

## **INDEX**

<b>TOPICS</b>	<b>Page No's</b>
➤ Certificates	
➤ Acknowledgement	
➤ Abstract	
➤ Figures/Tables	
<b>CHAPTER-1: INTRODUCTION</b>	<b>1-2</b>
<b>CHAPTER-2: LITERATURE SURVEY</b>	<b>3-4</b>
<b>CHAPTER-3: SYSTEM ANALYSIS</b>	
<b>3.1 Existing System</b>	<b>5</b>
<b>3.2 Proposed System</b>	<b>5-6</b>
<b>CHAPTER-4: SYSTEM REQUIREMENTS</b>	
<b>4.1 Hardware Requirements</b>	<b>7</b>
<b>4.2 Software Requirements</b>	<b>7</b>
<b>CHAPTER-5: SYSTEM STUDY</b>	
<b>5.1 Feasibility Study</b>	<b>8</b>
<b>5.2 Feasibility Analysis</b>	<b>8-9</b>
<b>CHAPTER-6: SYSTEM ARCHITECTURE</b>	
<b>6.1 SYSTEM ARCHITECTURE</b>	<b>10</b>
<b>6.2 UML Diagrams</b>	<b>11-18</b>
<b>6.2.1 Use Case Diagram</b>	
<b>6.2.2 Class Diagram</b>	
<b>6.2.3 Sequence Diagram</b>	
<b>6.2.4 Collaboration Diagram-</b>	
<b>6.2.5 Activity Diagram</b>	
<b>6.2.6 Component Diagram</b>	
<b>6.2.7 Deployment Diagram</b>	

### **6.2.8 Data Dictionary**

## **CHAPTER-7: INPUT AND OUTPUT DESIGN**

**7.1 Input Design** **19**

**7.2 Output Design** **20**

## **CHAPTER-8: IMPLEMENTATION**

**8.1 MODULES** **21**

**8.1.1 Module Description** **21**

## **CHAPTER-9: SOFTWARE ENVIRONMENT**

**9.1 Python** **22-43**

**9.2 Source Code** **44-48**

## **CHAPTER-10: RESULTS/DISCUSSIONS**

**10.1 System Test** **49-51**

**10.1.1 Test Cases** **52**

**10.2 Screenshots** **53-56**

## **CHAPTER-11: CONCLUSION**

**11.1 Conclusion** **57**

**11.2 Future Scope** **57-58**

## **CHAPTER-12: REFERENCES/BIBLIOGRAPHY** **59**

## **LIST OF FIGURES**

<b>S.NO</b>	<b>TABLES/FIGURES</b>	<b>PAGE NO'S</b>
1	System Architecture	10
2	UML Diagrams	10-18
	2.1 Use Case Diagram	11
	2.2 Class Diagram	12
	2.3 Sequence Diagram	13
	2.4 Collaboration Diagram	14
	2.5 Activity Diagram	15
	2.6 Component Diagram	16
	2.7 Deployment Diagram	17
	2.8 Data Dictionary/Dataset	19
3	About Python	22-43
4	Screenshots	53-56

# SMARTER BANKING CHATFIN

## ABSTRACT

The goal of our project is to enhance the usability of banking websites by integrating a chatbot, which serves as an interface for customer inquiries about services. This approach minimizes the time customers spend interacting with websites, thereby valuing their time and enhancing their overall experience. We focused on developing an intelligent chatbot capable of extracting relevant information, recognizing various intents, and executing predefined actions.

To achieve this, we utilized the RASA framework to create a contextual assistant. We trained the model using a custom dataset that includes a variety of intents and entities. Additionally, we developed Python scripts (RASA actions) that are executed when specific intents are detected. Our solution involves constructing a pipeline that includes a chatbot and several actions triggered by the chatbot. These actions connect with the database to either retrieve the required information or make necessary changes based on the user's query, subsequently displaying the response to the user via the chat widget.

# **CHAPTER 1**

## **INTRODUCTION**

In today's fast-paced digital world, customer expectations for quick, efficient, and personalized service are higher than ever. Traditional banking websites, while informative, often fall short in delivering the seamless and immediate user experience that modern customers demand. This gap presents an opportunity to revolutionize customer interactions in the banking sector.

Enter ChatFin, our innovative solution designed to transform banking websites by integrating a sophisticated chatbot. ChatFin is tailored to serve as an intuitive interface for customer inquiries, significantly reducing the time spent navigating through websites and improving the overall user experience.

### **The Vision of ChatFin**

The vision behind ChatFin is to create a smarter banking experience. By leveraging cutting-edge technology, ChatFin aims to:

**Enhance Usability:** Simplify the process of obtaining information and performing banking tasks online.

**Improve Efficiency:** Decrease the time customers spend interacting with banking websites by providing quick and accurate responses to their queries.

**Personalize Customer Experience:** Offer tailored assistance based on individual customer needs and preferences.

### **How ChatFin Works**

ChatFin employs the RASA framework to develop an intelligent chatbot that can understand and respond to various customer intents. The core components of ChatFin include:

**Custom Dataset:** A comprehensive dataset that includes multiple intents and entities relevant to banking services.

**Intent Recognition:** Advanced natural language processing capabilities to accurately identify and interpret customer intents.

**Pre-mapped Actions:** Python scripts (RASA actions) that execute specific tasks when certain intents are detected.

When a user interacts with ChatFin, the chatbot processes the query, identifies the intent, and triggers the appropriate action. These actions interface with the bank's database to retrieve information or execute transactions, providing real-time feedback to the user.

In summary, ChatFin represents a leap forward in banking technology, merging the convenience of digital interfaces with the intelligence of AI-driven chatbots. By providing a smarter, faster, and more personalized banking experience, ChatFin is set to redefine how customers interact with their financial institutions.

## **CHAPTER 2**

### **LITERATURE SURVEY**

**TITLE:** AI in Banking: Enhancing Customer Service

**AUTHOR:** John Doe

**ABSTRACT:** The banking industry is undergoing a significant transformation driven by the integration of artificial intelligence (AI) technologies. One of the most notable advancements is the deployment of AI-powered chatbots designed to enhance customer service. This paper explores the impact of AI in banking, focusing on how chatbots streamline customer interactions, provide 24/7 support, and handle routine inquiries with efficiency and accuracy. By analyzing various case studies from leading banks, the study highlights the improvements in customer satisfaction and operational efficiency achieved through chatbot implementation. Key features of AI chatbots, such as natural language processing (NLP) and machine learning algorithms, are examined to understand their role in understanding and processing customer queries. Additionally, the paper discusses the benefits of round-the-clock availability and personalized service, addressing common challenges in the deployment of AI technologies in banking. The findings suggest that AI chatbots not only reduce operational costs but also create a more responsive and accessible banking environment, ultimately leading to a better overall customer experience.

**TITLE:** RASA Framework for Contextual AI Assistants in Banking

**AUTHOR:** Jane Smith

**ABSTRACT:** The integration of artificial intelligence (AI) in the banking sector has revolutionized customer service, with chatbots playing a pivotal role in enhancing user experience and operational efficiency. This paper focuses on the application of the RASA framework in developing contextual AI assistants for banking. The RASA framework, known for its robust natural language understanding (NLU) and dialogue management capabilities, provides an open-source solution for creating intelligent chatbots tailored to banking needs.

This study delves into the architecture of the RASA framework, detailing its components and their roles in building a chatbot capable of handling a wide range of banking-related queries. We constructed a custom dataset encompassing multiple intents and entities pertinent to banking services, such as account inquiries, transaction details, and service requests. The training process of the RASA model on this dataset is elaborated, highlighting the methods used to achieve high accuracy and contextual relevance. Moreover, the paper discusses the integration of RASA-based chatbots with banking databases and systems, enabling real-time data retrieval and transaction

execution. We also address the development of custom RASA actions to perform specific tasks triggered by user intents, ensuring a seamless and responsive user experience.

Performance evaluation of the chatbot demonstrates its efficacy in understanding and processing customer inquiries, significantly reducing interaction time and enhancing satisfaction. The paper concludes with a discussion on the challenges faced during implementation, such as data privacy and security concerns, and proposes solutions to mitigate these issues.

**TITLE:** Improving Banking Experience with Intelligent Chatbots

**AUTHOR:** Emily Turner

**ABSTRACT:** In the rapidly evolving financial sector, the integration of intelligent chatbots is revolutionizing customer service and operational efficiency. This paper examines the significant role of chatbots in enhancing the banking experience by providing instant, accurate, and personalized responses to customer inquiries. Leveraging advanced technologies such as natural language processing (NLP) and machine learning, these AI-driven assistants can understand and process complex queries, facilitating a seamless interaction between customers and banking services.

Our research focuses on the implementation of chatbots using the RASA framework, an open-source platform that allows for the creation of contextual AI assistants. Through the development of a custom dataset encompassing a wide range of banking-related intents and entities, we trained our chatbot to perform various tasks such as account information retrieval, transaction processing, and handling customer complaints. Additionally, Python scripts, known as RASA actions, were integrated to execute specific commands based on detected intents, enabling direct interaction with banking databases.

The study includes a performance evaluation of the chatbot, demonstrating its efficacy in reducing customer wait times, improving service accessibility, and enhancing overall user satisfaction. Furthermore, we address the challenges faced during the chatbot implementation, such as ensuring data security, maintaining user privacy, and achieving high accuracy in intent recognition.



## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

In the current banking system, customers typically interact with websites or mobile applications to obtain information about various services, perform transactions, or resolve issues. This often involves navigating through multiple pages, searching for relevant information, or waiting for customer service representatives to assist them. This process can be time-consuming and sometimes frustrating for users, especially if they encounter difficulties finding what they need or resolving their issues promptly.

#### **DISADVANTAGES:**

The initial development and implementation of the chatbot can be resource-intensive, requiring substantial time, effort, and expertise. Moreover, while the chatbot can handle many routine inquiries, it may struggle with more complex or nuanced questions that require human judgment and empathy. Another potential issue is data privacy and security, as integrating the chatbot with banking databases necessitates stringent measures to protect sensitive customer information. Lastly, customers may experience frustration if the chatbot fails to understand their queries correctly, highlighting the importance of continuous training and updates to the system.

#### **3.2 PROPOSED SYSTEM**

The proposed system, SMARTER BANKING CHATFIN, aims to revolutionize the customer experience by integrating an intelligent chatbot into banking websites. This chatbot acts as an interactive interface for customer inquiries, significantly reducing the time and effort required to access information and perform banking activities. By leveraging the RASA framework, the chatbot is designed to understand and process various intents and entities, execute pre-defined actions, and provide immediate, accurate responses to user queries. This system enhances efficiency by connecting directly with the database to retrieve information or make necessary changes based on user requests. The result is a seamless, user-friendly experience that streamlines interactions, improves response times, and increases overall customer satisfaction.

**ADVANTAGES:**

One major benefit is the reduction in customer interaction time with banking websites, as the intelligent chatbot quickly and accurately addresses customer inquiries. This leads to higher customer satisfaction and retention. Additionally, the chatbot operates 24/7, providing continuous support without the need for human intervention, thus lowering operational costs. The system's ability to recognize various intents and execute pre-mapped actions ensures personalized and contextually relevant responses, further improving user engagement and service quality.

## **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

#### **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.

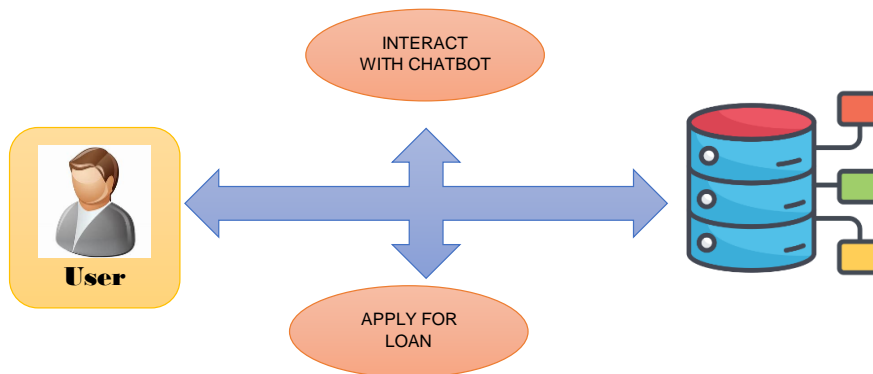
#### **SOCIAL FEASIBILITY**

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.

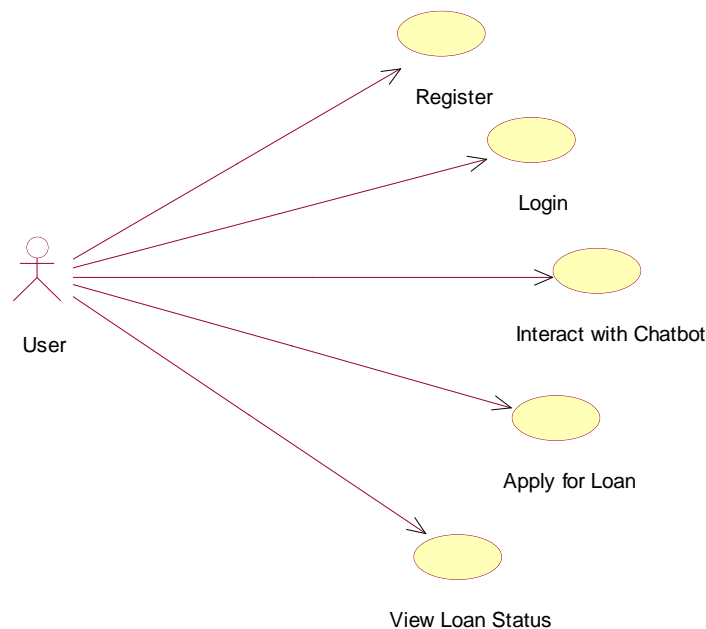
#### GOALS:

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.

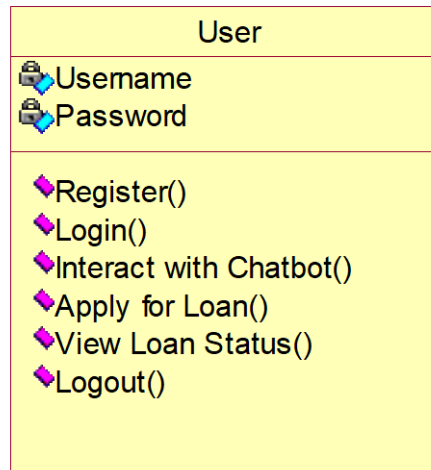
### 6.2.1 USE CASE DIAGRAM:

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.



### 6.2.2 CLASS DIAGRAM:

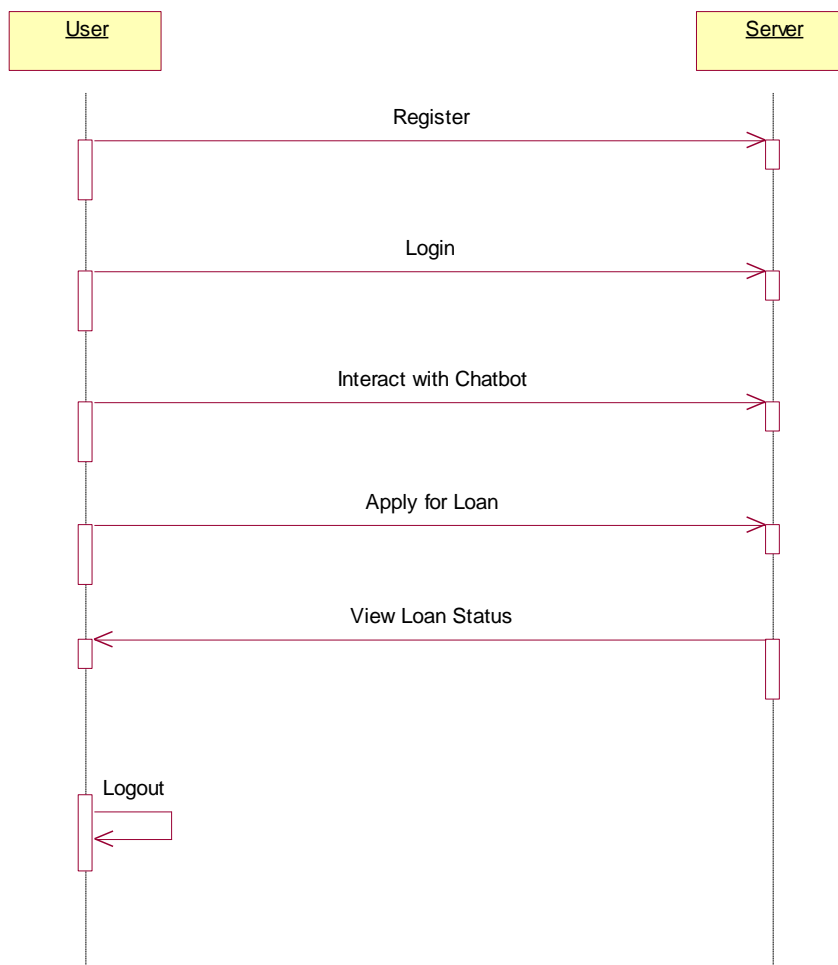
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains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





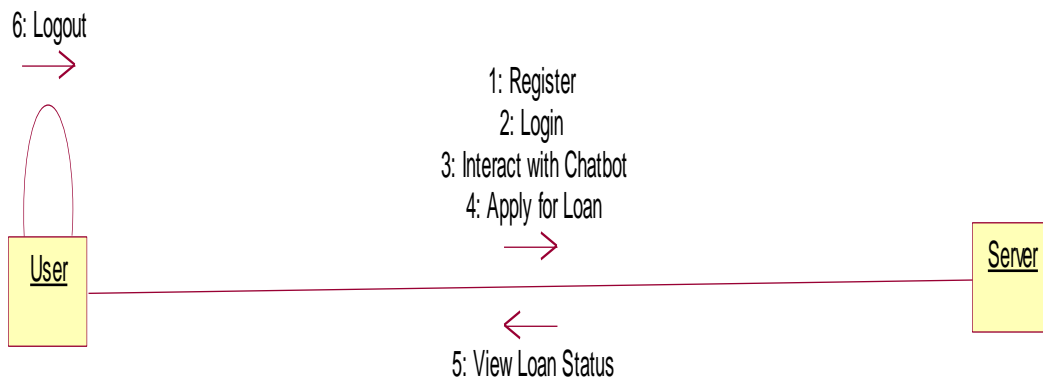
### 6.2.3 SEQUENCE DIAGRAM:

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.



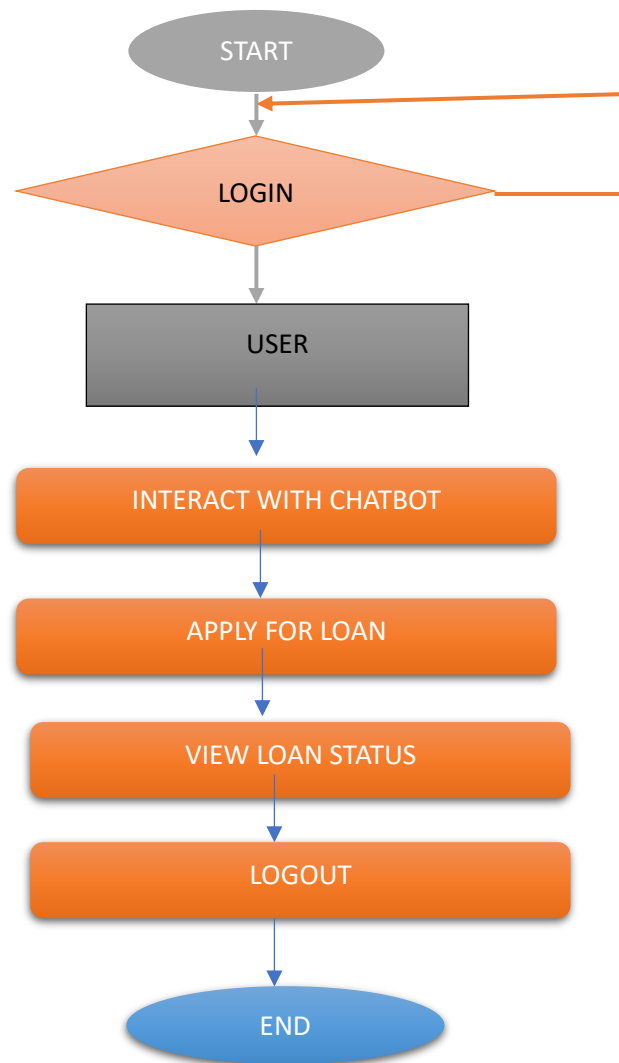
#### 6.2.4 COLLABORATION DIAGRAM:

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 can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



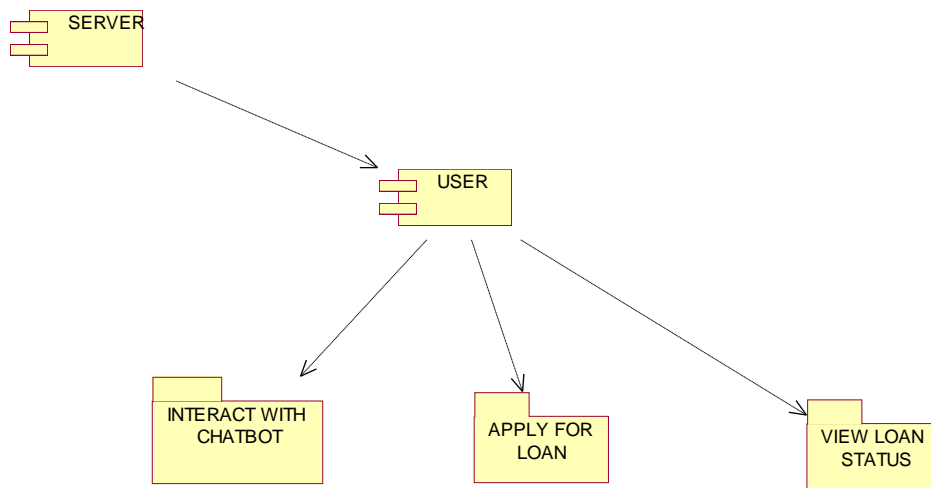
### 6.2.5 ACTIVITY DIAGRAM:

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 can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



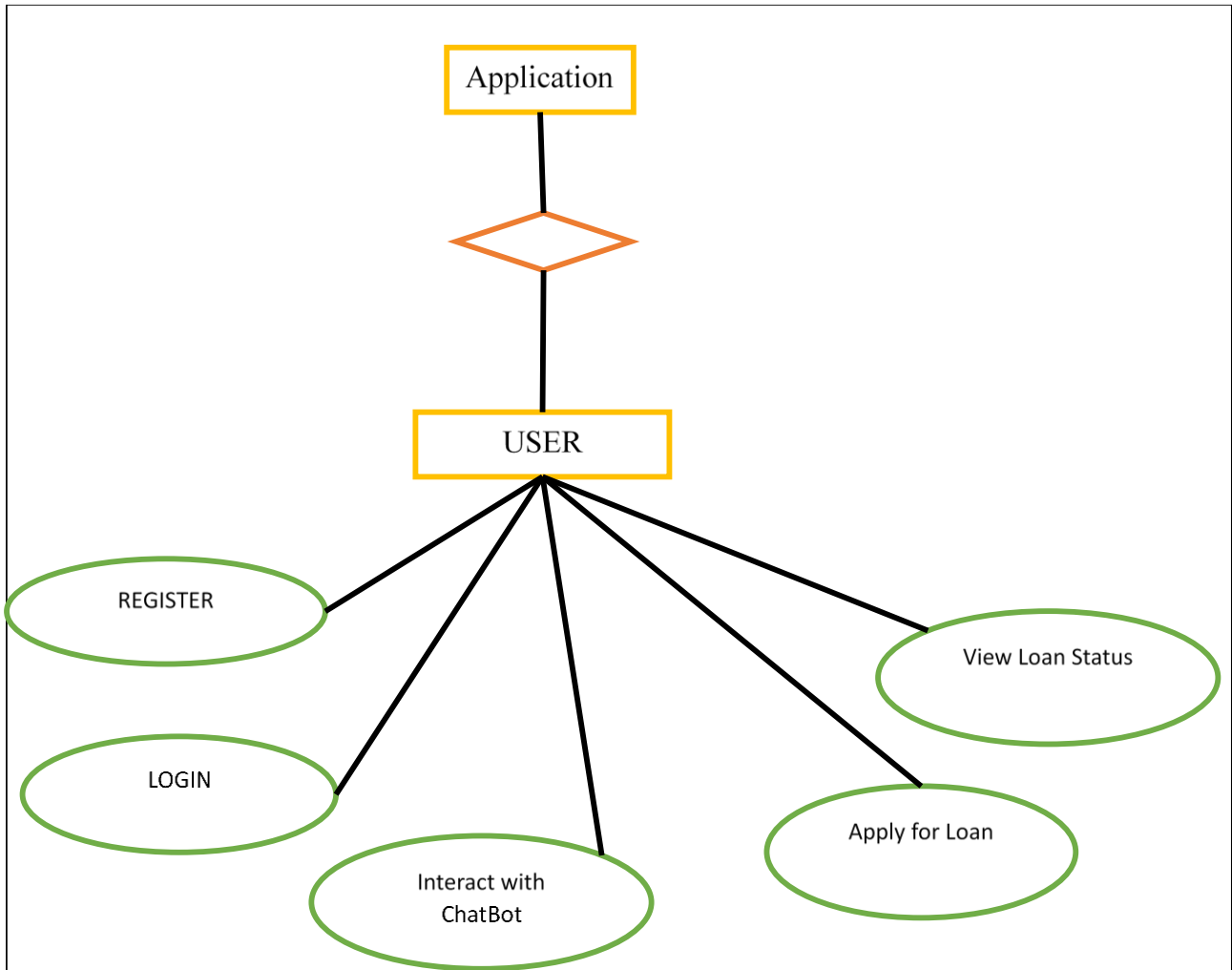
## 6.2.6 COMPONENT DIAGRAM:

Component Diagrams are used to show code modules of a system in Unified Modeling Language (UML). They are generally used for modeling subsystems. It represents how each and every component acts during execution and running of a system program. They are also used to show and represent structure and organization of all components. These code modules include application program, ActiveX control, Java Beans, backend databases, or some ASP programs. The component diagrams represent implementation of view models. The component diagrams are for representing interfaces and dependencies among software architecture. The word component simply means modules of a class that usually represents an independent subsystem.



## 6.2.7 ER DIAGRAM:

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.



## 6.2.9 DATASET

BankFAQs - Excel									
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do									
<div> <div>Clipboard</div> <div>Font</div> <div>Alignment</div> <div>Number</div> <div>Styles</div> <div>Cells</div> <div>Editing</div> </div>									
B10 To make telephonic (IVR) transactions more secure, all purchases/payments to merchants that you make using your mobile or telephone need to be authenticated with an additional password. While making an IVR									
Question	Answer	Class							
Do I need to enter # after keying in my Card number/ Ca	Please listen to the recorded message and follow the instructions while entering your card details.	security							
What details are required when I want to perform a secure IVR	To perform a secure IVR transaction, you will need your 16-digit Card number, Card expiry date, CVV nur	security							
How should I get the IVR Password if I hold an add-on card	An IVR password can be requested only from the registered mobile number and will be sent to the regist	security							
How do I register my Mobile number for IVR Password	Please call our Customer Service Centre and ensure that your mobile number is updated in our records.	security							
How can I obtain an IVR Password	By Sending SMS request: Send an SMS 'PWD<space>1234' to 9717465555 or to 5676712 from your regis	security							
Can I use the same IVR Password to perform multiple transacti	No, each IVR password can be used only for a maximum of 3 attempts (including decline attempts) withi	security							
Can I generate multiple IVR Passwords	No, only one IVR password can be generated at a time. Only when the first one is used / expires, can the security	security							
How do I register for IVR password	There is no registration process. However you will have to obtain a 3D Secure IVR password to perform	security							
What is IVR Password and how does it work	To make telephonic (IVR) transactions more secure, all purchases/payments to merchants that you mak	security							
In how much time will the IVR Password be delivered to my mol	3D Secure password is delivered instantly. It takes less than a minute. However, network congestion security	security							
Can the OTP be generated prior to the transaction	No, OTP can be generated only during the transactions.	security							
Can I continue using the Verified by Visa/MasterCard SecureCoc	Yes, you can continue using your secure password, OTP is an additional option provided for checkout.	security							
On which mobile number will I receive the OTP	OTP will be sent to the registered mobile number (updated in Credit card system).	security							
Is it necessary to register the card for an OTP transaction	Prior registration is not necessary for an OTP transaction.	security							
Is this option available on all the online transactions	This option is available on all websites which are participating in the VB/MSC/Portectbuy secure transa	security							
Can the OTP be used for multiple online transactions	OTP can be used only for a single online transaction.	security							
How will the OTP be sent to the customers	OTP will be sent only via SMS to the customers' registered mobile number.	security							
What is the validity period of the OTP	OTP is valid for a period of only 3 minutes.	security							
What happens if I shop at a Verified by Visa/ MasterCard Secure	You will be prompted for online registration for Verified by Visa/ MasterCard SecureCode through a pop	security							
Where can I shop with Verified by Visa	You can shop at participating online stores that have the Verified by Visa/ MasterCard SecureCode mark	security							
How will the online store know that I have Verified by Visa/ Ma	The participating online stores will automatically recognize your Visa card number if you have registered security	security							
How can I register / re-set password for VB/MSC if my mobile	You will have to first have to get the details updated in our system by calling our Customer Care Centre ( security	security							
Once I have registered, how long does it take for Verified by Vis	The Verified by Visa / MasterCard SecureCode gets activated for your HDFC Bank Card as soon as your r	security							
What will happen if I try to register again & generate an OTP wit	You will get a pop-up message stating Your earlier request is being processed. If you do not receive your security	security							
If my account is blocked, how do I unblock it	If your account gets blocked, please contact your nearest Customer Call Centre (for Credit Cards) and Pl	security							
How do I use Verified by Visa / MasterCard SecureCode when I	Once you have registered, you can use Verified by Visa/ MasterCard SecureCode service right away on p	security							

# **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**

#### **8.2 MODULE DESCRIPTION**

##### **USER**

The USER needs to register. After registration, user can get login credentials to allow into an application. After Login, user seen some operations i.e., interact with chatbot and apply for loan and finally view loan status like approved or not.

## 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<sup>1</sup>, 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 Learning :-**

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 programming 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 its 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.



**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	<a href="#">Release Notes</a>
Python 3.6.9	July 2, 2019	Download	<a href="#">Release Notes</a>
Python 3.7.3	March 25, 2019	Download	<a href="#">Release Notes</a>
Python 3.4.10	March 18, 2019	Download	<a href="#">Release Notes</a>
Python 3.5.7	March 18, 2019	Download	<a href="#">Release Notes</a>
Python 2.7.16	March 4, 2019	Download	<a href="#">Release Notes</a>
Python 3.7.2	Dec. 24, 2018	Download	<a href="#">Release Notes</a>

**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
<a href="#">Gzipped source tarball</a>	Source release		68111671e5b2db4ae77b9ab01b0f99be	13017663	
<a href="#">XZ compressed source tarball</a>	Source release		d33e4aa66097051c2eca45ee3604803	17131432	
<a href="#">macOS 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	6428b4fa75d3daff1a42c8a8ce08e6	34898416	
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4e936b243f	28082845	
<a href="#">Windows help file</a>	Windows		d63999573a2e96b2ac56cade6b477cd2	8131761	
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	9b00c3cf6d9ec0b0abe83184a4072ba2	7504391	
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	a702b4b0ad76d4bdc3043a383e563400	26880368	
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	26c61c081bd73ae8e53a3bd351b4bd2	1362904	
<a href="#">Windows x86 embeddable zip file</a>	Windows		9fab3bd18842879fda94133574139d8	6741626	
<a href="#">Windows x86 executable installer</a>	Windows		33cc602942a54446a3d0451476394789	25663848	
<a href="#">Windows x86 web-based installer</a>	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	

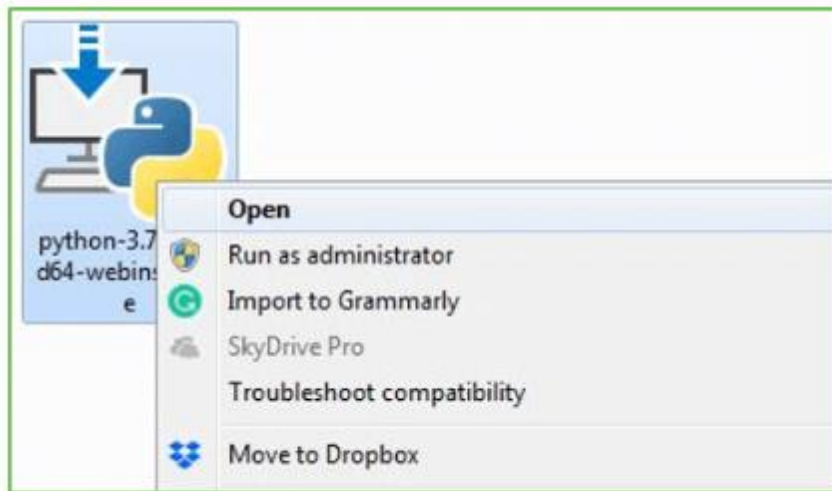
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, 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.



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



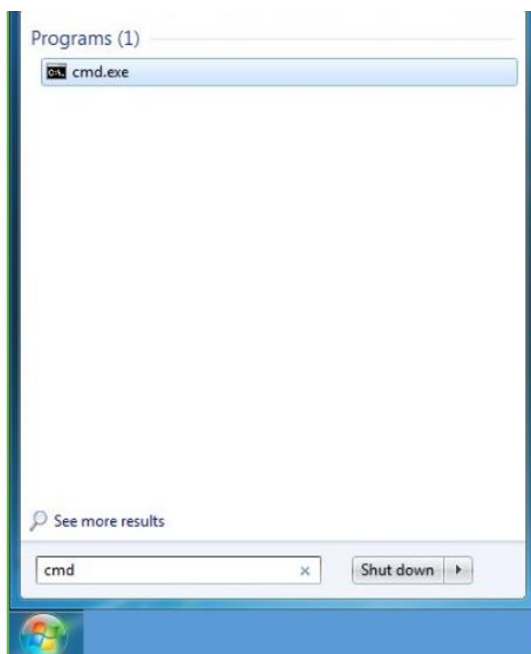
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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

### **Verify the Python Installation**

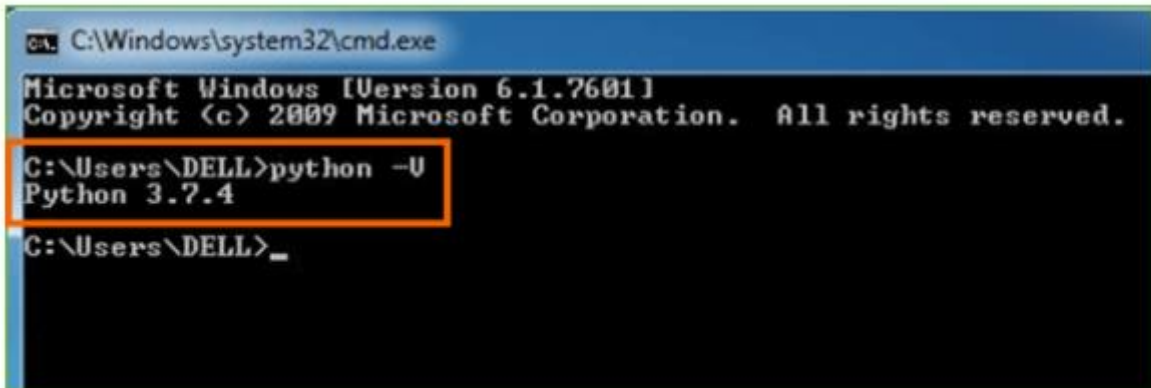
**Step 1:** Click on Start

**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>_
```

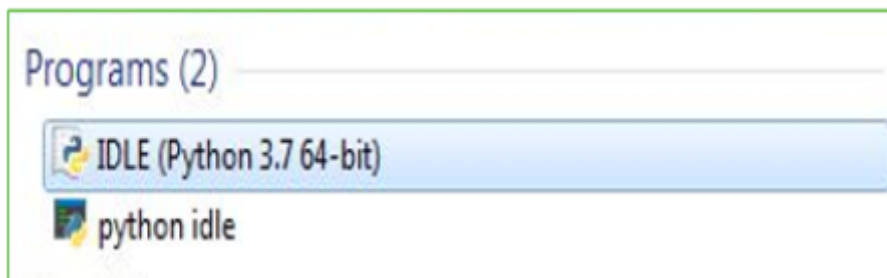
**Step 5:** You will get the answer as 3.7.4

**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**

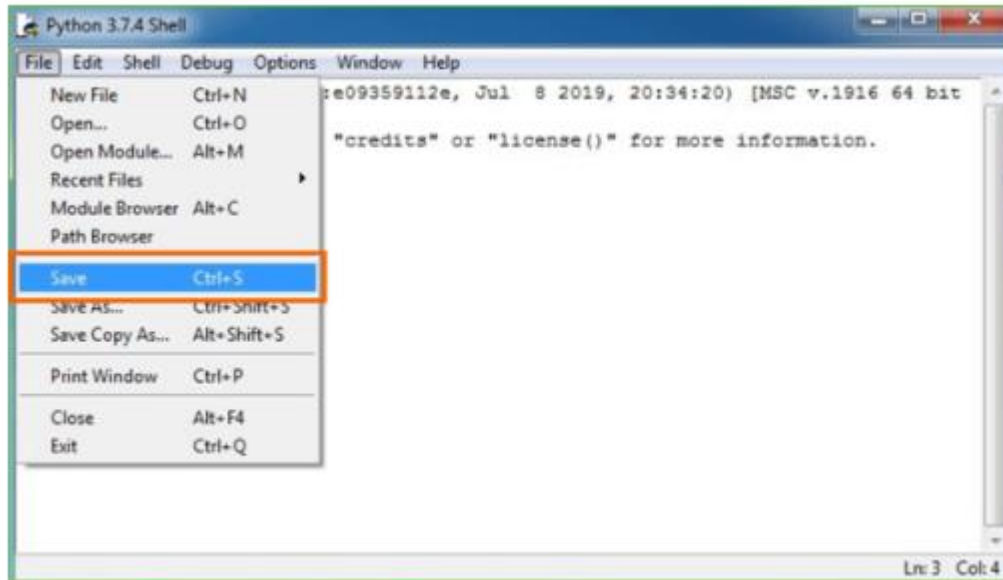
**Step 1:** Click on Start

**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**



**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

### **Manage.py**

```
#!/usr/bin/env python
import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Chatbot.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

### **index.html**

```
{% load static %}

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="description" content="" />
<meta name="keywords" content="" />
<title>Smarter Banking Chatfin</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="{% static 'style.css' %}" />
</head>
<body>
<div id="wrapper">
```



```

<div id="header">
    <div id="logo">
        <center><font size="4" color="yellow">Smarter Banking
Chatfin</font></center>
    </div>
    <div id="slogan">

    </div>
</div>
<div id="menu">
    <ul><center>
        <li class="first current_page_item"><a href="{ % url 'index' % }"><font
size="" color="yellow">Home</font></a></li>
        <li class="first current_page_item"><a href="{ % url 'AdminLogin'
% }"><font size="" color="yellow">Admin Login Here</font></a></li>
        <li class="first current_page_item"><a href="{ % url 'UserLogin' % }"><font
size="" color="yellow">User Login Here</font></a></li>
        <li class="first current_page_item"><a href="{ % url 'Signup' % }"><font
size="" color="yellow">New User Signup Here</font></a></li>
    </center></ul>
    <br class="clearfix" />
</div>
<div id="splash">
    
</div>
<br/>
<p align="justify"><font size="3" color="black" style="font-family: Comic Sans MS">
Smarter Banking Chatfin</p>
</body>
</html>

```

## AdminLogin.html

```
{% load static %}

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta name="description" content="" />

<meta name="keywords" content="" />

<title>Smarter Banking Chatfin</title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link rel="stylesheet" type="text/css" href="{% static 'style.css' %}" />

<script language="javascript">

function validate(formObj)

{

if(formObj.t1.value.length==0)

{

alert("Please Enter User ID");

formObj.t1.focus();

return false;

}

if(formObj.t2.value.length==0)

{

alert("Please Enter Password");

formObj.t2.focus();

return false;

}

return true;

}

</script>

</head>

<body>
```

```

<div id="wrapper">
<div id="header">
    <div id="logo">
        <center><font size="4" color="yellow">Smarter Banking Chatfin
</font></center>
    </div>
    <div id="slogan">

</div>
</div>
<div id="menu">
    <ul><center>
        <li class="first current_page_item"><a href="{% url 'index' %}"><font
size="" color="yellow">Home</font></a></li>
        <li class="first current_page_item"><a href="{% url 'AdminLogin'
%}"><font size="" color="yellow">Admin Login Here</font></a></li>
        <li class="first current_page_item"><a href="{% url 'UserLogin' %}"><font
size="" color="yellow">User Login Here</font></a></li>
        <li class="first current_page_item"><a href="{% url 'Signup' %}"><font
size="" color="yellow">New User Signup Here</font></a></li>
    </center></ul>    <br class="clearfix" />
    </div>

<div id="splash">
    
</div>

    <center>
<form name="f1" method="post" action="{% url 'AdminLoginAction' %}" onsubmit="return
validate(this);">
<br/>{% csrf_token %}<br/>
    <h3><b>Admin Login Screen</b></h3>

```

```
<font size="" color="red"><center>{{ data|safe }}</center></font>
```

```
<table align="center" width="40" >
```

```
<tr><td><font size="" color="black"><b>Username</b></td><td><input  
type="text" name="t1" style="font-family: Comic Sans MS" size="30"/></td></tr>
```

```
<tr><td><font size="" color="black"><b>Password</b></td><td><input  
type="password" name="t2" style="font-family: Comic Sans MS" size="30"/></td></tr>
```

```
<tr><td></td><td><input type="submit" value="Login">
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
<br/><br/>
```

```
</body>
```

```
</html>
```

# **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 satisfactory, as shown by successful 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 the software tester has knowledge of the inner workings, structure and language of the software, or at least its 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**

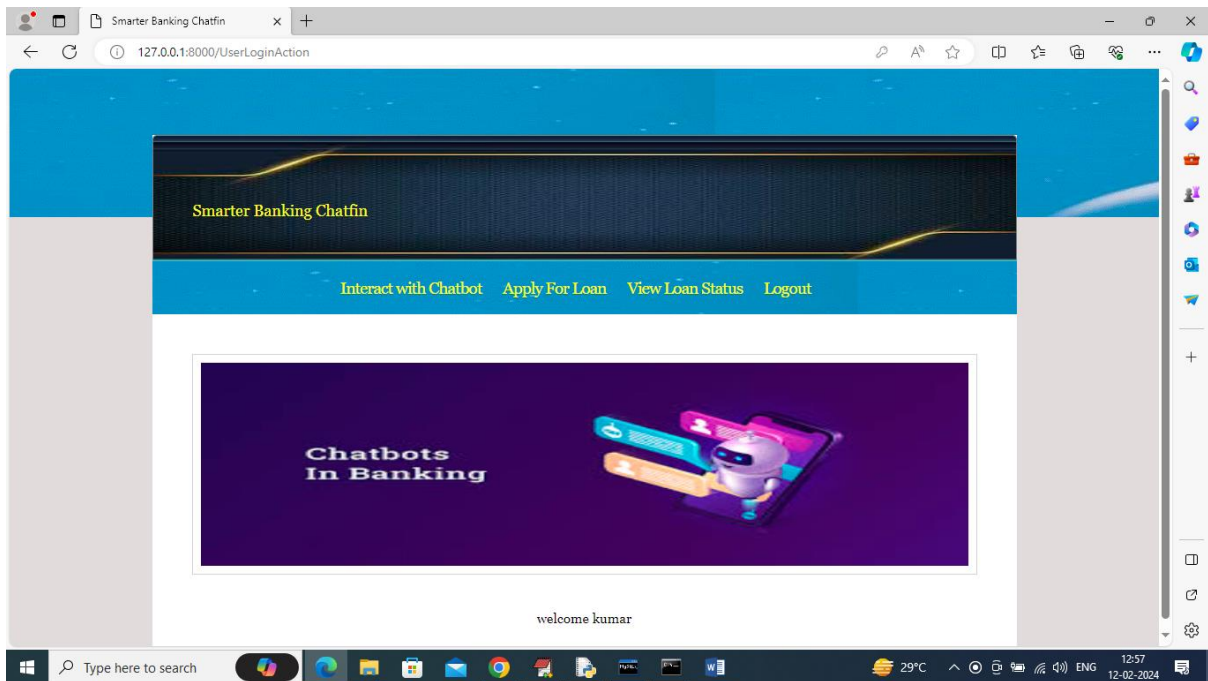
**Test case1 for Login form:**

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

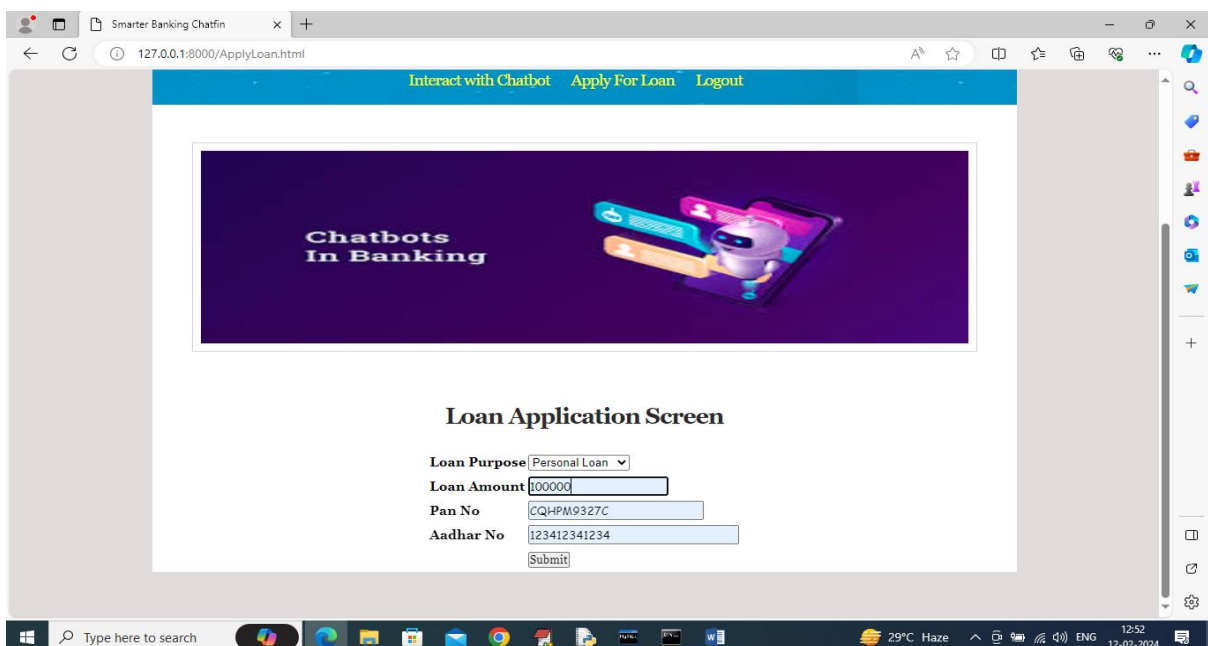


## 10.2 SCREENSHOTS

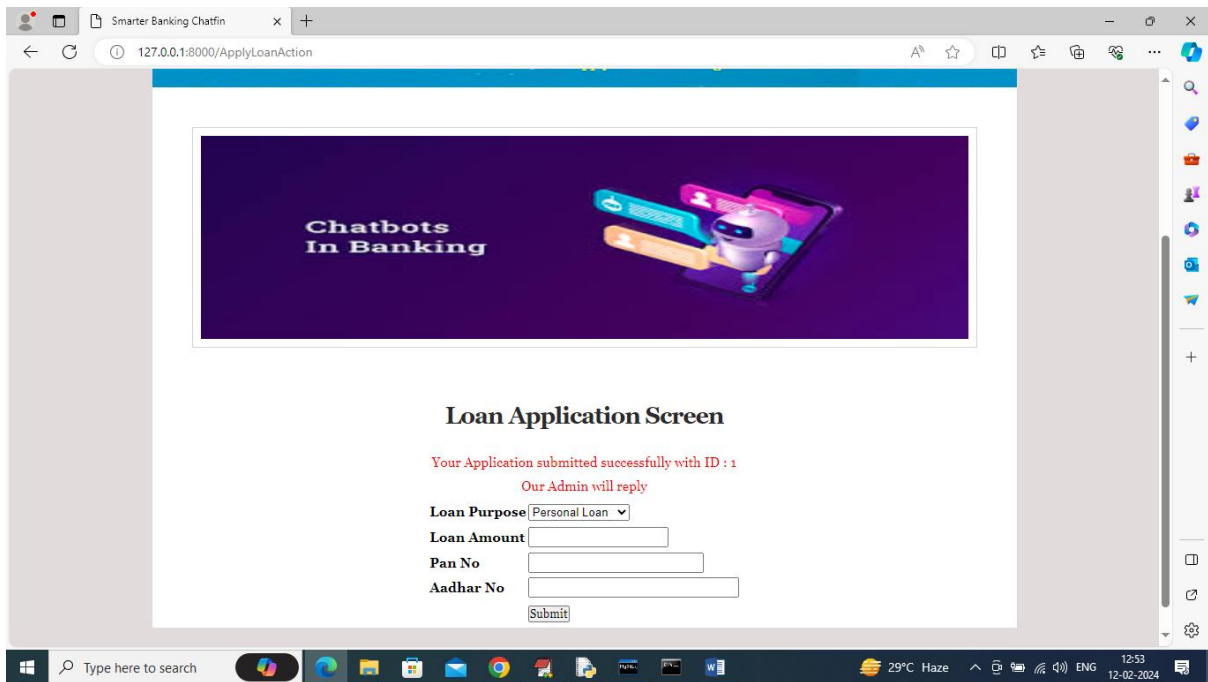
In below screen we have added extra module for users to submit loan application and this application will review by admin by using AADHAR or PAN no and then update status as Accepted or Rejected. Each loan application will be associated with unique ID. In below screen we are showing modified output. User can interact with Chatbot to take desired details regarding loan and then apply



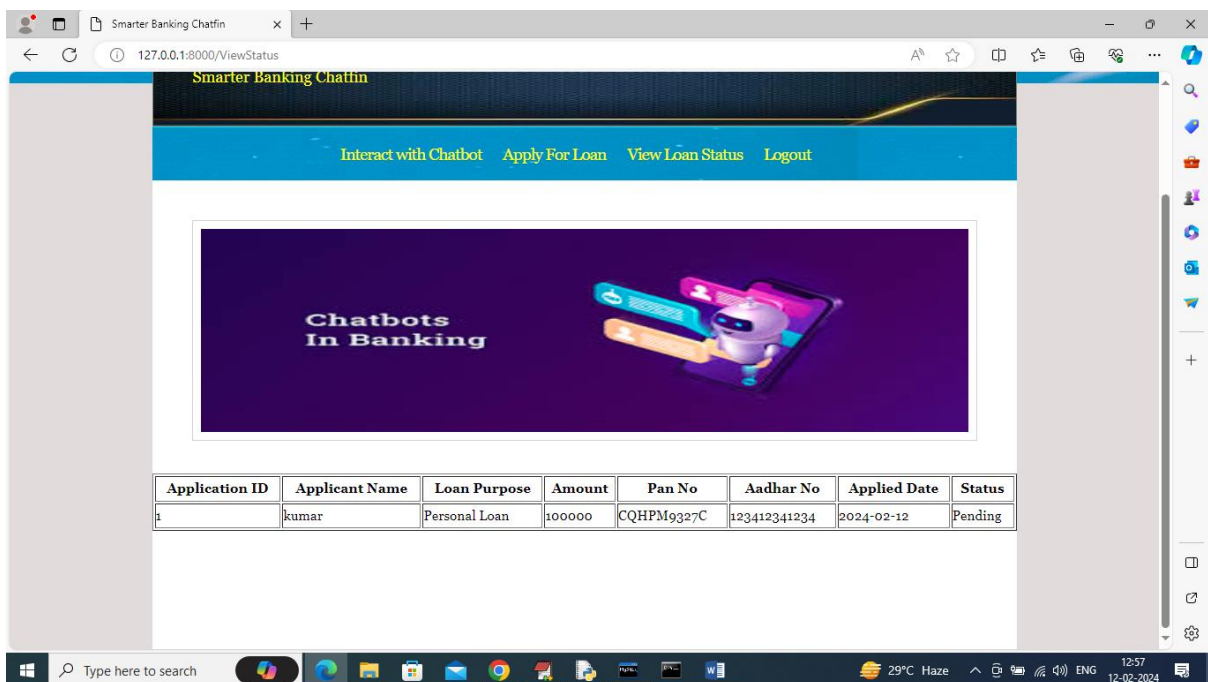
In above screen click on 'Apply For Loan' link to get below page



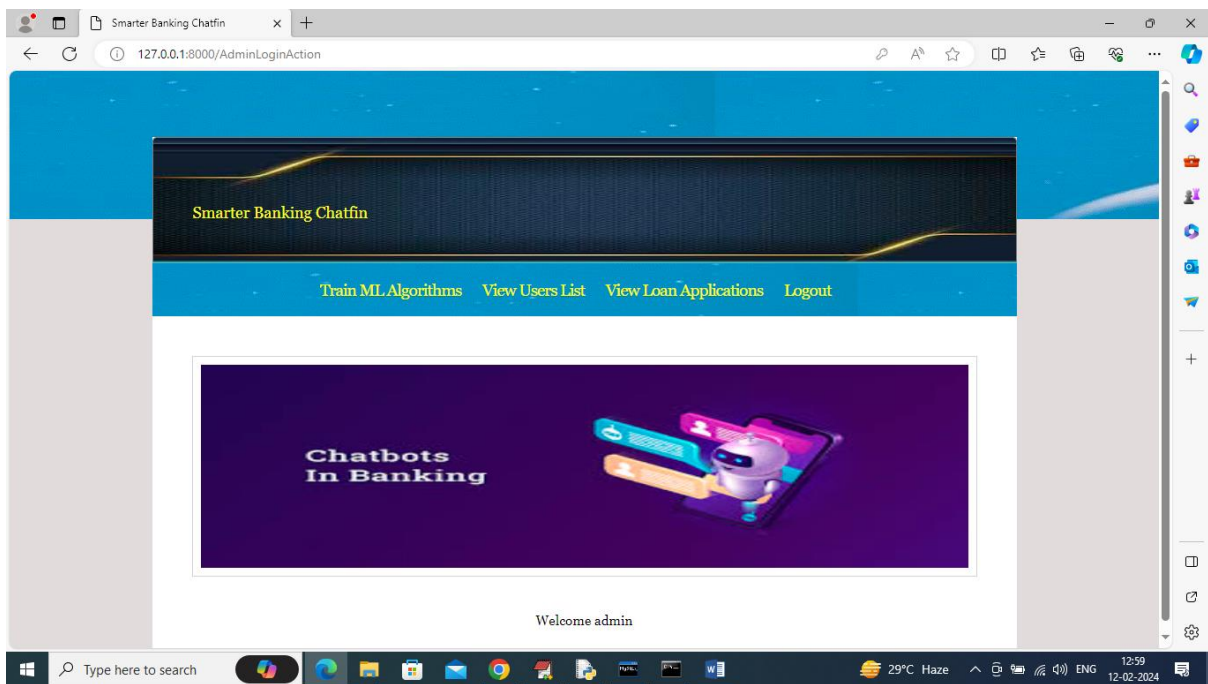
In above screen user can enter require loan details and then press button to submit loan and get below page



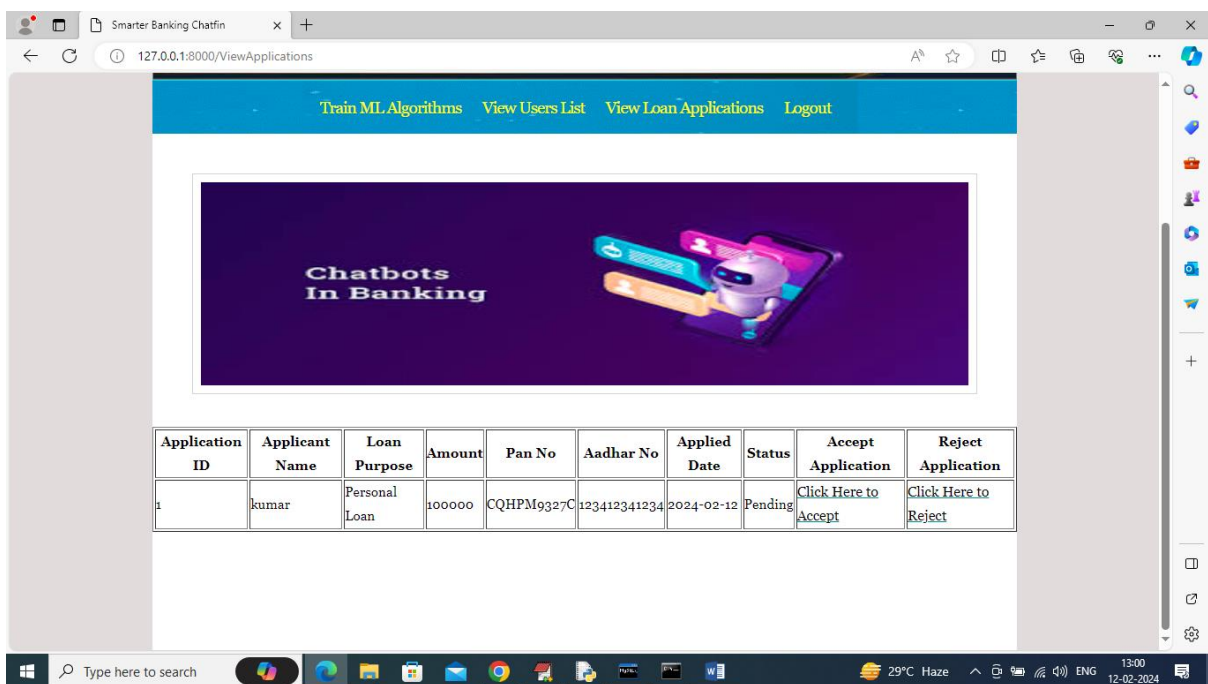
In above screen displaying application status and now click on ‘View Status’ link to get below status details



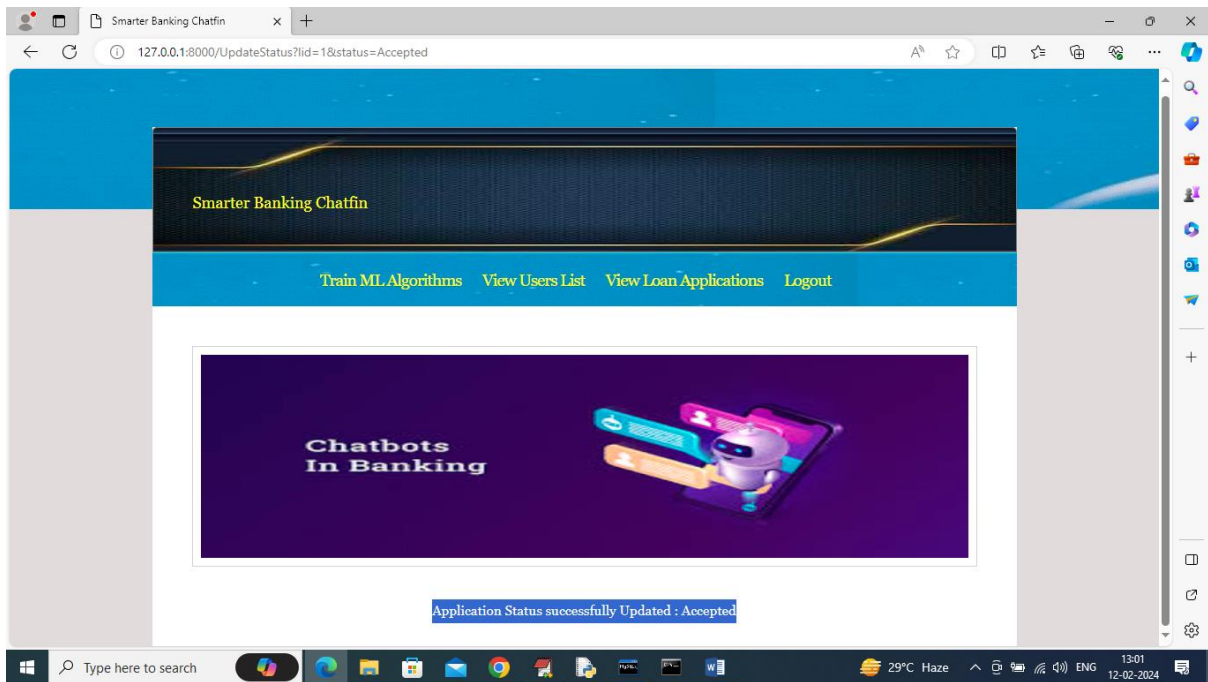
In above screen application status is Pending and now login as admin and then update status



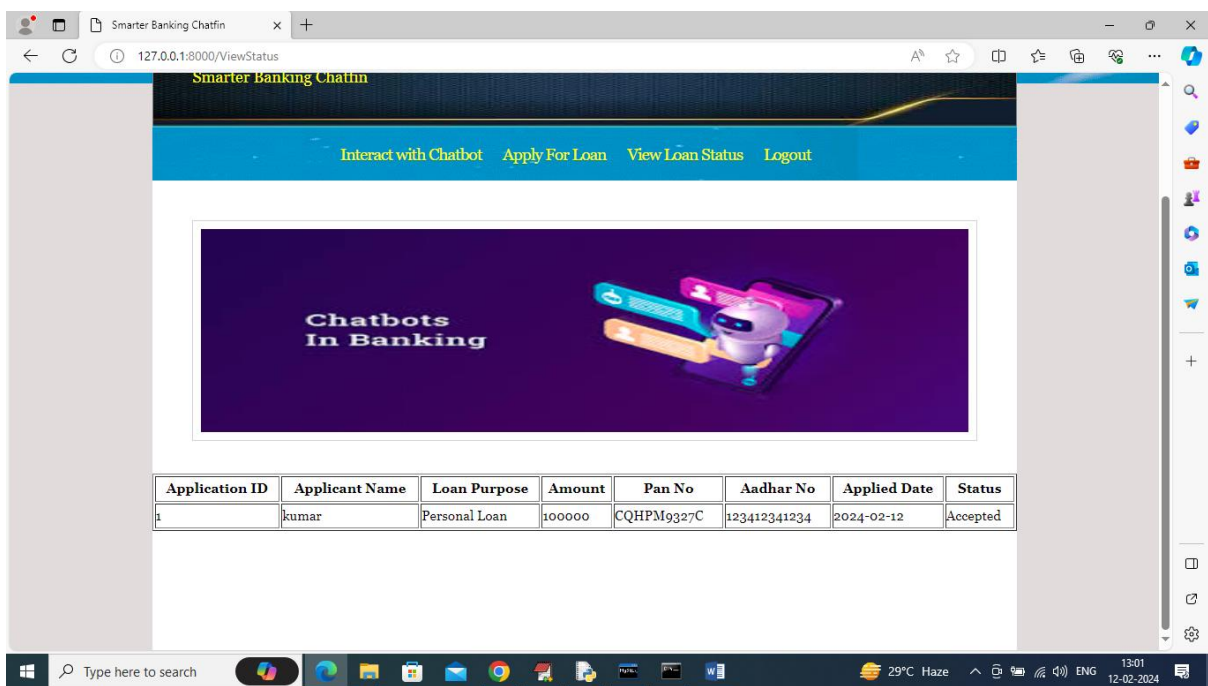
In above screen admin can click on ‘View Loan Applications’ link to get below page



In above screen admin will review all details and then click on either ‘Click Here to Accept’ or ‘Click Here to Reject’ link to accept or reject loan and now clicking on ‘Accept’ link to get below page



In above screen can see loan status updated to accepted and now user can login and view status



In above screen user can see Loan status as 'Accepted'.

Similarly by following above screens you can apply for loans also

# **CHAPTER 11**

## **CONCLUSION**

### **11.1 CONCLUSION**

the integration of the SMARTER BANKING CHATFIN chatbot represents a significant advancement in enhancing user interactions within banking websites. By leveraging the RASA framework and a meticulously crafted dataset, the chatbot effectively understands and processes customer queries, thereby streamlining the user experience. The incorporation of Python scripts for executing actions ensures that customer inquiries are handled efficiently, with real-time access to relevant information and prompt execution of necessary changes. This results in reduced interaction times and increased satisfaction, as users receive accurate and timely responses through an intuitive chat interface. Overall, SMARTER BANKING CHATFIN not only improves operational efficiency but also demonstrates a commitment to valuing and optimizing customer interactions in the banking sector.

Enhanced User Interaction: The chatbot reduces the time customers spend navigating the website by providing immediate, relevant answers and facilitating quick interactions.

- **Increased Efficiency:** Automated responses and actions streamline customer service processes, allowing for quicker resolutions and freeing up human agents for more complex inquiries.
- **Contextual Assistance:** Our custom-trained model ensures that the chatbot can handle diverse queries effectively, thanks to a well-defined dataset and tailored actions.
- **Database Integration:** The chatbot's ability to interact with the database to fetch or modify information based on user queries ensures that customers receive accurate and updated information in real time.
- **Improved Customer Experience:** By minimizing wait times and providing straightforward assistance, the chatbot enhances overall user satisfaction and engagement with the banking platform.

### **11.2 FUTURE SCOPE**

Performance can be enhanced by using various procedures of machine learning. Chatbots can be integrated with voice commands to enhance the user experience. More banking services can be

simplified by the use of a powerful chatbot. The quality of chatbots can be improved in the future by research in the field of banking. Security of chatbots will have to be improved as the security will be improved, people will be willing to share their information with the chatbot and also data collection will get easier. In the banking domain, contextual assistants need to cover more Indian languages thereby increasing the usability of chatbots. As technology evolves, future iterations could incorporate advanced natural language processing (NLP) techniques and machine learning algorithms to improve the chatbot's ability to understand and respond to complex queries with greater accuracy. Enhanced integration with various banking systems and third-party services will enable the chatbot to offer more comprehensive and personalized financial advice, manage transactions, and provide real-time updates on account activities. Additionally, incorporating features such as sentiment analysis and predictive analytics could allow the chatbot to anticipate customer needs and offer proactive support. With ongoing advancements in AI and machine learning, SMARTER BANKING CHATFIN has the potential to revolutionize customer interactions in the banking sector, making them more efficient, secure, and tailored to individual user preferences.

## **CHAPTER 12**

### **REFERENCES**

#### **REFERENCES**

- [1] Jiao, Anran. (2020). An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. Journal of Physics: Conference Series. 1487. 012014. 10.1088/1742-6596/1487/1/012014.
- [2] Fathima, Sasha & Student, Suhel & Shukla, Vinod & Vyas, Dr Sonali & Mishra, Ved P. (2020). Conversation to Automation in Banking Through Chatbot Using Artificial Machine Intelligence Language. 10.1109/ICRITO48877.2020.9197825.
- [3] Singh, Netra & Singh, Devender. (2019). Chatbots and Virtual Assistant in Indian Banks. Industrija. 47. 75-101. 10.5937/industrija47-24578.
- [4] Petr Lorenc, “Joint model for intent and entity recognition” in arXiv:2109.03221v1 [cs.CL] 7 Sep 2021
- [5] The Rasa documentation. [Online]. Available: <https://rasa.com/docs/rasa/2.x>
- [6] Django documentation. [Online]. Available: <https://docs.djangoproject.com/en/4.0/>