|  |  |  |
| --- | --- | --- |
| S.NO. | TOPIC | REMARKS |
| 1. | Introduction |  |
| 2. | Function of chatterbot |  |
| 3. | Objective |  |
| 4. | Development Environment |  |
| 5. | Project Analysis |  |
| 6. | Design of the system |  |
| 7. | System Analysis |  |
| 8. | Feasibility study |  |
| 9. | Cost/Benefit Analysis |  |
| 10. | Dataflow diagrams |  |
| 11. | Coding and screen |  |
| 12. | Testing |  |
| 13. | Security mechanisms |  |
| 14. | Cost estimation of the project |  |
| 15. | Pert chart |  |

**INTRODUCTION**

There is a lots of format for a user interface application software i.e. command line, graphical, and traditional ones. We can consider even voice as the user interface application. Basically, Chat-Bots system can be defined as the virtual agents who interact with the human beings. Chat-Bot System is an interface application which allows a user to interact with the text-based commands as well as the speech responses. It’s an integrated system with the web-pages which helps to utilize by the larger dividends.

A chatbot is an AI-based software designed to interact with humans in their natural languages. These chatbots are usually converse via auditory or textual methods, and they can effortlessly mimic human languages to communicate with human beings in a human-like manner. A chatbot is arguably one of the [best applications of natural language processing.](https://www.upgrad.com/blog/5-applications-of-natural-language-processing-for-businesses/)

Chatbots can be categorized into two primary variants – **Rule-Based and Self-learning.**

The Rule-based approach trains a chatbot to answer questions based on a set of pre-determined rules on which it was initially trained. These set rules can either be very simple or very complex. While rule-based chatbots can handle simple queries quite well, they usually fail to process more complicated queries/request.

**Function of ChatterBot**

When a user enters a specific input in the chatbot (developed on ChatterBot), the bot saves the input along with the response, for future use. This data (of collected experiences) allows the chatbot to generate automated responses each time a new input is fed into it.

The program chooses the most-fitting response from the closest statement that matches the input, and then delivers a response from the already known selection of statements and responses. Over time, as the chatbot engages in more interactions, the accuracy of response improves.

**Objective**

* Chatbots are mainly used to provide customer support.
* It helps in catering a huge amount of target audience at the same time 24/7
* Can Schedule meetings, Broadcast newsletters, auto-sequences
* [Acquire leads from Comments](https://botsify.com/acquire-users-from-comments)
* Create conversational forms and saving all the data on spreadsheets
* Chatbots are very intelligent. You train them once and they will communicate with your target audience in their language.  
  Multilingual chatbots have saved you from investing much on hiring different languages resources.
* If you are a company that’s functional all around the world, you get hands-on [chatbot](https://chatbotslife.com/) asap! Because, while you are asleep, your bot can entertain your customers anywhere in the world.

# 

# DEVELOPMENT ENVIRONMENT

# The minimum configuration required to run this project are:

1. Main processor : Pentium III (or) IV
2. RAM : 128MB
3. Hard Disk : 4.2GB
4. Clock Speed : 550 MHZ
5. System Bus Speed : 400 MHz
6. Cache RAM : 256 KB

# SOFTWARE ENVIRONMENT

Language : Python

Front End Design : Python

Module : tkinter, chatterbot, pillow, spacy

Operating System : Windows

**PROJECT ANALYSIS**

1. **PURPOSE OF THE PROJECT**
   * Provide a Single environment from where an administrator can interact with the system resources with in the network.
   * Provides Info regarding the no of services running on the system.
   * To provide an upgrade support for the system resources.
   * Provides interface to interact with the system parameters like network settings, OS Settings, Services, and Process etc.
   * Provides means of reporting the system setting to a person through email. There by an administrator can know the status of a systems throughout the network
2. **SOLUTION OF PROBLEMS**

* As Snack wizard is a tool. It requires the settings to be done by the administrator, in order to allow the remote access to the resources.
* Three different Categories are provided to know the current status of the services, drivers etc.
* The single application can be used to configure multiple versions of operating systems which is possible by allowing the application settings to get configures and show appropriate panel for the administrator.
* Updating of components can be done by accessing the components of the system as well as the resources available on the game .

## PERT CHART

PERT CHART stand for Program Evaluation Review Techniques, unlike bar chart PERT can be both a cost and a time management system. PERT is organized by events and activity or task. One advantage of the PERT chart is that it is a schedule device that also shows graphically which task must be completed before others are begun.   
Advantage of the pert chart is as follows:

* It force the manager to the plan
* It shows the interrelationship among the tasks in the project and in particularly it identified the critical path of the project, thus helping to focus on it.
* It exposes all the particularly all-possible parallelism in the activities and thus help in allocating the resources.
* It allows scheduling and simulation of alternative schedules.
* It enables the manager to monitor and control the project.

**Design of the system**

The design of an information system produces the details that clearly describe how a system will meet the requirements identified during system analysis. Systems specialists often refer to this stage as logical design, in contrast to the process of developing program software, which is referred to as physical design.

**Development of the software:-** Software development may install purchased software or they may develop new, custom designed programs. The choice depends on the each option, the time available to develop software and the availability of programmers. Generally it has been observed that programmers are part of permanent staff in a big organization. In smaller organization, without programmers, outside programming services may be hired or retained on a contractual basis. Programmers are also responsible for documenting the program, providing an explanation of how and why certain procedures are coded in specific ways. Documentations is essential to test the program and carry on maintenance once the application has been installed.

**SYSTEM ANALYSIS**

System analysis:-A System is the collection of interrelated components that works together to archive some common objective and system analysis is the specification as what the system required to do. It is a management technique which helps us in designing a new system or improving an existing system.

A system analyst should have various skills to effectively carry out the job specifically. These skills can be divided into two categories. These are interpersonal skills and technical skills interface of the analyst with people in business. They are useful in establishing trust, resolving conflict, and communication information. Technical skills on the other hand, focus on the procedures and techniques for operations analysis, system analyst and computer science.

The interpersonal skills which are relevant to systems work are following:-

1. Communication: - Communication is not just reports, telephonic conversations, and interview. It is people talking, listening, felling, and reacting to one another, their experience and reaction. Some indicators of one another, their experience and reaction to the, their experience and reactions. Some indicator of climate of closed communication is defensive memos, excessive correspondence, and failure to speak up for fear of being identified. Therefore, opening communication channels are a must for system development.
2. Understanding: - Identified problems and assessing their remedies is one of the attributes of good system analyst. A system analyst should have the grasp of company goals and objective.
3. Teaching: - A system analyst should educate people in the use of computer systems, selling the system to the user and giving the support when needed.
4. Selling: - a system analyst should have selling ideas and promoting innovations in problem solving using computer.

For system analysis we know that we operate in a dynamic environment where way of life is. To construct a system the following key elements must be considered:

**(1) Outputs and inputs**: A major objective of a system is to produce an output that has value to its user. Whatever the nature of the output (goods, services, or information), it must be in line with the expectations of the intended user. Inputs are the elements (material, human recourses, information) that enter the system for processing. Output is the outcome of processing. A system feeds on input to produce output in much the same way that a business brings in human, financial, and material resources to produce goods and services. Output is a first step in specifying the nature, amount, and regularity of the input needed to operate a system. Input and processing design follow.

Compare Output

Management Control

Services

Human re

Transformation

Standard of performance

Input

Processing Output

**(2)** **The processor(s):** The processor is the element of a system that involves the actual transformation of input into output. It is the operational component of a system. Processor may modify the input totally or partially, depending on the specifications of the output.

**(3) Control:** The control elements guide the system. It is the decision-making subsystem that controls the pattern of activities governing input, processing, and output. In an organization context, management of the bank as a decision-making body controls the inflow, handling, and outflow of activities that affect the welfare of the business. In a computer system, the operating system and accompanying software influence the behavior of the system.

**(4) Feedback:** Feedback may be positive or negative, routine or informational. Positive feedback reinforces the performance of the system. It is routine in nature. Negative feedback generally provides the controller with information for action. In system analysis, feedback is important in different ways. During analysis, the user may be told that the problems in a given application verify his/her initial concerns and justify the need for change. Another form of feedback comes after the system is implemented. The user informs the analyst about the performance of the new installation. This feedback often results in enhancements to meet the user’s requirements.

**(5)Environment:-**The environment is the “super system” within which library operates. It is the source of external elements that impinge on the system. In fact, it often determines how a system must function such that the Bank’s environment, consisting of customers and others.

**(6)Boundaries and Interface:** A system should be defined by its boundaries –the limits that identify its components, process, and interrelationships when it interfaces with another system.

There are generally eight types of System analysis technique:-

**1. Requirement technique**:- Requirement determination is generally done through Extensive study of the system includes the understanding of the goal. Process And constraints of the system for which information are designed. Several Forms are also designed and illustrated in the text of system analysis. There are no straight forward algorithms to elicit the requirement from the user. It is an iterative process, which the analyst use while interviewing several user groups. It is art rather then science.

**2. Diagrammatic technique**:- Data flow document flow diagrams represent perhaps most the most widely used diagramming technique of the system analyst. The document flow diagram graphically represents the various documents that flow across the system.: the information carried by the paper document must be generated and proceed by proposed information system.

**3. Data flow diagram(DFD)**:- Data flow diagram is a powerful diagram that can be used to document the information flow. It also presents itself to be broken down in top down fashion .At the top level ,data flow are represented at the very abstract aggregate level .Each component of the data flow is further broken down to different levels, so that at each level. We have just a few entities to concatenate on , DFD have developed a representation scheme to represent data store. Process(where some changes are made to system) and entities(the player in the game) and the actual information flows.

**4. Data Dictionary**:-Another powerful tool that is extensively used in system

analysis in the data dictionary. DDs provides a detailed reference to every data item—the different names by which the item is represented in different program modules, different data structure used to represent the item in different modules. The module where the date item is generated ,where it is stored destroyed .In essence it provides a quick snapshot of every data item is generated ,where it is stored and destroyed .In essence it provides a quick snapshot of every date item is used by the information system. In essence it provides a quick snapshot of every data item used by the information system .It is very useful for consistency checks. System modification and completeness checking.

While these techniques are general in nature and by the analyst in the different stages of the system life-cycle the following are specific to some of the steps of the system life-cycle.

**5.Feasibility Report**:- A typical structure of the feasibility report will be under:-

A preamble that sets the stage for the project followed by goals statement that quantity precisely the goals of the proposed information system. This is followed by a short narrative that describe in unambiguous yet jargon free Language the present system. This is understandable to any intelligent person not necessary a computer professional or a even a computer literate .the proposal alternative are then describe once again in a reasonably jargon free language . Being a feasibility study the alternative are unlikely to be detailed to the full extent. Until full system is developed in its entirety, the full detail are unlikely to be known. Yet we can’t go ahead with the final system without doing a feasibly analyses. The detail of the system to be built may bum ford gradually from understood by the user as well as analyst. Based on ‘sketchy’ design of the proposed alternatives, an order of magnitude cost benefit study is preferred. The end-user decides a particular alternative that is worked out in detail for further implementation. The detail design phase starts here.

**6. Detailed design:** - The detailed specifications are worked out followed by hardware/software plan. This constitutes system designs which once again need to be whetted by the user. Once this is done detailed system design starts. Effectively the analysis phase ends here and the design phase begins. It may involve substantial effort on the part of technical system analyst, hardware, software, communication specialists etc. a major component of the detailed system design is the database design actual coding is undertaken after the database design is completed.

1. **Database Design**: - DBMS permits efficient storage and manipulation of data files they do not cater to the structuring of the database themselves .There is the need for the right abstraction of data into the database so that any update/query operation captures the spirit of the data stored in the database. Normalization is used which leads to the decomposition in such a way that no information is lost due to processing of data. Database theory details further degrees of normalization including 4NF and 5NF. While theoretically sound, such further refinement add (?) like, if, any, to data modeling real world data. Since our text is primarily on information system and not on database theory we will not further elaborate an advanced normalization.

**8. System implementation**:- System implementation includes the detailed design of the process, their validation and thought checking. while the formal methods of proving program correctness are evolving ,they are still not useful to test out large commercial software to help information system planning .Many of the analysts use experimental version using what is known as “Parallel runs ”. Here both the current system and proposed new system are run in parallel for a specified time period and the current system is used to validate the purposed system.

**Feasibility study:🡪**

Feasibility is the determination of the whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. These types of study determine if a project can and should be taken. Once it has been determination that a project a feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements. Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report . The contents and processed, postpone or cancel the project. Thus, since the Feasibility study may lead to the commitment of large resource, it becomes Necessary that it should be conducted competently and that no fundamental errors of judgment made.

Types of feasibility:🡪

1. Technical feasibility
2. Operational feasibility
3. Economical feasibility
4. Social feasibility
5. Management feasibility
6. Legal feasibility
7. Time feasibility
8. **Technical feasibility**:- Technical feasibility involves determining whether or not a system can actually be constructed to solve the problem at end. This is considered with specifying equipment and software that will successfully satisfied the user requirement. The technical needs of the system may vary considerable ,but might include:-

🡪The facility to produce outputs in a given time.

🡪Response time under certain condition.

🡪Ability to process a certain volume of transaction at a particular Speed.

🡪Facility to communicate data to distant location.

1. **Operational feasibility**: - Proposed projects are of course beneficial only if they can be turned into information system that will meet the organization’s operating requirement. Simply started, these tests of feasibility ask the system will work when developed and installed. There are major barrier for implementation. It is related to human organizational and political aspects. The points to be considered are :-

🡪 What change will be brought with the system?

🡪 What organizational structures are distributed?

🡪 What new skills will be required? Does the existing staff member have?

These skills? If not, can they be trained in the due course of time?

1. **Economical feasibility**: - Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with cost. If benefit out weights costs, a decision is taken to design and implementation the system.
2. **Social Feasibility**: - Social Feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination Typical examines the probability of the project being accepted by group Directly affected by the proposed system change.
3. **Management feasibility**:- It is a determination of whether a proposed project will be acceptable to management. If management does not accept a project or gives a negligible support to it, the analyst will tend to view the project as a non-feasible one.

**6. Legal feasibility**: - Legal feasibility is a determination of whether a proposed project infringes on known acts, Statutes, as well as any pending legislation. Although in some instance the project might appear sound, on closer investment it may be found to infringe on several legal areas.

**7. Time feasibility**: - Time feasibility is a determination of whether a project can be implementing fully with in a stipulated time frame .If a project takes too much time it is likely to be rejected.

**COST/BENEFIT ANALYSIS**:

In developing cost estimates for a system, we need to consider several cost elements.

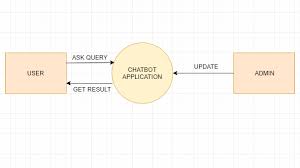
1. **Hardware costs** relate to the actual purchase or lease of the computer and peripherals. Determining the actual cost of hardware is generally more difficult when various users than for a dedicated stand-alone system share the system. In some cases, the best way to control for this cost is to treat it as an operating cost.
2. **Personnel costs** include EDP staff salaries and benefits as well as pay for those involved in developing the system. Once the system is installed the costs of operating and maintaining the system become recurring costs.
3. **Facility costs** are expenses incurred in the preparation of the physical site where the application or the computer will be in operation.
4. **Operating costs** include all costs associated with the day-to-day operation of the system; the amount depends on the working hour, the nature of the application.
5. **Supply costs** are variable costs that increase with increased use of paper, ribbon, disks, and the like.

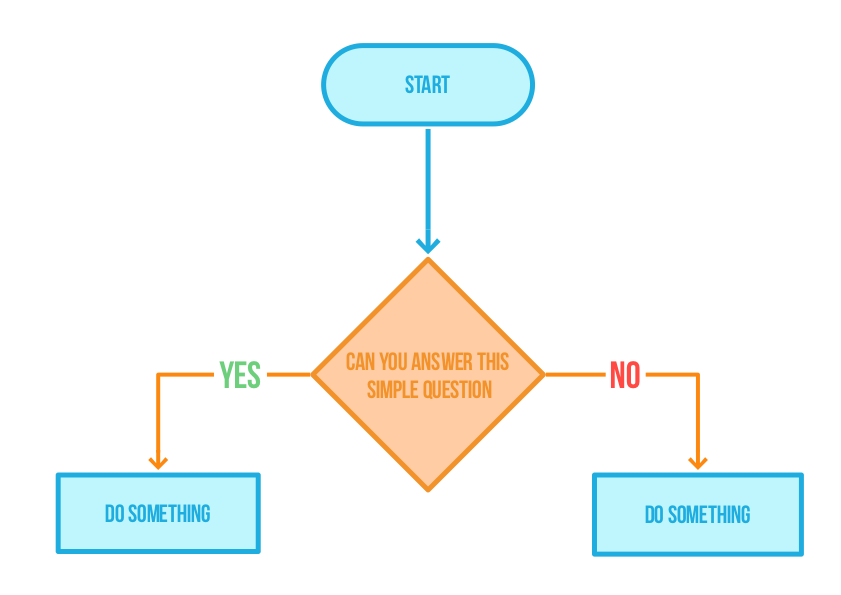
**Procedure for Cost/Benefit Analysis:**

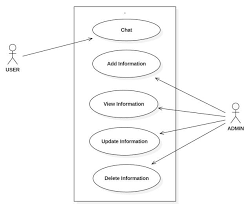
1. Identify the costs and benefits pertaining to a given project.
2. Categorize the various costs and benefits for analysis.
3. Select a method of evaluation.
4. Interpret the results of the analysis.
5. Take action.

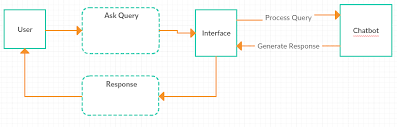
# DATA FLOW DIAGRAMS

Data flow Diagram is concerned with designing a sequence of functional transformation that converts system inputs into the required outputs. The design is represented as data flow diagram. These diagrams illustrate how data flow through a sequence of functional transformations.



****

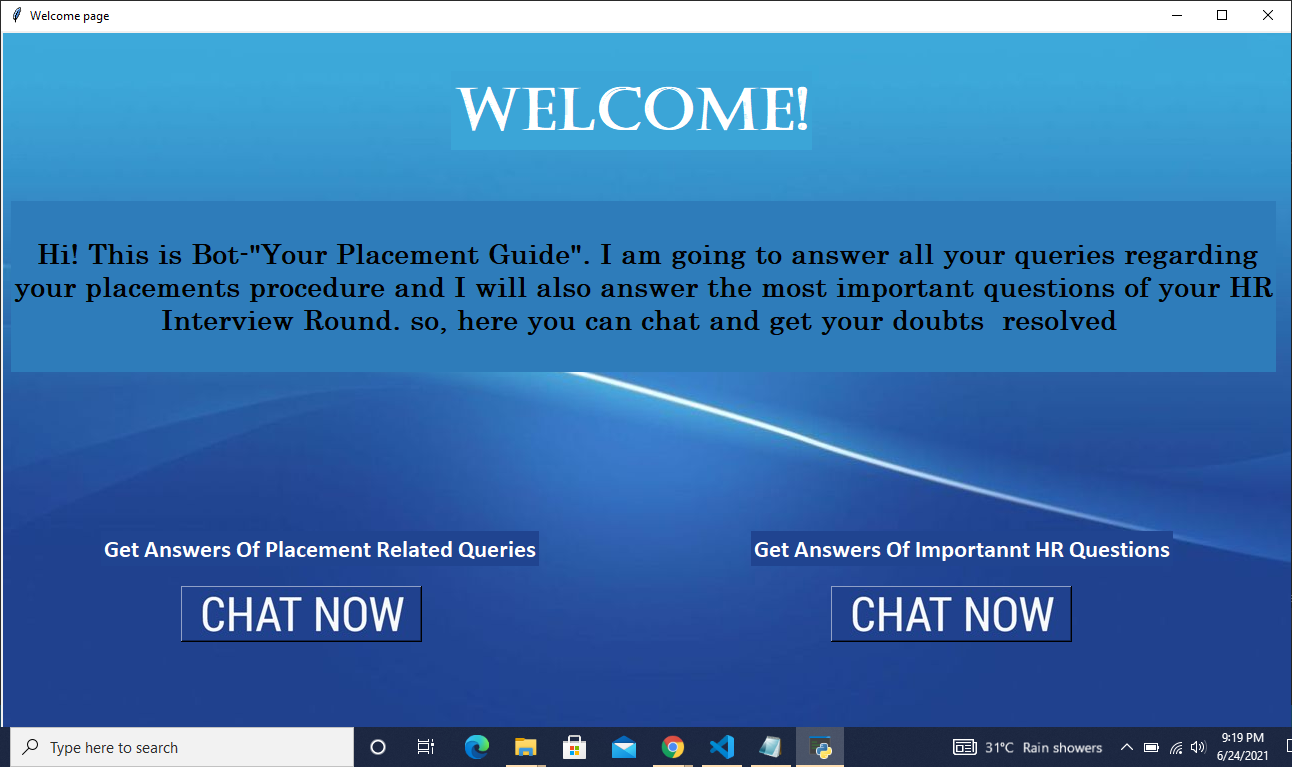
****

****

**Probabilistic acceptance criterion:**

Based on the EER obtained from the previous section, we found that the information of x-coordinate and y-coordinate are the most dependable followed by the pressure and velocity. A probabilistic acceptance criterion has been proposed in this section to accept a signature with condition: X1Y1(PcV) = ACCEPT

**Coding and Screen**

****

**from tkinter import\***

**from PIL import Image,ImageTk**

**rt=Tk()**

**rt.title('Welcome page')**

**rt.geometry('1290x800+150+0')**

**rt.maxsize(1290,800)**

**rt.minsize(1290,800)**

**#--------function---------**

**def btn1():**

**rt.destroy()**

**import chat\_imp**

**def btn2():**

**rt.destroy()**

**import chat\_doubts**

**#-------background---------**

**img=Image.open("images/wallpaper2you\_245333.jpg")**

**photo=ImageTk.PhotoImage(img)**

**b\_img=Label(rt,image=photo)**

**b\_img.place(x=0,y=0)**

**title=Label(rt,text="WELCOME!",font="Castellar 45 bold",bg="#3ba5d7",fg="white")**

**title.place(x=450,y=40)**

**title=Label(rt,text='\n Hi! This is Bot-"Your Placement Guide". I am going to answer all your queries regarding your placements procedure and I will also answer the most important questions of your HR Interview Round. so, here you can chat and get your doubts resolved \n', wraplength="1300",bg="#2e7cba",fg="black",font="century 20 bold")**

**title.place(x=10,y=170)**

**pic=Image.open("images/360\_F\_400911587\_8hmMcZE9Soxm5LS8xa9XbgyrHBc65334.jpg")**

**Photo=ImageTk.PhotoImage(pic)**

**l=Label(rt,text="Get Answers Of Placement Related Queries",bg="#204490",fg="white",font="Calibri 18 bold" )**

**l.place(x=100, y=500)**

**b1=Button(rt,image=Photo,bg="#20418e",command=btn1)**

**b1.place(x=180,y=555)**

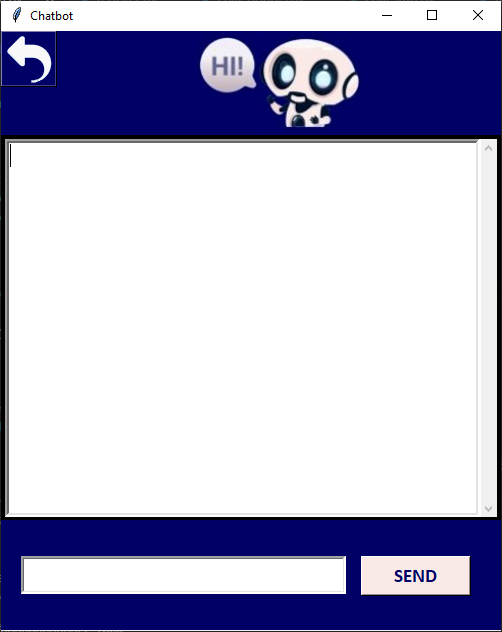
**l1=Label(rt,text="Get Answers Of Importannt HR Questions",bg="#204490",fg="white",font="Calibri 18 bold" )**

**l1.place(x=750, y=500)**

**b2=Button(rt,image=Photo,bg="#20418e",command=btn2)**

**b2.place(x=830,y=555)**

**rt.mainloop()**

****

****

**from tkinter import \***

**from PIL import Image, ImageTk**

**from chatterbot import ChatBot**

**from chatterbot.trainers import ListTrainer**

**root=Tk()**

**root.title('Chatbot')**

**root.geometry('500x600+400+50')**

**root.maxsize(500,600)**

**root.minsize(500,600)**

**bot=ChatBot("Bot")**

**trainer=ListTrainer(bot)**

**ques=[**

**"Hi",**

**"Hello my friend",**

**"Hello",**

**"Hey",**

**"Who are you?",**

**"I'm chatterBot-Your placement guide",**

**"Tell us about yourself",**

**"This is your big chance to make a good first impression. A brilliant answer here would be a professional snapshot of yourself – your core areas, studies and skills (according to the company concerned) and what you’re trying to achieve in your career. Briefly outline your personal life, but remember – the recruiters aren’t really interested in these things. Prepare a two-minute reply, which works as an ice-breaker and can lead directly towards the job / company you’re applying for.",**

**"About yourself",**

**"This is your big chance to make a good first impression. A brilliant answer here would be a professional snapshot of yourself – your core areas, studies and skills (according to the company concerned) and what you’re trying to achieve in your career. Briefly outline your personal life, but remember – the recruiters aren’t really interested in these things. Prepare a two-minute reply, which works as an ice-breaker and can lead directly towards the job / company you’re applying for.",**

**"Introduce yourself",**

**"This is your big chance to make a good first impression. A brilliant answer here would be a professional snapshot of yourself – your core areas, studies and skills (according to the company concerned) and what you’re trying to achieve in your career. Briefly outline your personal life, but remember – the recruiters aren’t really interested in these things. Prepare a two-minute reply, which works as an ice-breaker and can lead directly towards the job / company you’re applying for.",**

**"Strength and weakness",**

**"Questions about strengths and weaknesses are typical in interviews. All you have to do is to pick and emphasize strengths that are relevant to the job you’re applying for. As far as your weaknesses are concerned, choose to talk about those that are not very important for the job.",**

**"What are your strength and weaknesses?",**

**"Questions about strengths and weaknesses are typical in interviews. All you have to do is to pick and emphasize strengths that are relevant to the job you’re applying for. As far as your weaknesses are concerned, choose to talk about those that are not very important for the job.",**

**"What are you doing to overcome your weaknesses?",**

**"Keep it short and simple. Say that you are working hard to get rid of your weaknesses.",**

**"What are your hobbies?",**

**"State what excites you or something that you love doing the most in your free time. Remember – don’t lie. You will get caught out sooner or later and your prospective employer won’t be impressed if they find out that you had lied to them from the very start.",**

**"Why should we hire you for this job?",**

**"Tell the employer something unique about yourself, something other candidates can not offer.",**

**"What would you consider as your biggest achievement till date?",**

**"We all have achievements in life, big or small. Always remember to choose an achievement that is relevant to the job.",**

**"Biggest achievement",**

**"We all have achievements in life, big or small. Always remember to choose an achievement that is relevant to the job.",**

**"What is your dream job?",**

**"Be very cautious while answering this. Whatever you answer, try to align it with the job you have applied for. You can say, “I want to be in the senior management of this (XYZ) company and this job is the gateway for the same.” This will show your dedication and the interviewer would get an idea that you are someone who would like to stay.",**

**" Where do you see yourself five years down the line?",**

**"I would like to work hard and have a managing role in five years time. However, I understand that I need to learn a lot, and I believe that this position is a perfect starting point.” A great answer to impress the interviewer. It sort of builds a connection between your goals and the company.",**

**"What motivates you?",**

**" State something that you genuinely believe in. Also, link it with the role that you have applied for.",**

**"Are you a team player?",**

**"Never say a no. You always have to work with a team, irrespective of domain. A company would always like to have people who would work well in a team.",**

**"Give us an example of a time when you handled a major crisis/difficult situation. You can talk about something that you handled well in pressure.",**

**"What do you know about our organization?",**

**"Never say that a friend told me or your consultant told you about it. Give an impression to the recruiter that you knew about the company right from start and have done some solid research before coming. You can say that how you were always interested in advertising and “XYZ” company is what you always dreamt of.",**

**"What do you know about us?",**

**"Never say that a friend told me or your consultant told you about it. Give an impression to the recruiter that you knew about the company right from start and have done some solid research before coming. You can say that how you were always interested in advertising and “XYZ” company is what you always dreamt of.",**

**"What salary are you expecting?",**

**"While answering this question, try to emphasize that the money isn’t the deciding factor. And if you have a number in your mind, put it up in a smart way. It would be good to have some statistics to back it up.",**

**"Are you willing/open to change in your role?",**

**"It is smart and wise to keep your options open.",**

**"Do you have any questions for us?",**

**"It is always advised to ask some questions at the end of the interview. It shows that you are interested and you care about the job. Try not to be repetitive and ask something that has been discussed in the interview .Be sure to prepare yourself for these commonly asked questions but also be careful of any tricky questions that may come your way. Be cool and think before you answer and you’re sure to make a good impression with your prospective employer. Good luck!",**

**"Which is your favourite subject?",**

**"Your favourite subject or which subject you enjoy the most should be in sync with the job you are applying for. That means, keep two or three subjects in the pipeline which you can quote as your favourite subject and pick the one that goes best with the job profile as the right answer.",**

**"Tell me something about yourself that is not on your resume?",**

**"Here the interviewer is looking to know more about your personality and possibly evaluating culture fit, they want to see you as you are beyond your resume. Apart from your marks and internships and work experiences, what are the things that make you who you are and what that tells about what kind of an employee you will prove to be. \n In such a case, talk about one of your personal qualities that you might not have included in the resume, but that makes you a better candidate. If the job profile includes client interaction then you can talk about your exceptional oratory skills or you can highlight your problem solving skills if that is what the requirement is about. Similarly, you can focus on your work ethics to enrich your answer and impress the interviewer while moving beyond your resume. Quoting examples from your personal life can be a good practice while answering such questions.",**

**"Have you taken part in extra-curricular activities?",**

**"Do not talk about your hobbies here. There’s a difference between hobbies and extra-curricular activities. While hobbies are activities you do in your spare time, extra-curricular activities are activities you do at school/college-level where you have a certificate to back it up. It could be voluntary work like organizing an event or participating in sports, theatre and other competitions.",**

**"What’s your greatest achievement?",**

**"This need not always be academic achievement. It could be from your social life in the past too. Think and answer it.",**

**"Are you an introvert or extrovert?",**

**"Saying that you’re an introvert or an extrovert can classify you as an extreme, which is bad for you. When at work, your behavior needs to be as per the requirement. You can say that you behave as per the requirement and can be both.",**

**"Do you have plans for further studies?",**

**"This could be a tricky question to answer. If you answer with a yes, the company might feel that your further studies and work life may not complement each other. They might also feel that you would be working with them for a limited time. Be diplomatic here and say that you haven’t thought about it yet and will probably consider it after gaining some work experience in the industry and gaining some insights.",**

**"Where do you see yourself 5 years from now?",**

**"Learn about the industry you’re in and do an analysis. Talk to experienced professionals from the industry and understand it’s future as well as the growth pattern and how much time it’s going to take to get promoted from one position to another and whether you have the skills required. You need to be brutally honest with yourself here. If you don’t already have the skills, are you willing to develop them and how?",**

**"What is success for you?",**

**"Success is a subjective thing and can mean different things to different people. Don’t give them a cliched answer to this question. There’s not definitive answer to what success it, so tell them what it means to you.",**

**"Why does this role interest you? why have you applied?",**

**"This is asked to assess your interest levels in the role you’ve applied for. The interviewee wants to know if you’re really interested or have applied because you were jobless. Try and relate the job requirements with your candidature and explain to the interviewee why you’re suitable for the post.",**

**"What skills do you want to develop to success in the role?",**

**"Mention and talk about the roles necessary for the role you have applied for. This will show the interviewee that you’re ready to learn and also that you’re aware of your shortcomings. Tell them that, as a fresher, you have the ability to work hard but require a formal training to enhance the skills you have for professional environment.",**

**"Are you willing to relocate in India?",**

**"Answer this honestly and tell them if you have any plans to relocate in India and explain why.",**

**"Do you mind working in shifts?",**

**"You don’t have to agree just for the sake of it. Answer honestly and tell them if whether you’re ready to work in shifts or not. Mention if you have any inhibitions about working in shifts.",**

**"Why do you want to join our organization?",**

**"This is asked to check whether you have done your homework on the organization. Make sure you have read thoroughly about the organization through their website and social media and have gathered all their details before the interview. Tell them why you think their organization is great and how it is going to help you grow in your career.",**

**"How long will you work for us?",**

**"Every company knows that every professional working in their company will aspire a change as some point in their career. You could however, be diplomatic and tell them you haven’t thought about it or you could answer honestly too.",**

**" Are you speaking to other companies too?",**

**"Honestly answer this and say yes if you are",**

**"Which companies?",**

**"Don’t reveal the names of the companies. You can say that you respect the confidentiality of that company and you wouldn’t like to name them and say this politely.",**

**"Why should we hire you?",**

**"This is asked almost in every interview. You can talk about your greatest strengths here and link it with your job requirements. The company will hire you if they see that you can add value to the job. Tell them how your skills will help that position and about your ability to grasp things and learn quickly and your flexible attitude. These are some of the qualities interviewers look for in freshers.",**

**"How soon can you join us?",**

**"Think and answer this question and tell them frankly if you need time. Sometimes if you need to relocate for the job you will need some time for your commitments. A lot of freshers commit to an early joining date due to anxiety and face problems later. If you post-pone the joining it will give them a bad impression even before you join the company. Therefore give a practical answer.",**

**"Interview Do’s",**

**"1. Ask questions and be frank in your approach. 2.Be aware of the company’s profile and how well you can fulfil their requirement.3.Be Confident.Be comfortable while talking with your hand gestures.4.Be punctual and ensure that you reach at least 10 minutes early. 5.Bring a copy of all relevant documents. 6.Dress appropriately and look neat and clean. 7.Express yourself in simple words and clearly. 8.If you are being interviewed by a panel, ensure to make eye contact with the person who asks the question. 9.Be alert and listen to the questions and answer thoughtfully. 10.Present your skills in a positive light, even your weaknesses. 11.Make sure to fully understand the question and raise a question if you have a doubt about a certain statement. 12.In an interview, try to maintain the positive image that your CV has already created after the first round of short-listing. 13.Show enthusiasm for joining the company and the position.",**

**"Interview Don’ts",**

**"There are certain things to steer clear of. Go through them and ensure not to make these mistakes on your important day. 1. Don’t sit in a stiff posture. 2. Don’t answer questions with a simple “yes” or “no”. Make sure to explain your sentance on the statement. 3. Don’t dress casually or look untidy. 4. Don’t fidget while sitting. 5. Don’t interrupt the interviewer before they have finished asking a question. 6. Don’t lie when it comes to internship experiences and roles in the college. They can always cross-check. 7. Don’t make derogatory remarks about anyone, including your professors, supervisors, and fellow students. 8. Avoid asking too many questions about salary, holidays or bonuses. You may discuss these at length after an offer is made. 9. Don’t wear strong perfumes or fragrances.",**

**"What is your ambition?",**

**"Avoid generic statements like ‘I want to be a good manager’ or ‘I want to be a CEO in five years’. One can go for short-term and long-term ambitions. Try to make the answer explanatory and detailed so that the interviewer knows what plan one has charted out to achieve the bigger goal."**

**]**

**trainer.train(ques)**

**#----------function-----------**

**def reply():**

**if(entry.get()==""):**

**msg="Please enter your question"**

**l1.config(text=msg,fg="red")**

**else:**

**question=entry.get()**

**question=question.capitalize()**

**answer=bot.get\_response(question)**

**textarea.insert(END,"\n You: "+question+"\n")**

**textarea.insert(END,"Bot: "+str(answer)+"\n")**

**entry.delete(0,END)**

**msg=""**

**l1.config(text=msg,fg="red")**

**def clicker(event):**

**send.invoke()**

**def bck():**

**root.destroy()**

**import welcome**

**#--------logo-----------**

**top\_frame=Frame(root,width=500,bg="#000066")**

**top\_frame.pack()**

**img1=Image.open("images/images.png")**

**photo1=ImageTk.PhotoImage(img1)**

**back=Button(top\_frame,image=photo1,bg="#000066",command=bck)**

**back.pack(anchor=NW,side=LEFT)**

**img=Image.open("images/robot-chatter-bot-say-hi-over-circuit-background-vector-19840344.jpg")**

**photo=ImageTk.PhotoImage(img)**

**b\_img=Label(top\_frame,image=photo,bg="#000066",width=480,height=100)**

**b\_img.pack()**

**#----------main frame----------**

**main\_frame=Frame(root,bd=4,bg="Black",width=500)**

**main\_frame.pack()**

**scrollbar=Scrollbar(main\_frame)**

**scrollbar.pack(side=RIGHT,fill=Y)**

**textarea=Text(main\_frame,width=60,height=16,bd=4,font="century 14 italic",yscrollcommand=scrollbar.set,wrap="word")**

**textarea.pack()**

**scrollbar.config(command=textarea.yview)**

**#---------frame----------------**

**f1=Frame(root,bg="#000066")**

**f1.place(x=0,y=489,height=110,width=500)**

**entry=Entry(root,font="arial 15",bd=3)**

**entry.place(x=20,y=525,height=38,width=325)**

**send=Button(root,text="SEND",font=("Calibri 14 bold"),width=10,bg="#f9eae7",fg="#000066",command=reply)**

**root.bind("<Return>",clicker)**

**send.place(x=360,y=525)**

**msg=""**

**l1=Label(root,text=msg,font="arial 12 bold",bg="#000066")**

**l1.place(x=30,y=565)**

**root.mainloop()**

****

**from tkinter import \***

**from PIL import Image, ImageTk**

**from chatterbot import ChatBot**

**from chatterbot.trainers import ListTrainer**

**root=Tk()**

**root.title('Chatbot')**

**root.geometry('500x600+400+50')**

**root.maxsize(500,600)**

**root.minsize(500,600)**

**bot=ChatBot("Bot")**

**trainer=ListTrainer(bot)**

**chat=[**

**"Hi",**

**"Hello my friend",**

**"Hello",**

**"Hey",**

**"Hey",**

**"Hello my friend",**

**"Who are you?",**

**"I'm chatterBot-Your placement guide",**

**"How are you",**

**"I'm fine. And I Hope the same to you too!",**

**"How can we prepare for placement",**

**"You need to prepare for Quantitative Aptitude, Reasoning, Verbal & Interview.",**

**"How to prepare for placement",**

**"You need to prepare for Quantitative Aptitude, Reasoning, Verbal & Interview.",**

**"What are the syllabus of Quantitative aptitude?",**

**"For aptitude you need to study maths which includes topics such as:- Percentage, Profit & Loss, Time & Work, Speed Time & Distance, Permutation & Combination, Simple Interest, Compound Interest, Mensuration, Probability, Average, Ratio & Proportion, Mixture & Alligation, Simple Equation, Problem on Numbbers, Indices and Surds",**

**"Syllabus of Quantitative aptitude?",**

**"For aptitude you need to study maths which includes topics such as:- Percentage, Profit & Loss, Time & Work, Speed Time & Distance, Permutation & Combination, Simple Interest, Compound Interest, Mensuration, Probability, Average, Ratio & Proportion, Mixture & Alligation, Simple Equation, Problem on Numbbers, Indices and Surds",**

**"Syllabus of quant",**

**"For quant you need to study maths which includes topics such as:- Percentage, Profit & Loss, Time & Work, Speed Time & Distance, Permutation & Combination, Simple Interest, Compound Interest, Mensuration, Probability, Average, Ratio & Proportion, Mixture & Alligation, Simple Equation, Problem on Numbbers, Indices and Surds",**

**"Correct time start preparation",**

**"There is no fixed time to start. It may vary from person to person but as a BCA or Bsc-IT student where placements starts from 5th semester one is adviced to be prepared till that time.",**

**"Syllabus for verbal test",**

**"Passage/Sentence Rearrangement, Error Detection & Correction, Fill in the Blanks(icludes all parts of speech), Comprehension Passages",**

**"What is the syllabus for verbal test?",**

**"Passage/Sentence Rearrangement, Error Detection & Correction, Fill in the Blanks(icludes all parts of speech), Comprehension Passages",**

**"What is the syllabus for reasoning?",**

**"Series : Missing Numbers, Odd One Out, Data Sufficiency, Assumptions & Conclusions, Courses of Action, Puzzles, Syllogism, Cubes, Coding-Decoding",**

**"Syllabus for reasoning?",**

**"Series : Missing Numbers, Odd One Out, Data Sufficiency, Assumptions & Conclusions, Courses of Action, Puzzles, Syllogism, Cubes, Coding-Decoding",**

**"How to start preparation?",**

**"One must focus on his quatitative aptitude, verbal and resoning for their first round of placement and then on their technical skills and communication skills for their interview.",**

**"How to prepare for interview?",**

**"One must have command in atlest one programming language and have knowledge of their course whatever they have studied throughout their academics and it is more beneficial to have a project included.",**

**"Which programming languages should I learn?",**

**"You can learn as many programming languages you want but make sure to master at least any one programming language of your choice. Recruiters only see whether you are able to code a given problem in any language of your choice (C, C++, Java etc.). We recommend you to master any one structured programming language like C and any one of the Object Oriented Programming language like C++, Java etc.",**

**"How many programming languages should I learn?",**

**"You can learn as many programming languages you want but make sure to master at least any one programming language of your choice. Recruiters only see whether you are able to code a given problem in any language of your choice (C, C++, Java etc.). We recommend you to master any one structured programming language like C and any one of the Object Oriented Programming language like C++, Java etc.",**

**"Are projects important for placement?",**

**"Projects lay a very vital role in your interview. It puts up a good impression on interviewers and also let them have an idea about your skills and your knowledge.",**

**"How to contact a company about placement?",**

**"You need to work out the best way to apply when you want to contact a placement company. You need to find whether the company has a career portal for job seekers on their website or not. You also require to figure out if they prefer meeting face-to-face or taking a telephonic interview.",**

**"When should we start preparing for the placement test?",**

**"There is no fixed time to start. It may vary from person to person but as a BCA or Bsc-IT student where placements starts from 5th semester one is adviced to be prepared till that time.",**

**"How can I start my placement preparation?",**

**"One must focus on his quatitative aptitude, verbal and resoning for their first round of placement and then on their technical skills and communication skills for their interview.",**

**"What do we need to prepare for our interview?",**

**"One must have command in atlest one programming language and have knowledge of their course whatever they have studied throughout their academics and it is more beneficial to have a project included.",**

**"What programming languages should I learn?",**

**"You can learn as many programming languages you want but make sure to master at least any one programming language of your choice. Recruiters only see whether you are able to code a given problem in any language of your choice (C, C++, Java etc.). We recommend you to master any one structured programming language like C and any one of the Object Oriented Programming language like C++, Java etc.",**

**"How much projects are important to be added in resume?",**

**"Projects add a pinch of salt to your resume. Try to add 2-3 good projects with somehow description of 1-2 lines. Also, be prepared to deliver detail description of your project and also some of the corner scenarios your project should handle. While applying off-campus, try to include projects according to the technology on which company has vacancies because many companies have a resume filtering option. That is when you apply for a vacancy on their website your resume passes through an initial keyword matching process before it reaches to the hiring team.",**

**"How do I contact a company about placement?",**

**"You need to work out the best way to apply when you want to contact a placement company. You need to find whether the company has a career portal for job seekers on their website or not. You also require to figure out if they prefer meeting face-to-face or taking a telephonic interview.",**

**"Does writing clean code increases the chance of getting selected?",**

**"Writing clean code surely creates a good impression of you in front of the recruiter. Recruiters always prefer someone who can write code which is easily understandable and should also be efficient at the same time.",**

**'',**

**''**

**]**

**trainer.train(chat)**

**#----------function-----------**

**def reply():**

**if(entry.get()==""):**

**msg="Please enter your question"**

**l1.config(text=msg,fg="red")**

**else:**

**question=entry.get()**

**question=question.capitalize()**

**answer=bot.get\_response(question)**

**textarea.insert(END,"\n You: "+question+"\n")**

**textarea.insert(END,"Bot: "+str(answer)+"\n")**

**entry.delete(0,END)**

**msg=""**

**l1.config(text=msg,fg="red")**

**def clicker(event):**

**send.invoke()**

**def bck():**

**root.destroy()**

**import welcome**

**bot=ChatBot("Bot")**

**trainer=ListTrainer(bot)**

**trainer.train(chat)**

**#--------logo-----------**

**top\_frame=Frame(root,width=500,bg="#000066")**

**top\_frame.pack()**

**img1=Image.open("images/images.png")**

**photo1=ImageTk.PhotoImage(img1)**

**back=Button(top\_frame,image=photo1,bg="#000066",command=bck)**

**back.pack(anchor=NW,side=LEFT)**

**img=Image.open("images/robot-chatter-bot-say-hi-over-circuit-background-vector-19840344.jpg")**

**photo=ImageTk.PhotoImage(img)**

**b\_img=Label(top\_frame,image=photo,bg="#000066",width=480,height=100)**

**b\_img.pack()**

**#----------main frame----------**

**main\_frame=Frame(root,bd=4,bg="Black",width=500)**

**main\_frame.pack()**

**scrollbar=Scrollbar(main\_frame)**

**scrollbar.pack(side=RIGHT,fill=Y)**

**textarea=Text(main\_frame,width=60,height=16,bd=4,font="century 14 italic",yscrollcommand=scrollbar.set,wrap="word")**

**textarea.pack()**

**scrollbar.config(command=textarea.yview)**

**#---------frame----------------**

**f1=Frame(root,bg="#000066")**

**f1.place(x=0,y=489,height=110,width=500)**

**entry=Entry(root,font="arial 15",bd=3)**

**entry.place(x=20,y=525,height=38,width=325)**

**send=Button(root,text="SEND",font=("Calibri 14 bold"),width=10,bg="#f9eae7",fg="#000066",command=reply)**

**root.bind("<Return>",clicker)**

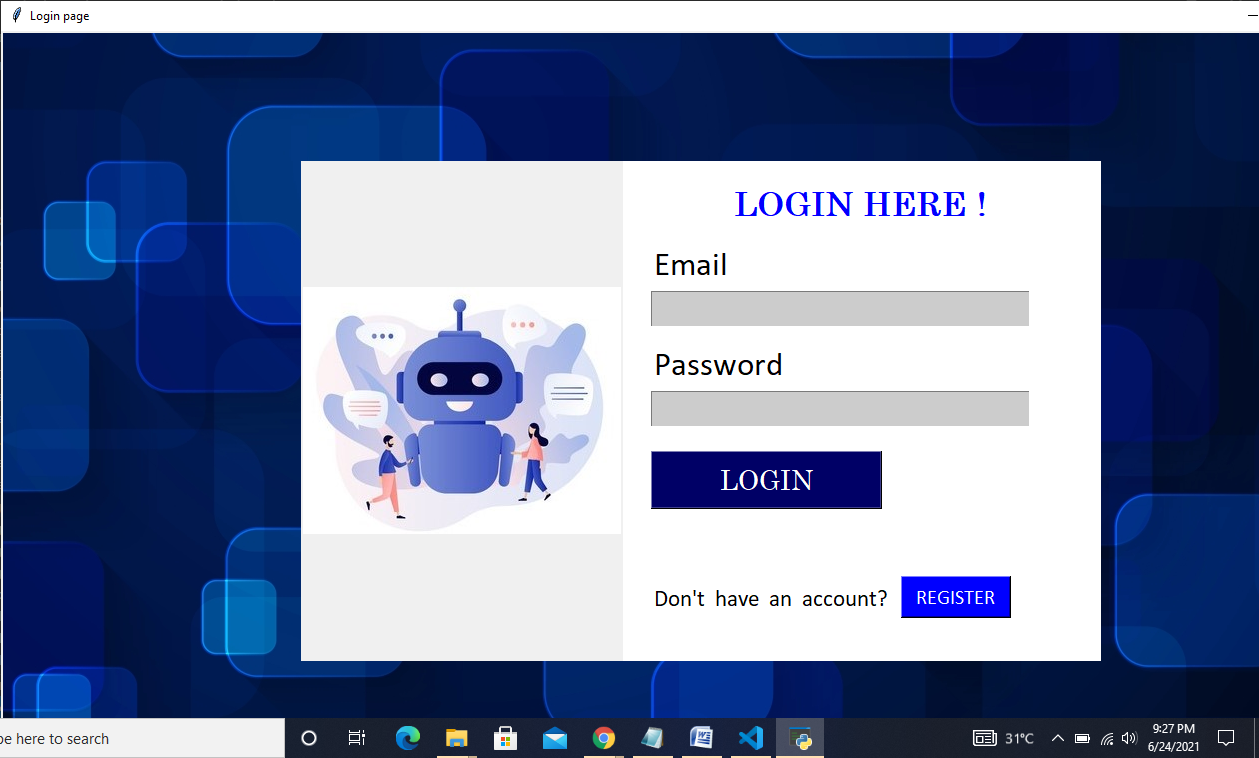
**send.place(x=360,y=525)**

**msg=""**

**l1=Label(root,text=msg,font="arial 12 bold",bg="#000066")**

**l1.place(x=30,y=565)**

**root.mainloop()**

****

**from tkinter import \***

**import pymysql**

**from PIL import Image, ImageTk**

**import tkinter.messagebox as MessageBox**

**root=Tk()**

**root.title('Login page')**

**root.geometry('1366x768+100+10')**

**root.maxsize(1366,768)**

**root.minsize(1366,768)**

**f=("Calibri 25")**

**#--------function-------**

**def register():**

**root.destroy()**

**import register**

**def login():**

**if(e1.get()=="" or e2.get()==""):**

**MessageBox.showerror("ERROR","Please fill all the Fields")**

**else:**

**mycon = pymysql.connect(host="localhost",user="root",password="",db="regst")**

**mycur = mycon.cursor()**

**mycur.execute("select \* from student where email=%s and pas=%s",(e1.get(),e2.get()))**

**row=mycur.fetchone()**

**if(row==None):**

**MessageBox.showerror("ERROR","Invalid Email or Password")**

**else:**

**MessageBox.showinfo("SUCCESS","Successfully Logged In")**

**root.destroy()**

**import welcome**

**mycon.close()**

**#--------BACKGROUND DETAILS---------------**

**img=Image.open("images/wallpapersden.com\_blue-digital-art-squares\_1366x768.jpg")**

**photo=ImageTk.PhotoImage(img)**

**b\_img=Label(root,image=photo)**

**b\_img.place(x=0,y=0)**

**#---------frame----------------**

**f1=Frame(root,bg="#ffffff")**

**f1.place(x=300,y=130,height=500,width=800)**

**#-------LOGO----------**

**logo=Image.open("images/image.jpg")**

**photo1=ImageTk.PhotoImage(logo)**

**icon=Label(root,image=photo1)**

**icon.place(x=300,y=130,height=500)**

**#----------login details-------**

**name=Label(root,text="LOGIN HERE !",font="century 25 bold",bg="#ffffff",fg="blue")**

**name.place(x=730,y=150)**

**uname=Label(root,text="Email",font=f,bg="#ffffff")**

**uname.place(x=650,y=210)**

**e1=Entry(root,font="arial 20",width=25,bg="#cccccc")**

**e1.place(x=650,y=260)**

**upass=Label(root,text="Password",font=f,bg="#ffffff")**

**upass.place(x=650,y=310)**

**e2=Entry(root,font="arial 20",width=25,bg="#cccccc")**

**e2.place(x=650,y=360)**

**login=Button(root,text="LOGIN",font=("century 20"),width=13,bg="#000066",fg="white",command=login)**

**login.place(x=650,y=420)**

**l1=Label(root,text="Don't have an account?",fg="black",font="Calibri 17",bg="#ffffff")**

**l1.place(x=650,y=550)**

**regs=Button(root,text="REGISTER",font=("Calibri 15"),width=10,bg="#0000ff",fg="white",command=register)**

**regs.place(x=900,y=545)**

**root.mainloop(****)**

TESTING

In the ideal circumstances a software engineer design an computer , a system, or a product with “Testability” in mind. This enables the individuals charged with testing to design effective test cases more easily. But what is “Testability”? James Bach describes testability in the following manner:-

Software testability is simply how easily a computer program can be tested. Since testing is so profoundly difficult, it pays to know what can be done to it.

Sometimes programmers are willing to do things that will help the testing process and a checklist of possible design points, features etc. can be useful in negotiation with them. There are certainly metrices that could be used to measures testability in most of its aspects. Sometimes, testability is used to measure testability in most of aspects. Sometimes, testability is used to mean how adequately a particular set of tests will cover the product. Its also used by the military to mean how easily a tool can be checked and repaired in the field. These two meanings are not the same as software testability. The checklist that follows provides a set of characteristics that lead to testable software.

1. **BLACK BOX TESTING**

Black box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of inputs conditions that fully exercise all functional requirements for the program. Black box testing is not an alternative to white box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors then white box methods.

Black box testing attempts to find errors in the following categories:

1. Incorrect or missing function
2. Interface errors
3. Errors in data structures or external database access
4. Behavior or performance errors, and
5. Initialization or termination errors

When computer software is considered, black box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information is maintained. A black box test examines some fundamental aspect of a system with regard for the internal logical structure of the software.

We have tested each and every function in the project and come to the conclusion that they all have executed correctly. At the time when the project was in progress a number if functions where deleted and then added due to which errors where generated. This errors where removed time to time in the testing phase where I had to go in the depth study of the project. Its each and every parts where executed at least 100 times to see whether the working was anticipating the expected results or not.

**Tests where Designed to answer the following questions?**

* How is functional validity tested?
* How is system behavior and performance tested?
* What classes of inputs will make good test cases?
* Is the system particularly sensitive to certain input values?
* How are the boundaries of a data volume can the system tolerate?
* What effects will specific combinations of data have on system operations?

By applying Black Box technique, we derive a set of answers for the questions:

* The functional validity is checked by the execution of the functions one by one and fulfilling the requirements of the functions as per the demand of the software. The functional validity is checked by using suitable technique which lies on the code optimization, efficiency in executions.
* System behavior is tested by certain tools and performance tests lies on the dumping of large bulk of data into the application software database. The application database consists of data’s with their appropriate width. The validation of the data’s linking of one data’s with the other’s everything are tested before inputting them into the application software database.
* Interactive classes of input are generally better for the good test classes .
* The system is sensitive for the system data because the whole application software is a time domain software. Any error on the time parameter may crash the whole application. Time is taken as an important parameter for all the calculations. This is why this application software is sensitive to system time.
* Boundary of data classs are isolated by giving the range of the arrays. Its totally a static memory allocated. Hence, any error on the static array may crash the whole applications.
* The specific can tolerate 50MB of the memory of the data without any problem.
* The specific combinations of the data of center id and time. The application program has no control over time and id. Both are generated automatically.

### ALPHA AND BETA TESTING

It is virtually impossible for a software developer to foresee how the user will really use a program. Instructions for use may be misinterpreted. strange combinations of data may be regularly used; output that seemed clear to the tester may be unintelligible to the user in the filed.

When software is built for one person, a series of acceptance tests are conducted to enable the user to validate all requirements. Conducted by the end-user rather than software engineers, an acceptance test can range from an informal “test-drive” to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of time thereby uncovering errors that might degrade the system overtime.

If software is developed as a product to be used by many peoples, it is impractical to perform acceptance tests with one another. Most software builders use a process called alpha and beta testing to uncover error that only the end user seems able to find. The alpha test is conducted at the developer site by a patients. The software is used in a natural setting with a developer “looking over the shoulder” of the users and recording errors and usage problems. Alpha tests are conducted in an controlled environment. The beta test is conducted at one or more patients site by the end user of software. Unlike alpha testing, the developer is generally not present. Therefore the beta test is a “live” application of the software in an environment that cannot encountered in beta testing a reports these to the developer at regular interval. As a result of problems reported during beta tests. Software engineers make modifications and then prepare for the release of the software product to the entire employee base. I this stage alpha and beta testing where also done to determine the working of the software i.e.to check the validation of the software, whether the results which it is generating is valid or not.

**Alpha Testing**

When user software is build for one employee a serial of dependent tests ate conducted to enable the users to validate the requirement conducted by an user rather than software engineers. And acceptance test can from an informal to applied systematically. This test is conducted at developer site by a user the software is used in a natural setting with the developers as we can say-“looking over the soldier” of the user and recording errors alpha test are conducted then a control.

**Beta Testing**

The beta testing is conducted at one or more users’ side by the user of the software. Unlike alpha testing the developer generally is not present therefore Beta testing is live application of the software that controlled by the developer at the result the software engineerings may modify and then prepare for release of the software.

SECURITY MECHANISMS

Any user, who is authorized by the administrator to operate the software, receives a unique user ID and a password with the help of which he can log into the system. To login the user goes through the system menu. He is then presented with the login box where he has to enter the username and password to login.

In this software, the login Id and password which is created for the login for the operation is strictly maintained in the database by the automatic automation. This way, we can easily find out at what time which user login to handle the process which is done by the specified work. If there will be any problem by the handling of the money and the transaction, we can check is there any user is doing anything wrong or not. The database maintained when the user login for the transaction or for any other purpose, it shows time and date in which period he/she was working at that time. Same user can log on many times after taking logout from the software.

**In this security features user may belongs to any of the two categories:-**

1. Authorized User 2. Administration

First when the software is installed in the organization the software developer provide the user Id and password for the different work for different people. Like we provide the user Id and password to staff of the hospital to process some work but not all, same way the administrator get all the features for the processes. He can do anything as he wants to do. Administrator provides the work for the staff of the hospital as per their requirement.

The user login in may access only certain features within the software. A user will only be allowed to view the records and opening different types of features and maintaining the database, providing the requirement of the user as they demanded. Every user get the unique and password. Every new user will be assigned a unique name along with a password by the administrator. The list of all users is maintained by the administrator of the software.

Once the administrator logs into the system, he has access to all the features within the system other than all essential details, the administrator has a separate responsibility which includes managing users and their details. So if a new user has to be added, this can be done by the administrator. Therefore we have seen that a person who is logging in as a user who does not exists in the user logging in is not a valid user that is the user is not available in the table.

When sometimes the user logging in gives a incorrect password then message box is displayed which informs the user that the password enterd is not correct and may try again. Any user, who is authorized by the administrator to operate the software, receives a unique user Id and a password with the help of which he can log into the system. To login the user goes through the SYSTEM menu. He’s then presented with the login box where he has to enter the username and password to login.

Cost Estimation of the Project

* Predicting the resources required for a software development process

**Topics covered**

* Feasibility Analysis
* Productivity measures
* Estimation techniques
* Algorithmic cost modelling
* Project duration

**Feasibility Analysis**

Feasibility: – the measure of how beneficial or practical an information system will be to an organization.

Feasibility analysis: – the process by which feasibility is measured.

**Three Tests for Feasibility**

**Technical feasibility:** – a measure of the practicality of a technical solution and the availability of technical resources and expertise.

**Economic feasibility:** - a measure of the cost-effectiveness of a project or solution.

**Operational feasibility** – a measure of how well a solution will work or be accepted in an organization.

**Economic Feasibility = Cost-Benefit Analysis**

**Costs:**

* **Development costs** are one time costs that will not recur after the project has been completed.
* **Operating costs** are costs that tend to recur throughout the lifetime of the system. Such costs can be classified as:
* **Fixed costs** — occur at regular intervals but at relatively fixed rates.
* **Variable costs** — occur in proportion to some usage factor**.**

**Benefits:**

* **Tangible benefits** are those that can be easily quantified.
* **Intangible benefits** are those benefits believed to be difficult or impossible to quantify.

**Three Popular Techniques to Assess Economic Feasibility**

* Payback Analysis
* Return On Investment
* Net Present Value

The **Time Value of Money** is a concept that should be applied to each technique. The time value of money recognizes that a dollar today is worth more than a dollar one year from now.

**Software cost components**

* Hardware and software costs
* Travel and training costs
* Personnel costs (the dominant factor in most projects)
* salaries of engineers involved in the project
* Social and insurance costs
* Must also take project overhead into account
* costs of building, heating, lighting
* costs of networking and communications
* Costs of shared facilities (e.g. library, staff restaurant, etc.)

**Fundamental estimation questions**

* How much effort is required to complete an activity?
* How much calendar time is needed to complete an activity?
* What is the total cost of an activity?
* Project estimation and scheduling are interleaved management activities

**Costing and pricing**

* Estimates are made to discover the cost, to the developer, of producing a software system
* There is not a simple relationship between the development cost and the price charged
* Broader organisational, economic, political and business considerations influence the price charged

**Productivity measures**

* Size related measures based on some output from the software process. This may be lines of delivered source code, object code instructions, etc.
* Function-related measures based on an estimate of the functionality of the delivered software. Function-points are the best known of this type of measure

**Measurement problems**

* Estimating the size of the measure
* Estimating the total number of programmer months which have elapsed
* Estimating contractor productivity (e.g. documentation team) and incorporating this   
  estimate in overall estimate

**Lines of code**

* What is a line of code?
* Productivity measures will vary from language to language – consider difference between lines of code in assembler versus Java
* Relationship to functionality must be based on past efforts in the same language

**Productivity estimates**

|  |  |
| --- | --- |
| System Category | LOC/person-month |
| Real-time embedded systems | 40-160 |
| Systems programs | 150-400 |
| Commercial applications | 200-800 |

**Function points**

* Based on a combination of program characteristics
* external inputs and outputs
* user interactions
* external interfaces
* files used by the system
* A weight is associated with each of these
* The function point count is computed by multiplying each raw count by the weight and summing all values

**Function points**

* Function point count modified by complexity of the project
* FPs can be used to estimate LOC depending on the average number of LOC per FP for a given language
* LOC = AVC \* number of function points
* AVC is a language-dependent factor varying from 200-300 for assemble language to 2-40 for a 4GL
* FPs are very subjective. They depend on the estimator.
* Automatic function-point counting is impossible

**4GL Object points**

* Object points are an alternative function-related measure to function points when 4Gls or similar languages are used for development
* Object points are NOT the same as object classes
* The number of object points in a program is a weighted estimate of

a)The number of separate screens that are displayed

b)The number of reports that are produced by the system

c)The number of 3GL modules that must be developed to supplement the 4GL code

**Object Point Estimation**

* Object points are easier to estimate from a specification than function points as they are simply concerned with screens, reports and 3GL modules
* They can therefore be estimated at an early point in the development process. At this stage, it is very difficult to estimate the number of lines of code in a system

**Quality and productivity**

* All metrics based on volume/unit time are flawed because they do not take quality into   
  account
* Productivity may generally be increased at the cost of quality
* It is not clear how productivity/quality metrics are related
* If change is constant then an approach based on counting lines of code is not as meaningful

**Estimation techniques**

* There is no simple way to make an accurate estimate of the effort required to develop a software system
* Initial estimates are based on inadequate information in a user requirements definition
* The software may run on unfamiliar computers or use new technology
* The skills of people working on the project may be unknown
* Project cost estimates may be self-fulfilling
* The estimate defines the budget and the product is adjusted to meet the budget
* Expert judgement
* Estimation by analogy
* Parkinson's Law
* Pricing to win
* Algorithmic cost modelling

**Expert judgement**

* One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached.
* Advantages: Relatively cheap estimation method. Can be accurate if experts have direct   
  experience of similar systems
* Disadvantages: Very inaccurate if there are no experts!

**Estimation by analogy**

* The cost of a project is computed by comparing the project to a similar project in the same   
  application domain
* Advantages: Accurate if project data available
* Disadvantages: Impossible if no comparable project has been tackled. Needs systematically   
  maintained cost database

**Parkinson's Law**

* The project costs whatever resources are available (typically used within an organization)
* Advantages: No overspend
* Disadvantages: System is usually left unfinished

**Pricing to win**

* The project costs whatever the organisation has to spend on it
* Advantages: You get the contract
* Disadvantages: Costs do not accurately reflect the work required. Either: (1) the management does not get the desired system or (2) the management overpays.
* This approach may seem unethical and unbusiness-like
* However, when detailed information is lacking it may be the only appropriate strategy
* The most ethical approach:
  + The project cost is agreed on the basis of an outline proposal and the development is constrained by that cost
  + A detailed specification may be negotiated or an evolutionary approach used for system development

**Top-down and bottom-up estimation**

* Any of these approaches may be used top-down or bottom-up
* Top-down
* Start at the system level and assess the overall system functionality and how this is delivered through sub-systems
* Bottom-up
* Start at the component level and estimate the effort required for each component. Add these efforts to reach a final estimate
* Usable without knowledge of the system architecture and the components that might be part of the system
* Takes into account costs such as integration, configuration management and documentation
* Can underestimate the cost of solving difficult low-level technical problems

**Bottom-up estimation**

* Usable when the architecture of the system is known and components identified
* Accurate method if the system has been designed in detail
* May underestimate costs of system level activities such as integration and documentation

**Estimation methods**

* Each method has strengths and weaknesses
* Estimation should be based on several methods
* If these do not return approximately the same result and the differences cannot be reconciled, there is insufficient information available
* Some action should be taken to find out more in order to make more accurate estimates

**Experience-based estimates**

* Estimating is primarily experience-based
* However, new methods and technologies may make estimating based on experience inaccurate
* Object-oriented rather than function-oriented development
* Client-server systems rather than mainframe systems
* Many off the shelf components
* Component-based software engineering
* Use of new CASE tools and program generators

**Algorithmic cost modelling**

* A formulaic approach based on historical cost information and which is generally based on the size of the software
* Cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers

**Algorithmic cost modelling**

* Effort in PM = A ´ SizeB ´ M
* **A** is depends in on the type of software that is being developed (simple, moderate, embedded) [will vary 2.4-3.5]
* **Size** is an estimate of the code size or other functional assessmemt [thousands of lines of code, ie. 5,400 LOC🡪 5.4]
* **B** reflects the disproportionate effort for large projects over small projects [typically 1.0-1.5]
* **M** is a multiplier reflecting a combination of product, process and people attributes (e.g. desired reliability, reuse required, personnel capability and expereince, support facilities) [will vary up from 1.0]

**Estimation accuracy**

* The size of a software system can only be known accurately when it is finished
* Several factors influence the final size
  + Use of “off the shelf” components
  + Programming language
  + Distribution of system
* As the development process progresses then the size estimate becomes more accurate

**Estimate uncertainty**

x

2

x

4

x

0

.

5

x

0

.

2

5

x

F

e

a

s

i

b

i

l

i

t

y

R

e

q

u

i

r

e

m

e

n

t

s

D

e

s

i

g

n

C

o

d

e

D

e

l

i

v

e

r

y

Estimate uncertainty:

As the project progresses

the probablilty of a difference in

actual to estimate decreases

Actual

Person-

months

x = estimated person-months

**COCOMO**

* Constructive Cost Model
* An empirical model based on project experience
* Well-documented, ‘independent’ model which is not tied to a specific software vendor
* Long history from initial version published in 1981 (COCOMO-81) through various instantiations to COCOMO 2
* COCOMO 2 takes into account different approaches to software development, reuse, etc.
* Can be used as a sanity check

**Estimate Cost and Duration Very Early in Project**

1. Use the function point method to estimate lines of code

2. Use Boehm’s formulas to estimate labor required

3. Use the labor estimate and Boehm’s formula to estimate duration

**Basic COCOMO Formulae (Boehm)**

**Effort in Person-months = *a × KLOC b***

**Duration in Months = *c × Effort d***

Where c = labour estimate, d = complexity of project type

These values are selected from a table such as the one below.

**Software Project a b c d**

**Organic 2.4 1.05 2.5 0.38**

**Semidetached 3.0 1.12 2.5 0.35**

**Embedded 3.6 1.20 2.5 0.32**

PERT CHART

**PERT CHART** stand for Program Evaluation Review Techniques, unlike bar chart PERT can be both a cost and a time management system. PERT is organized by events and activity or task. One advantage of the PERT chart is that it is a schedule device that also shows graphically which task must be completed before others are begun.   
  
**Advantage of the pert chart is as follows:**

* It force the manager to the plan
* It shows the interrelationship among the tasks in the project and in particularly it identified the critical path of the project, thus helping to focus on it.
* It exposes all the particularly all-possible parallelism in the activities and thus help in allocating the resources.
* It allows scheduling and simulation of alternative schedules.
* It enables the manager to monitor and control the project.

**Program Evaluation Review Techniques (P.E.R.T) Chart**

Design

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Project Initiation

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Preliminary

Investigation

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Problem

Analysis

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

R.A

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Decision

Analysis

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Computation

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

Implementation

Preliminary

Investigation

s.start s.finish

Actual actual

Start Finish

**Gantt chart**

A Gantt chart the perhaps the simplest forms of formal project management. The Gantt chart is used almost exclusively for scheduling purpose and therefore controls only the time dimension of projects.

Gantt chart are a project control technique that can be used for several purposes, including, budgeting, and resource is planning. A Gantt chart is a bar chart, with each bar representing an activity. The length of each bar is proportional to the length of time planned for activity. A grant chart helps in scheduling the activities of a project, but it does not help in identifying them.

**Gantt chart**

**1ST AUG**

**Start**

**2ndAug-15thSep**

**Requirement analysis**

**16STSep-15THOct**

**Design**

**Approval of design**

**16thOct-30thNovSEPT**

**1STDec-15THJAN OCT**

**Coding**

**16thJan-25thJan**

**Testing**

**6thFeb-15thFeb**

**26thJan-5thFeb**

**Beta testing**

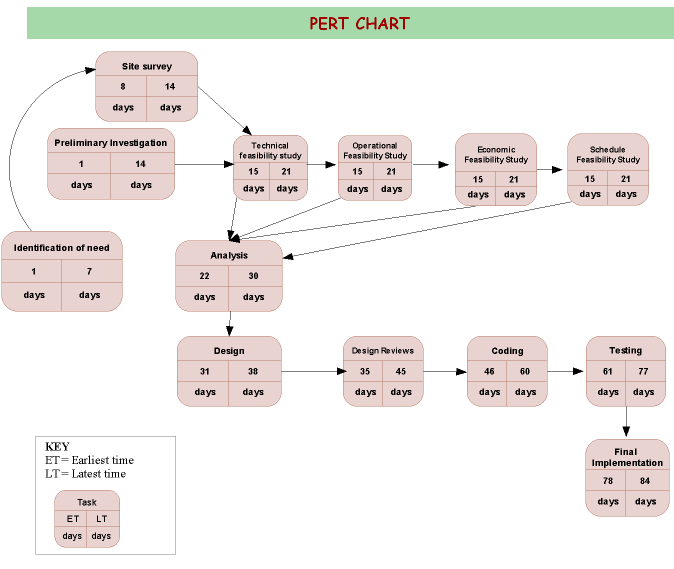
**Alpha Testing**

**16thFeb-15thMar**

**Documentation**

**Implementation & Evaluation**

**PERT CHART**

Pert chart is a graphic representation of a project’s schedule, showing the sequence of tasks, which tasks can be performed simultaneously, and the critical path of tasks that must be completed on time in order for the project to meet its completion deadline. The chart can be constructed with a variety of attributes, such as earliest and latest start dates for each task, earliest and latest finish dates for each task, and slack time between tasks. A PERT chart can document an entire project or a key phase of a project. The chart allows a team to avoid unrealistic timetables and schedule expectations, to help identify and shorten tasks that are bottlenecks, and to focus attention on most critical tasks.

**Conclusion**

What we’ve illustrated here is just one among the many ways of **how to make a chatbot in Python.** You can also use NLTK, another resourceful Python library to create a Python chatbot. And although what you learned here is a very basic**chatbot in Python** having hardly any cognitive skills, it should be enough to help you understand the anatomy of chatbots.

Once you understand the design of a **chatbot using python** fully well, you can experiment with it using different tools and commands to make it even smarter.