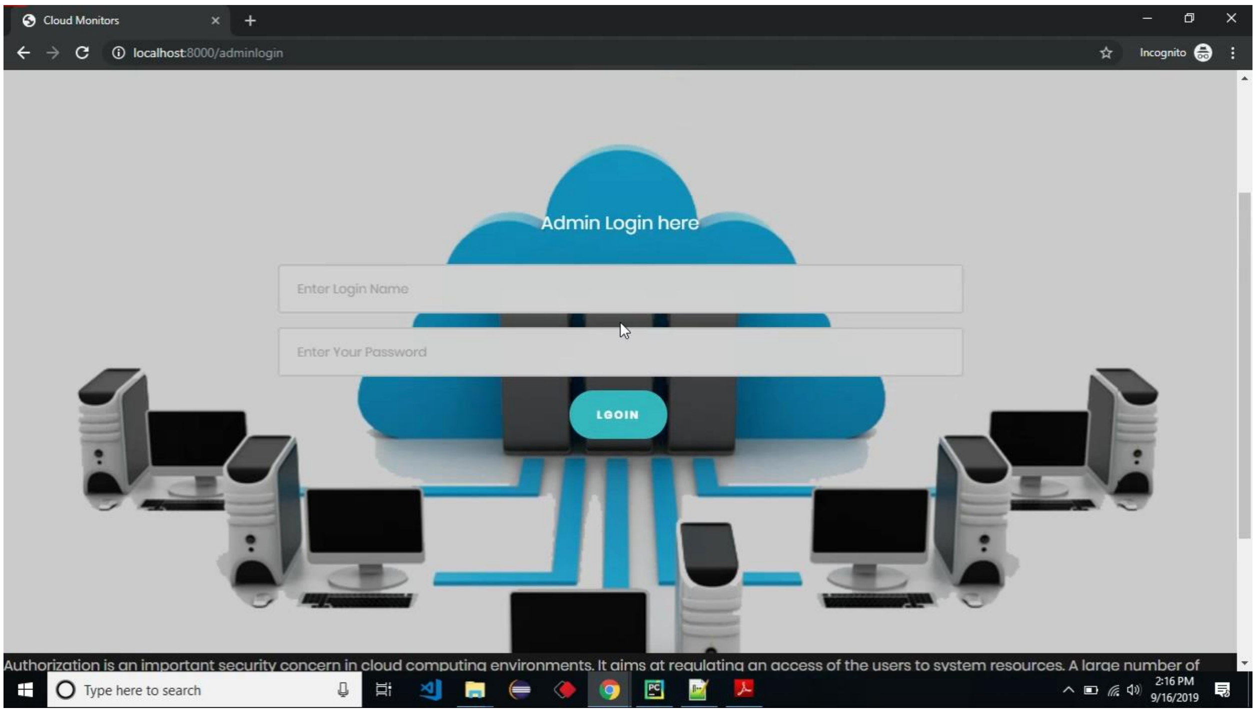
Your email address

Enter Password

Enter Password here

Enter Mobile number

Admin login

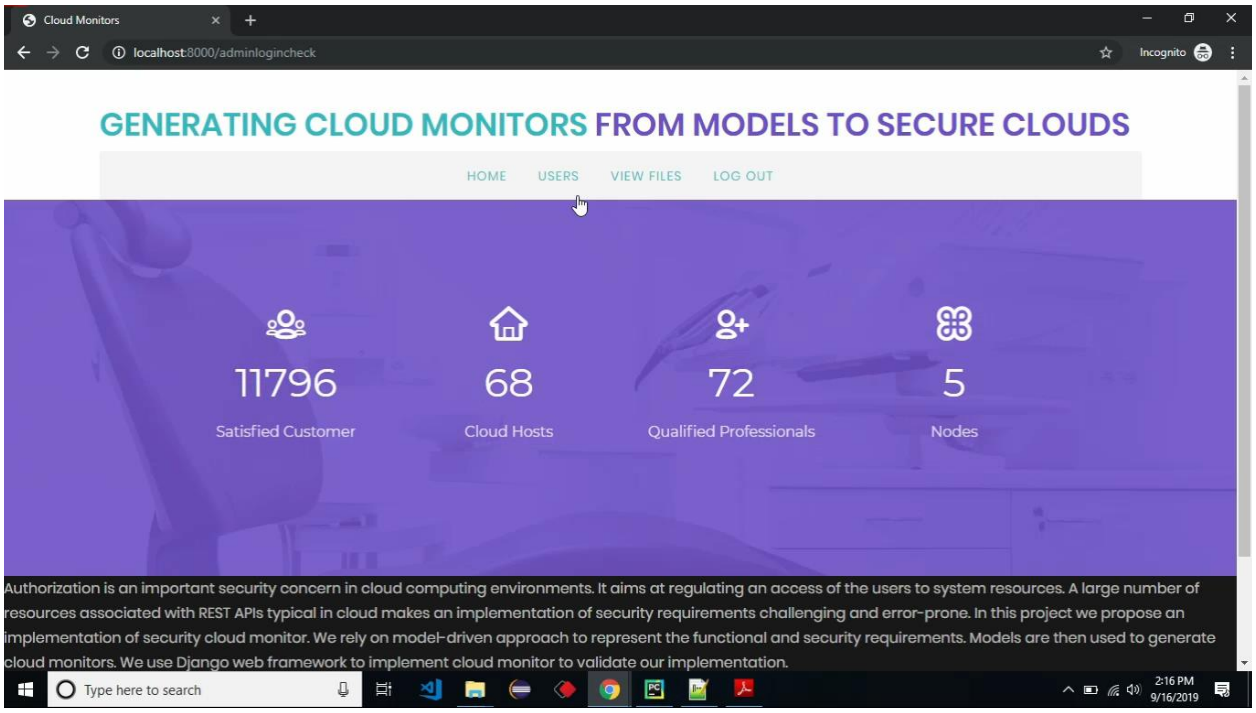

Admin Login here

Enter Login Name

Enter Your Password

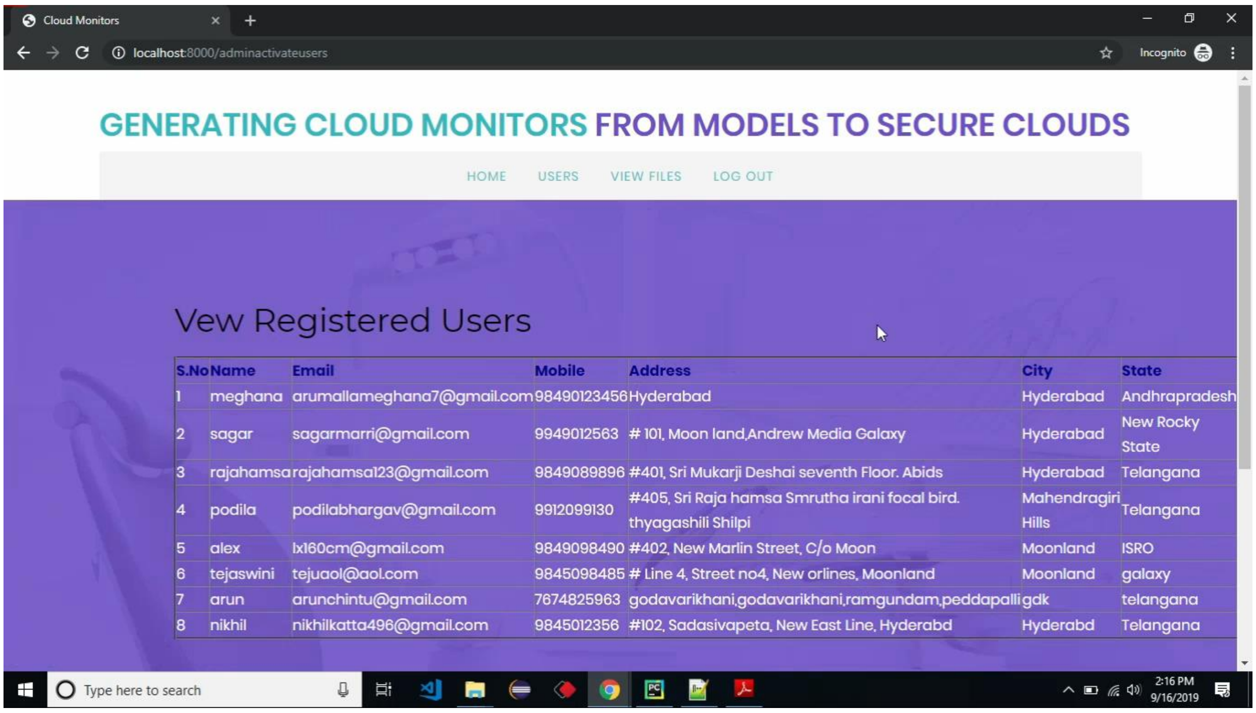
LOGIN

Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of



Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. In this project we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. We use Django web framework to implement cloud monitor to validate our implementation.

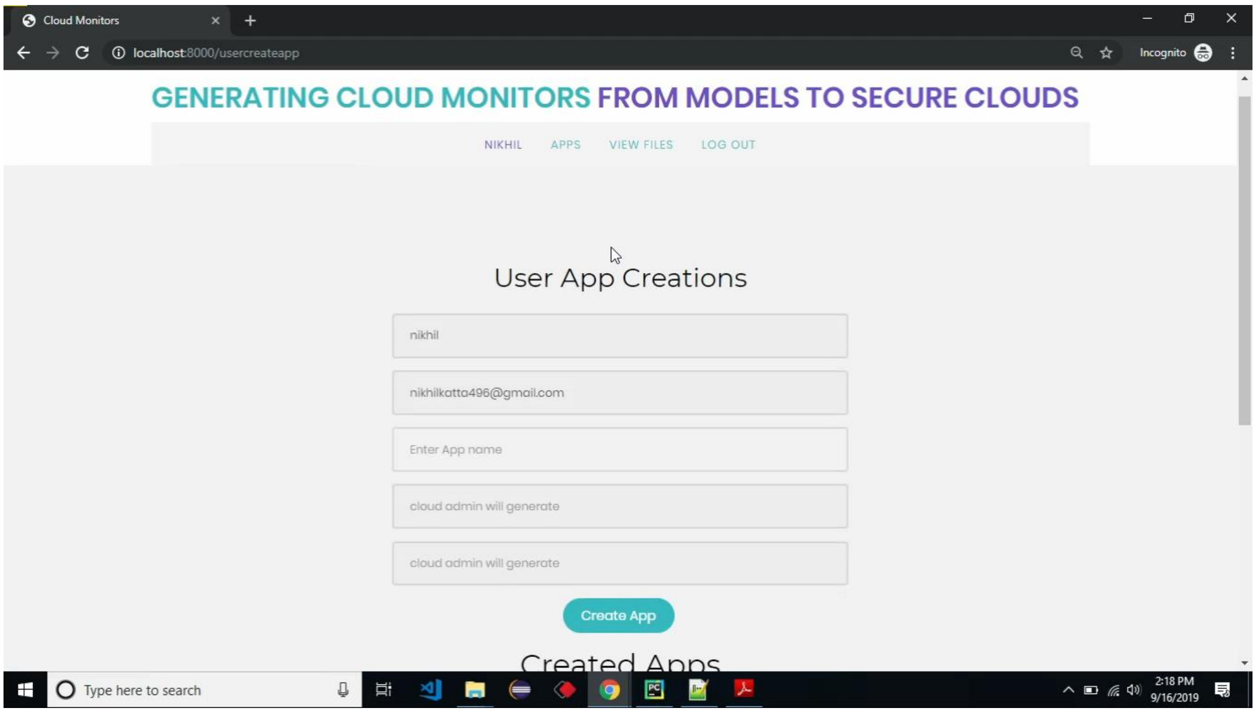
Admin approve user


Cloud Monitors
localhost:8000/adminactivateusers
Incognito
GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS
HOME
USERS
VIEW FILES
LOG OUT
Type here to search
2:16 PM
9/16/2019

Vew Registered Users

S.No	Name	Email	Mobile	Address	City	State
1	meghana	arumallameghana7@gmail.com	98490123456	Hyderabad	Hyderabad	Andhrapradesh
2	sagar	sagarmarri@gmail.com	9949012563	# 101, Moon land,Andrew Media Galaxy	Hyderabad	New Rocky State
3	rajahamsa	rajahamsa123@gmail.com	9849089896	#401, Sri Mukarji Deshai seventh Floor. Abids	Hyderabad	Telangana
4	podila	podilabhargav@gmail.com	9912099130	#405, Sri Raja hamsa Smrutha irani focal bird. thyagashili Shilpi	Mahendragiri Hills	Telangana
5	alex	lx160cm@gmail.com	9849098490	#402, New Marlin Street, C/o Moon	Moonland	ISRO
6	tejaswini	tejuaol@aol.com	9845098485	# Line 4, Street no4, New orlines, Moonland	Moonland	galaxy
7	arun	arunchintu@gmail.com	7674825963	godavarikhani,godavarikhani,ramgundam,peddapalli	gdk	telangana
8	nikhil	nikhilkatta496@gmail.com	9845012356	#102, Sadasivapeta, New East Line, Hyderabd	Hyderabd	Telangana

User app creation



GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

User App Creations

nikhil

nikhilkatta496@gmail.com

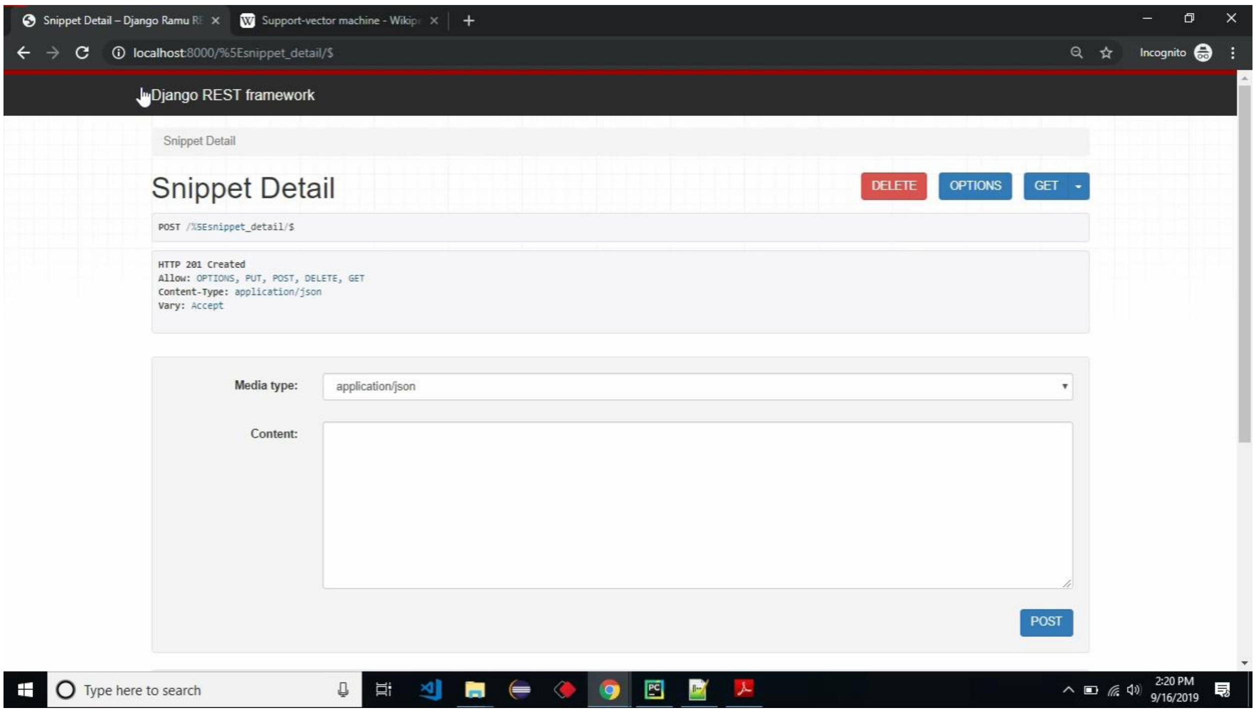
Enter App name

cloud admin will generate

cloud admin will generate

Created Apps

Django rest



Snippet Detail

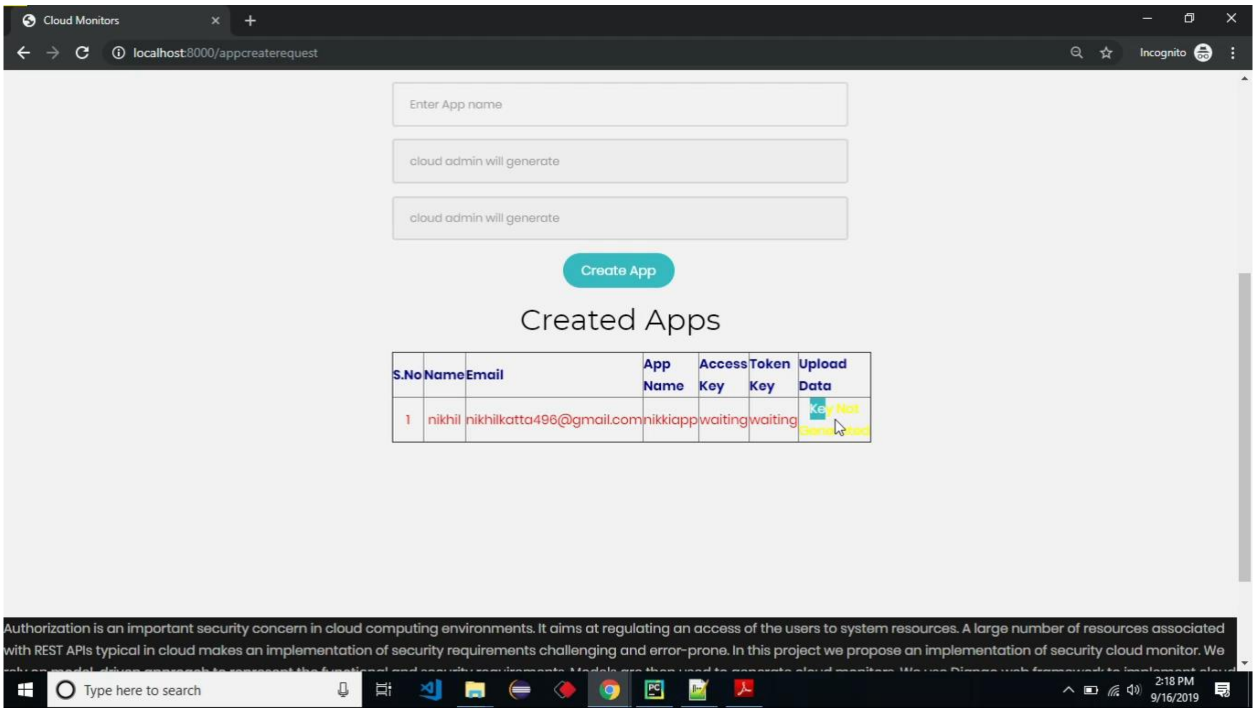
```
POST /%5Esnippet_detail/$

HTTP 201 Created
Allow: OPTIONS, PUT, POST, DELETE, GET
Content-Type: application/json
Vary: Accept
```

Media type: application/json

Content:

user app check

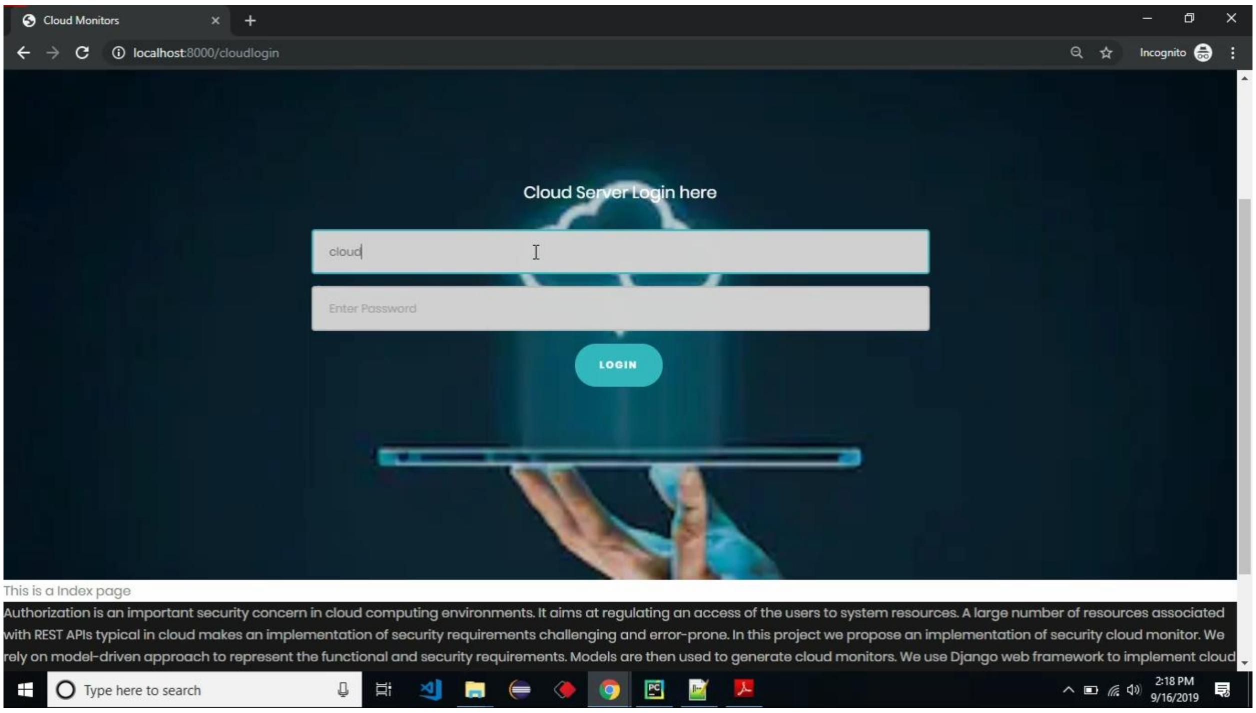

Cloud Monitors
localhost:8000/appcreaterequest
Enter App name
cloud admin will generate
cloud admin will generate
Create App

Created Apps

S.No	Name	Email	App Name	Access Key	Token Key	Upload Data
1	nikhil	nikhilkatta496@gmail.com	nikkiapp	waiting	waiting	Key Not Generated

Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. In this project we propose an implementation of security cloud monitor. We

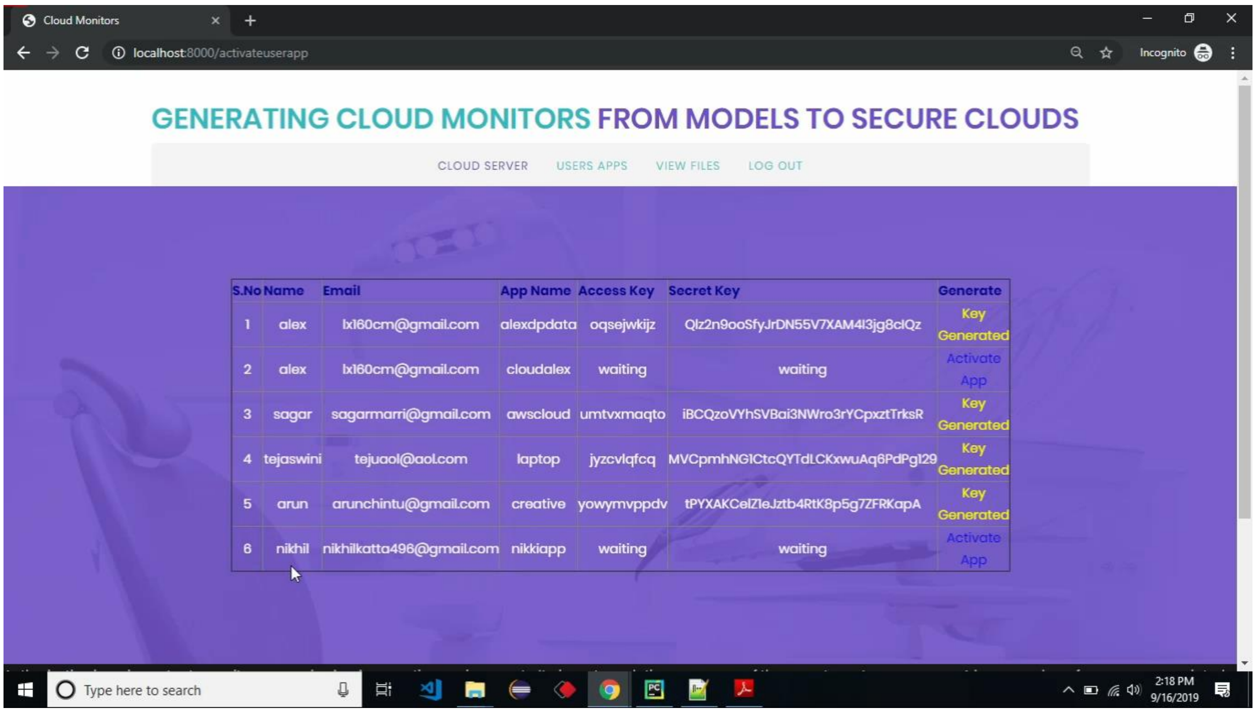
Cloud login



This is a Index page

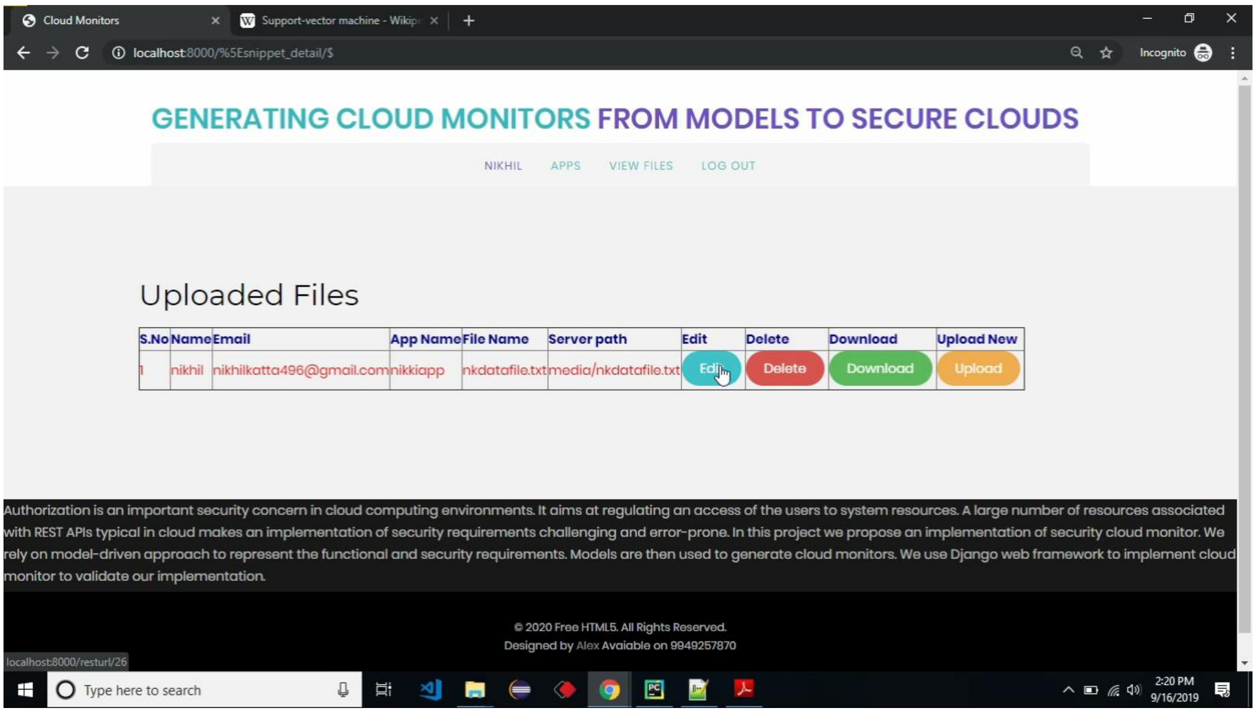
Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. In this project we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. We use Django web framework to implement cloud

Cloud approve app



S.No	Name	Email	App Name	Access Key	Secret Key	Generate
1	alex	lx160cm@gmail.com	alexdpdata	oqsejwkijz	Qlz2n9ooSfyJrDN55V7XAM4l3jg8clQz	Key Generated
2	alex	lx160cm@gmail.com	cloudalex	waiting	waiting	Activate App
3	sagar	sagarmarri@gmail.com	awscloud	umtvxmaqto	iBCQzoVYhSVBai3NWro3rYCpxztTrksR	Key Generated
4	tejaswini	tejuaol@aol.com	laptop	jyzcvlqfcq	MVCpmhNG1CtcQYTdLCKxwuAq6PdPg129	Key Generated
5	arun	arunchintu@gmail.com	creative	yowymvppdv	tPYXAKCelZ1eJztb4RtK8p5g7ZFRKapA	Key Generated
6	nikhil	nikhilkatta496@gmail.com	nikkiapp	waiting	waiting	Activate App

User uploaded file


Cloud Monitors
Support-vector machine - Wikip
localhost:8000/%5Esnippet_detail/$
Incognito
GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS
NIKHIL
APPS
VIEW FILES
LOG OUT
localhost:8000/resturl/26
Type here to search
2:20 PM
9/16/2019

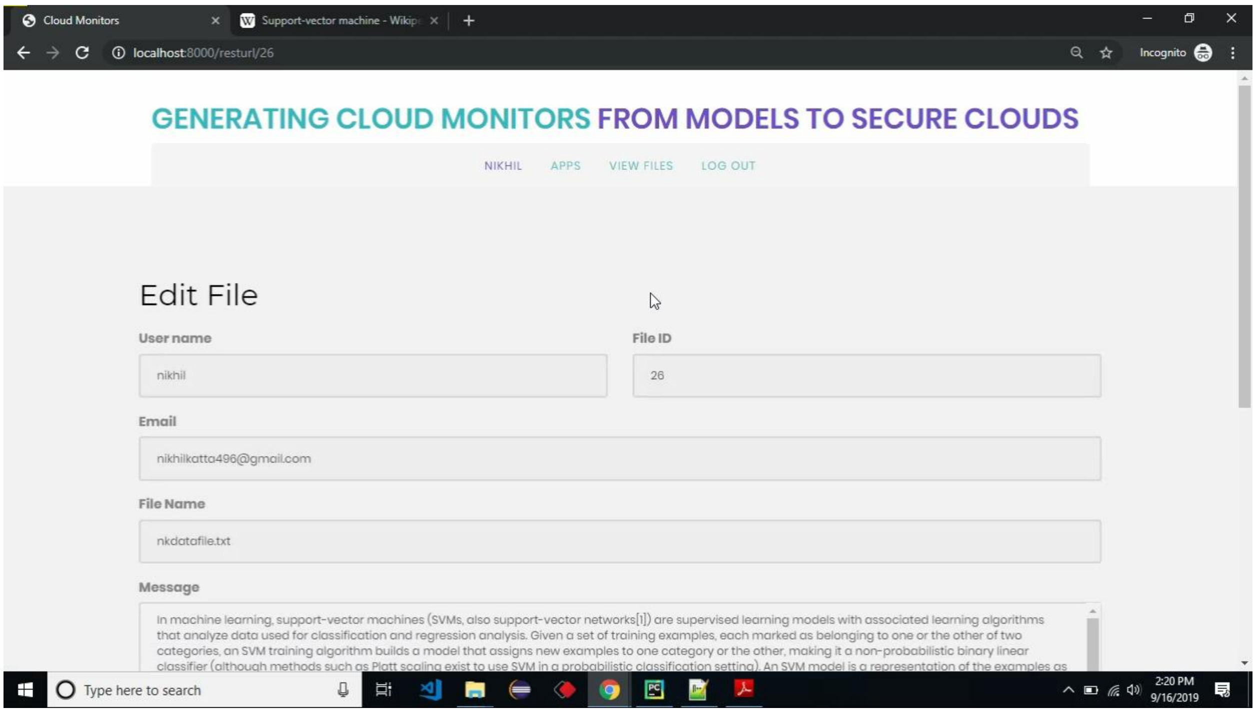
Uploaded Files

S.No	Name	Email	App Name	File Name	Server path	Edit	Delete	Download	Upload New
1	nikhil	nikhilkatta496@gmail.com	nikkiapp	nkdatafile.txt	media/nkdatafile.txt	Edit	Delete	Download	Upload

Authorization is an important security concern in cloud computing environments. It aims at regulating an access of the users to system resources. A large number of resources associated with REST APIs typical in cloud makes an implementation of security requirements challenging and error-prone. In this project we propose an implementation of security cloud monitor. We rely on model-driven approach to represent the functional and security requirements. Models are then used to generate cloud monitors. We use Django web framework to implement cloud monitor to validate our implementation.

© 2020 Free HTML5. All Rights Reserved.
Designed by Alex Avaiable on 9949257870

Edit file



GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

Edit File

User name: nikhil

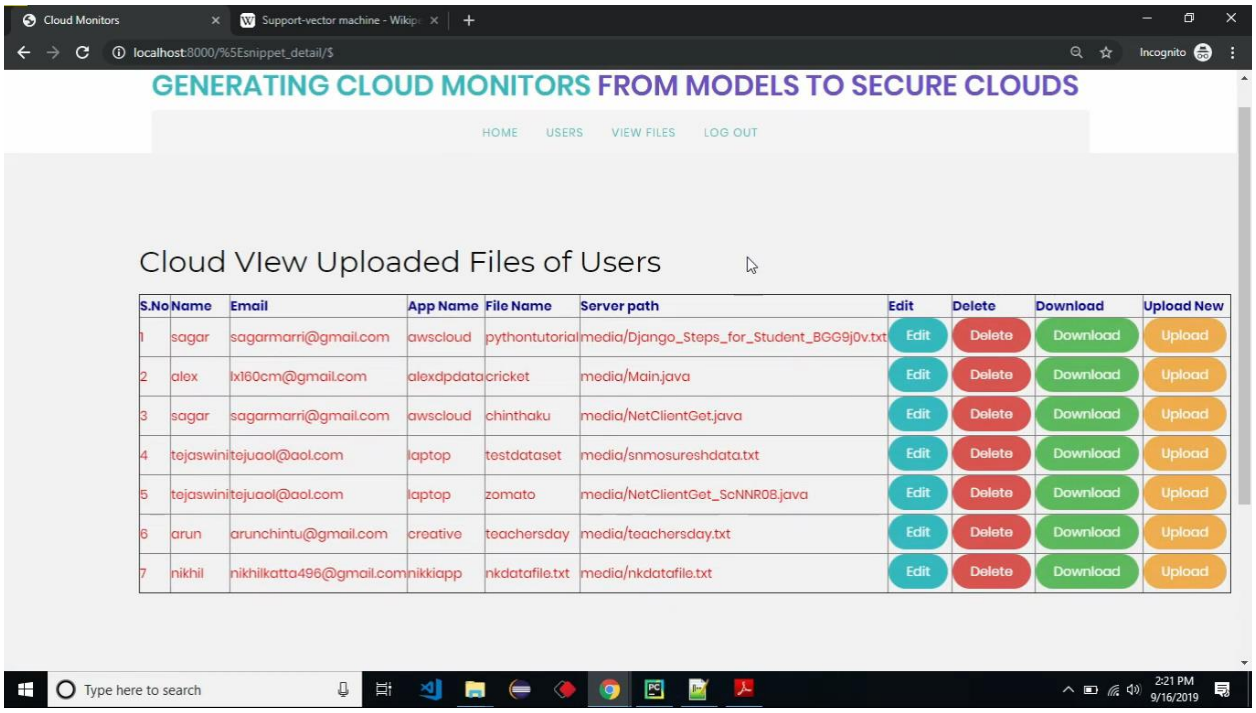
File ID: 26

Email: nikhilkatta496@gmail.com

File Name: nkdatafile.txt

Message

In machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as



GENERATING CLOUD MONITORS FROM MODELS TO SECURE CLOUDS

Cloud VIew Uploaded Files of Users

S.No	Name	Email	App Name	File Name	Server path	Edit	Delete	Download	Upload New
1	sagar	sagarmarri@gmail.com	awscloud	pythontutorial	media/Django_Steps_for_Student_BGG9j0v.txt	Edit	Delete	Download	Upload
2	alex	lx160cm@gmail.com	alexdpdata	cricket	media/Main.java	Edit	Delete	Download	Upload
3	sagar	sagarmarri@gmail.com	awscloud	chinthaku	media/NetClientGet.java	Edit	Delete	Download	Upload
4	tejaswini	tejuaol@aol.com	laptop	testdataset	media/snmosureshdata.txt	Edit	Delete	Download	Upload
5	tejaswini	tejuaol@aol.com	laptop	zomato	media/NetClientGet_ScNNR08.java	Edit	Delete	Download	Upload
6	arun	arunchintu@gmail.com	creative	teachersday	media/teachersday.txt	Edit	Delete	Download	Upload
7	nikhil	nikhilkatta496@gmail.com	nikkiapp	nkdatafile.txt	media/nkdatafile.txt	Edit	Delete	Download	Upload

CHAPTER 11
CONCLUSION

11.1 CONCLUSION

In this paper, we have presented an approach and associated tool for monitoring security in cloud. We have relied on the model-driven approach to design APIs that exhibit REST interface features. The cloud monitors, generated from the models, enable an automated contract-based verification of correctness of functional and security requirements, which are implemented by a private cloud infrastructure. The proposed semi-automated approach aimed at helping the cloud developers and security experts to identify the security loopholes in the implementation by relying on modelling rather than manual code inspection or testing. It helps to spot the errors that might be exploited in data breaches or privilege escalation attacks. Since open source cloud frameworks usually undergo frequent changes, the automated nature of our approach allows the developers to relatively easily check whether functional and security requirements have been preserved in new releases.

11.2 FUTURE SCOPE

The future scope for the project "Generating Cloud Monitors from Models to Secure Clouds" is highly promising, with significant potential to advance cloud security and monitoring. As cloud computing continues to evolve, the integration of automated cloud monitor generation will streamline the process, enabling continuous security checks and reducing manual effort. Expanding support to additional cloud platforms such as AWS, Azure, and Google Cloud will make the system more versatile, catering to a broader range of cloud environments.

Advancements in AI and machine learning can be leveraged to enhance the system's ability to predict and mitigate security threats in real time. Additionally, improving user interfaces with customizable dashboards and real-time alerts will provide a more intuitive experience for administrators and developers. The integration of this system with existing security tools and SIEM systems will create a more comprehensive security solution, while enhanced support for regulatory compliance will ensure that organizations meet legal requirements with ease.

Furthermore, encouraging community contributions and open-source development will drive innovation and ensure the system remains up-to-date with the latest security challenges. Overall, this project has the potential to significantly enhance cloud security, making it more robust, scalable, and adaptable to the ever-changing landscape of cloud computing through continuous technological advancements and integration.

CHAPTER 12
REFERENCES

REFERENCES

[1] Amazon Web Services. https://aws.amazon.com/. Accessed: 30.11.2017.

[2] Block Storage API V3 . https://developer.openstack.org/api-ref/ block-storage/v3/. retrieved: 126.2017.

[3] Cloud Computing Trends: 2017 State of the Cloud Survey. https://www. rightscale.com/blog/cloud-industry-insights/. Accessed: 30.11.2017.

[4] cURL. http://curl.haxx.se/. Accessed: 20.08.2013.

[5] Extensible markup language (xml). https://www.w3.org/XML/. Accessed: 27.03.2018.

[6] Keystone Security and Architecture Review. Online at https://www.openstack.org/summit/ openstack-summit-atlanta-2014/session-videos/presentation/keystonesecurity-and-architecture-review. retrieved: 06.2017.

[7] Nomagic MagicDraw. http://www.nomagic.com/products/magicdraw/. Accessed: 27.03.2018.

[8] OpenStack Block Storage Cinder. https://wiki.openstack.org/wiki/ Cinder. Accessed: 26.03.2018.

[9] OpenStack Newton - Installation Guide. https://docs.openstack.org/ newton/install-guide-ubuntu/overview.html. Accessed: 20.11.2017.

[10] urllib2 - extensible library for opening URLs. Python Documentation. Accessed: 18.10.2012.

[11] Windows Azure. https://azure.microsoft.com. Accessed: 30.11.2017. [

12] MM Alam et al. Model driven security for web services (mds4ws). In Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International, pages 498–505. IEEE, 2004.

[13] Mohamed Almorsy et al. Adaptable, model-driven security engineering for saas cloud-based applications. Automated Software Engineering, 21(2):187–224, 2014.

[14] Christopher Bailey et al. Run-time generation, transformation, and verification of access control models for self-protection. In Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pages 135–144. ACM, 2014.

[15] Tim Berners-Lee et al. Hypertext transfer protocol–HTTP/1.0, 1996.

[16] Gaurav Bhatnagar and QMJ Wu. Chaos-based security solution for fingerprint data during communication and transmission. IEEE Transactions on Instrumentation and Measurement, 61(4):876–887, 2012.

[17] David Ferraiolo et al. Role-based access control (rbac): Features and motivations. In Proceedings of 11th annual computer security application conference, pages 241–48, 1995.

[18] Django Software Foundation. Django Documentation. Online Documentation of Django 2.0, 2017. https://docs.djangoproject.com/en/2.0/.

[19] Michal Gordon and David Harel. Generating executable scenarios from natural language. In International Conference on Intelligent Text Processing and Computational Linguistics. Springer, 2009.

[20] Robert L Grossman. The case for cloud computing. IT professional, 11(2):23–27, 2009.

[21] A. Holovaty and J. Kaplan-Moss. The Django Book. Online version of The Django Book, 2010. http://docs.djangoproject.com/en/1.2/.

[22] Adrian Holovaty and Jacob Kaplan-Moss. The definitive guide to Django: Web development done right. Apress, 2009.

[23] Jan J¨urjens. Towards development of secure systems using umlsec. In International Conference on Fundamental Approaches to Software Engineering, pages 187–200. Springer, 2001.