



M.KUMARASAMY
COLLEGE OF ENGINEERING

NAAC Accredited Autonomous Institution

Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution

Thalavapalayam, Karur – 639 113.



A Minor Project Report

On

MKCE ChatBot

Submitted in partial fulfilment of requirements for the award of the

Degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Mrs. A. SELVI Assistant Professor/CSE

Submitted By

NAVEENA M (20BCS4068)

SHARMI K (20BCS4085)

SRINITHI B (20BCS4089)

YOGI N (20BCS4107)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

NOVEMBER 2021



M.KUMARASAMY
COLLEGE OF ENGINEERING

NAAC Accredited Autonomous Institution

Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution

Thalavapalayam, Karur – 639 113.



M. KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

KARUR – 639113

BONAFIDE CERTIFICATE

Certified that this minor project report “**MKCE ChatBot**” is the bonafide work of “**NAVEENA. M (20BCS4068) , SHARMI. K (20BCS4085) , SRINITHI. B (20BCS4089) , YOGI. N (20BCS4107) ”** who carried out the project work during the academic year (2021-2022) under my supervision.

Signature

Mrs.A.SELVI Assistant Professor/CSE

Department of Computer Science
and Engineering,
M. Kumarasamy College of Engineering,
Thalavapalayam, Karur -639113

Signature

Dr. S.THILAGAMANI M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Computer Science
and Engineering,
M. Kumarasamy College of
Engineering
Thalavapalayam, Karur -639113.






DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education




MISSION OF THE INSTITUTION

-  Produce smart technocrats with empirical knowledge who can surmount the global challenges
-  Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
-  Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

VISION OF THE DEPARTMENT

To achieve education and research excellence in Computer Science and Engineering

MISSION OF THE DEPARTMENT

-  To excel in academic through effective teaching learning techniques
-  To promote research in the area of computer science and engineering with the focus on innovation
-  To transform students into technically competent professionals with societal and ethical responsibilities

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Graduates will have successful career in software industries and R&D divisions through continuous learning.

PEO 2: Graduates will provide effective solutions for real world problems in the key domain of computer science and engineering and engage in lifelong learning.

PEO 3: Graduates will excel in their profession by being ethically and socially responsible.



PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.



M.KUMARASAMY
COLLEGE OF ENGINEERING

NAAC Accredited Autonomous Institution



Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution

Thalavapalayam, Karur – 639 113.



- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

-  **PSO1: Professional Skills:** Ability to apply the knowledge of computing techniques to design and develop computerized solutions for the problems.
-  **PSO2: Successful career:** Ability to utilize the computing skills and ethical values in creating a successful career.



ABSTRACT WITH PO AND PSO MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>This project is to develop a website based on forum where users can register with application and post questions or query related to college. Admin will maintain the user details and query details. Admin can view queries and update answers of user's query. It will also be useful for getting information about college. Users can satisfy their needs of getting information related to various questions and queries from a single source. To make a form that give information about various questions. The Admin maintain all the frequently asked questions in the database. Time to time they will be updating the questions and the related answers into the database. The primary goal of the MKCE Chatbot is to provide an environment that is both convenient and efficient for the students, parents and other unknown person through which the user can get clarified with their doubts and also for the administrators of a particular college to use in retrieving the information from database and also storing the information into database.</p>	<p>PO1(3) PO 2(3) PO 3(2) PO 4(2) PO 5(2) PO6(1) PO 7(3) PO 8(2) PO 9(3) PO 10(3) PO 11(2) PO 12(2)</p>	<p>PSO 1(3) PSO 2(2)</p>

Note: 1- Low, 2-Medium, 3- High

SUPERVISOR

HEAD OF THE DEPARTMENT

ABSTRACT

Main objective of this project is to develop a website based on forum where users can register with application and post questions or query related to college. Admin will maintain the user details and query details. Admin can view queries and update answers of user's query. It will also be useful for getting information about college. Users can satisfy their needs of getting information related to various questions and queries from a single source. To make a form that give information about various questions. The Admin maintain all the frequently asked questions in the database. Time to time they will be updating the questions and the related answers into the database. The primary goal of the MKCE Chatbot is to provide an environment that is both convenient and efficient for the students, parents and other unknown person through which the user can get clarified with their doubts and also for the administrators of a particular college to use in retrieving the information from database and also storing the information into database.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	Abstract	vii
	List of Figures	ix
	Acronyms/List of Abbreviations	x
1	Introduction	1
	1.1 Overview	1
	1.2 Domain Introduction	2
	1.3 Problem Statement	6
	1.4 Objective	7
2	Feasibility Study	8
3	Project Methodology	11
	3.1 Block diagram of MKCE ChatBot	11
	3.2 Module Description	13
4	Results and Discussion	15
5	Conclusion	24
	References	25

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	MKCE CHATBOT	1
2.1	Feasibility study for MKCE ChatBot	8
3.1	Block Diagram of MKCE ChatBot	11
4.1	Output Screenshot	15

LIST OF ACRONYMS / ABBREVIATIONS

MYSQL	My Structured Query Language
LAMP	Linux, Apache, MySQL, Perl/PHP/Python
RDBMS	Relational Database Management System

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Web Based MKCE Chatbot is to create an application that can provide college related query management system through web application. This System web application tracks all the difficulties of a student or user from the day one to the end of his course which can be used to discuss about all reporting purpose, tracking enrollment, progress in the course, completed semester years, final exam query and all queries clarified by using the system. We are proposing the application that will help to students as well as to parents or user. The primary goal of the MKCE Chatbot is to provide an environment that is both convenient and efficient to use in retrieving and storing the information into database. The database system must provide the safety to the information stored, despite system crashes or attempts of unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results. In this project the user can register and login the system using his/her user name and password. After the login process the user can post his/her college related query to admin. In this project the admin can login the system using his/her user name and password. After the login process the admin can view the user's query details the he/she updates the solution of user's query. If the admin update the solution of user query, the user can receive the automatic notification of his/her registered mobile number. Then he/she check the solution updated by college admin. In this project the overall process is maintain and monitored by admin only.

1.2 DOMAIN INTRODUCTION

FRONT END: 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. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community. 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 interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favor of a simpler, less-cluttered grammar. As Alex Martelli put it: "To describe something as 'clever' is not considered a compliment in the Python culture." Python's philosophy rejects the Perl "there is more than one way to do it" approach to language design in favour of "there should be one—and preferably only one—obvious way to do it".

Python developers strive to avoid premature optimization, and reject patches to non-critical parts of CPython that would offer marginal increases in speed at the cost of clarity.[When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. CPython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard for and bar.

A common neologism in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is *pythonic* is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called *unpythonic*. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as *Pythonists*, *Pythonistas*, and *Pythoneers*. Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python's initial development was spearheaded by Guido van Rossum in the late 1980s. Today, it is developed by the Python Software Foundation. Because Python is a multiparadigm language, Python programmers can accomplish their tasks using different styles of programming:

object oriented, imperative, functional or reflective. Python can be used in Web development, numeric programming, game development, serial port access and more.

There are two attributes that make development time in Python faster than in other programming languages:

1. Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code. Python focuses so much on this abstraction that its code can be understood by most novice programmers.
2. Python code tends to be shorter than comparable codes. Although Python offers fast development times, it lags slightly in terms of execution time. Compared to fully compiling languages like C and C++, Python programs execute slower. Of course, with the processing speeds of computers these days, the speed differences are usually only observed in benchmarking tests, not in real-world operations. In most cases, Python is already included in Linux distributions and Mac OS X machines.

BACK END: MYSQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google(though not for searches), Imagebook, Twitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator). MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community

1.3 PROBLEM STATEMENT

The existing system is a manual one. When the student, parent or user wants to clear his/her query he/she personally contacts other students or concerned person in the college or university. The concerned person will clear his/her doubt. But the Students can pick up the students or staffs at anytime is not possible. The existing system is handled manually. The system follows large number of paper work for maintaining query or enquiry details and user can be difficult to search the enquiry details in manual process. In this existing system takes lots of time for searching particular enquiry information.

Time consuming: Getting the required information from the available data takes a lot of time. Changing, editing and updating the information contained in several files are a slow and time consuming process.

Need of Effort: In manual system, a discussion information record is maintained in separate files so it takes much effort to collect data from several Stores for and if we want to change or delete the data of any updating then it has to be changed or deleted from all the files and places it stored.

Needs Large Space: In manual work done data item has to be stored at several places, similarly student's record is maintained in separate registers. It requires more storage space.

1.4 OBJECTIVE

The Web Based MKCE Chatbot System is to replace the existing manual system with a software solution. The proposed system is developed after a detailed study about the requirements requested by the user. Proposed system is a computerized one, where all the limitations of manual system are compensated. Query or enquiry details of online MKCE Chatbot have simplified the working information and make a user friendly environment, where the user is provided with much flexibility to manage effectively. It helps the admin to generate desirable reports more quickly and also to produce better results.

CHAPTER 2

1. FEASIBILITY STUDY

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. “FEASIBILITY STUDY” is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

- What are the user’s demonstrable needs and how does a candidate system meet them?
- What resources are available for given candidate system?
- What are the likely impacts of the candidate system on the organization?
- Whether it is worth to solve the problem?

During feasibility analysis for this project, events and alerts are to be considered. Investigation and generating ideas about a new system does this.

TECHNICAL FEASIBILITY

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

- Can the work for the project be done with current equipment existing software technology & available personal?
- Can the system be upgraded if developed?
- If new technology is needed then what can be developed?

ECONOMICAL FEASIBILITY

Economic justification is generally the “Bottom Line” consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase. The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced cost.
- The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.
- This feasibility checks whether the system can be developed with events and alert monitoring does not require the manual work. This can be done economically if planned judiciously, so it is economically feasible. The cost of project depends upon the number of man hours required.

OPERATIONAL FEASIBILITY

It is mainly related to human organizations and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organization structures are disturbed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the End users to operate it. It only needs basic information about Windows platform.

SCHEDULE FEASIBILITY

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems. A reliable VM monitoring system can be developed in the considerable amount of time.

3. PROJECT METHODOLOGY

3.1 BLOCK DIAGRAM

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software
- The composite of the design architectures for products and their life-cycle process-

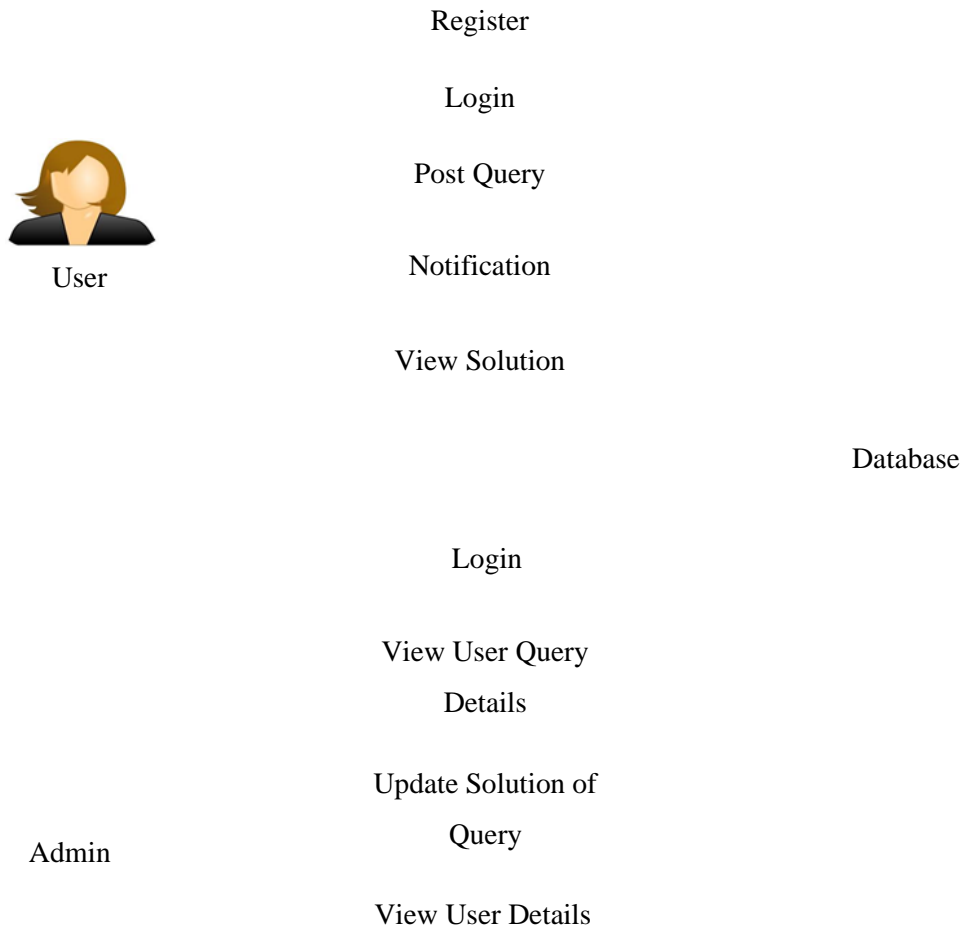


Figure 3.1

BLOCK DIAGRAM OF MKCE ChatBot

3.2 MODULE DESCRIPTION

MODULES

Admin

- Login
- View User Query Details
- Update Solution Of Query
- View User Details

User

- Register
- Login
- Post Query
- Notification
- View Solution

MODULE DESCRIPTION

Admin

- Login

In this module, the admin can login the system using his/her username and password.

- View User Query Details

After the login process the admin can view the user query details.

- Update Solution Of Query

Using this module the admin can post the solution of the user query.

- View User Details

In this module, the admin can view the user information's like user name, email, gender, mobile number, address, and etc.

User

- Register

There is registration form available where new user can create their account by providing required information to the system. The registration form details are like user name, email, gender, mobile number, address, and etc. These details are stored in the database. And then can getting to the username and password in the system.

- Login

In this module, user can login the system using his/her username and password.

- Post Query

After the login process the user can post his/her query related to college. The user query details are sending to the admin module.

- Notification

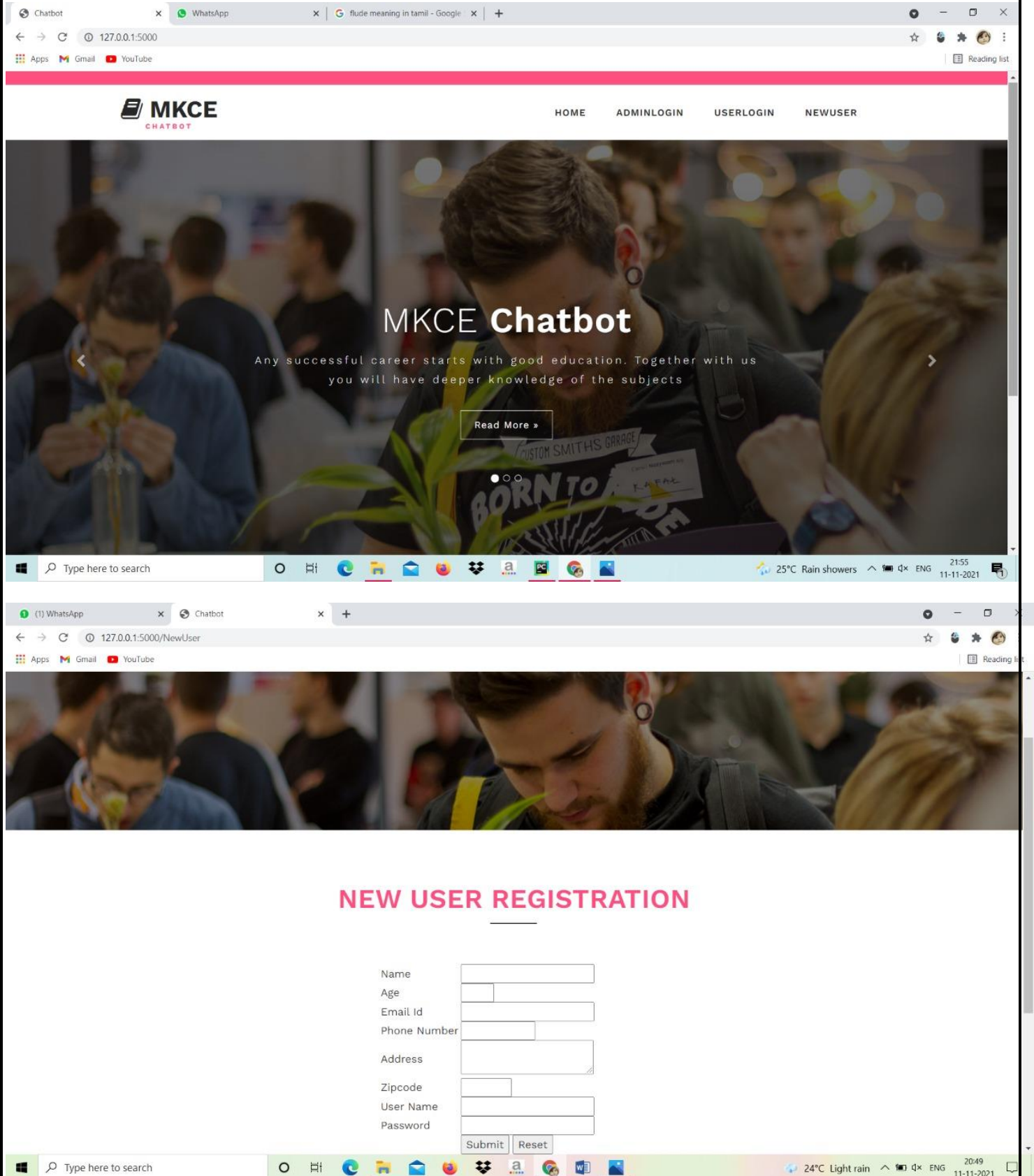
If the admin post the solution of user query, the automatic notification is send to that user's registered mobile number.

- View Solution

Using this module the user can view the solution of his/her query.

CHAPTER 4


RESULTS AND DISCUSSION



(1) WhatsApp x Chatbot x +

127.0.0.1:5000/NewUser

Apps Gmail YouTube Reading list



NEW USER REGISTRATION

Name

Age

Email Id

Phone Number

Address

Zipcode

User Name

Password

Type here to search

25°C Light rain 20:52 11-11-2021


(1) WhatsApp x Chatbot x +

127.0.0.1:5000/UserLogin

Apps Gmail YouTube Reading list

MKCE
CHATBOT

HOME ADMINLOGIN USERLOGIN NEWUSER



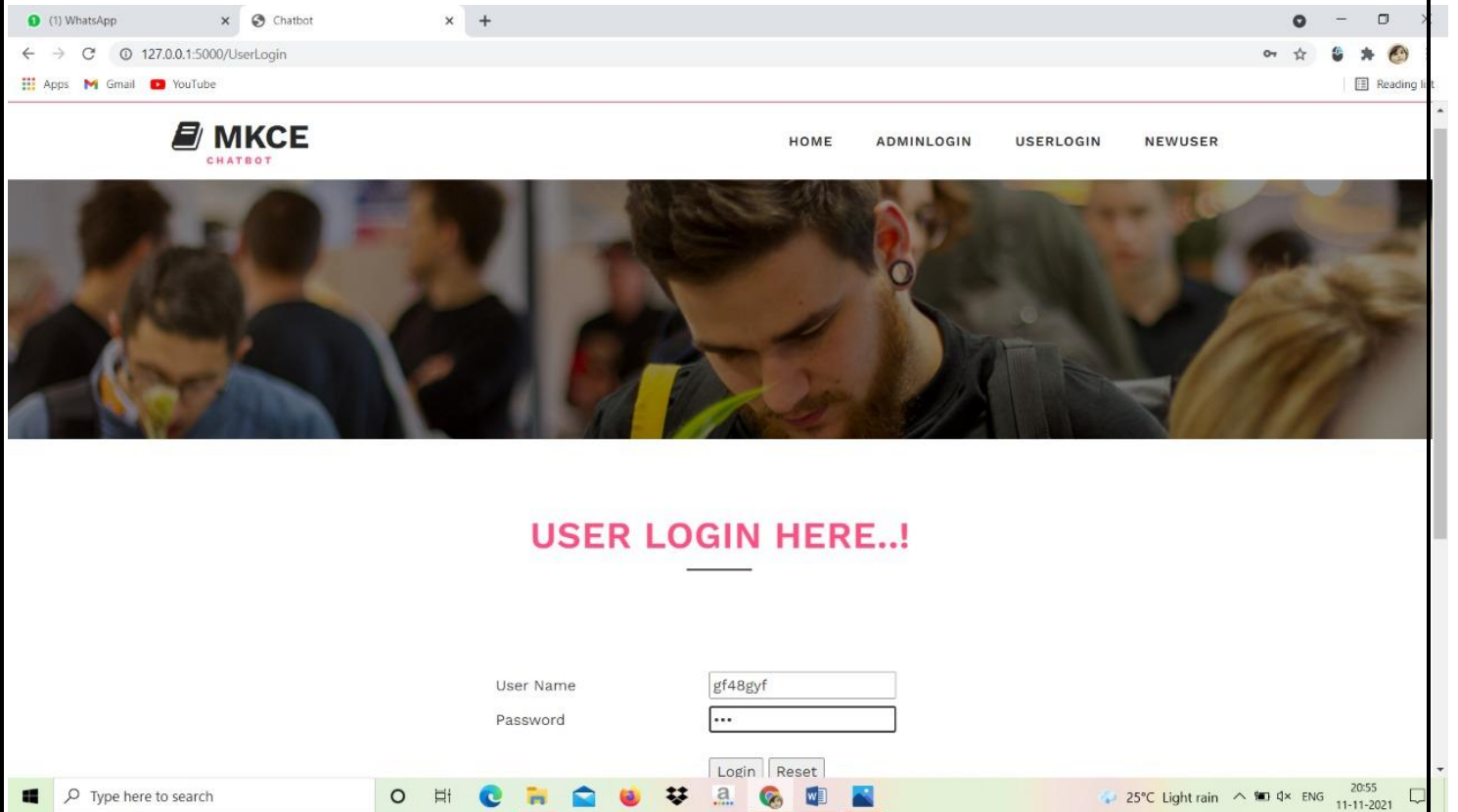
USER LOGIN HERE..!

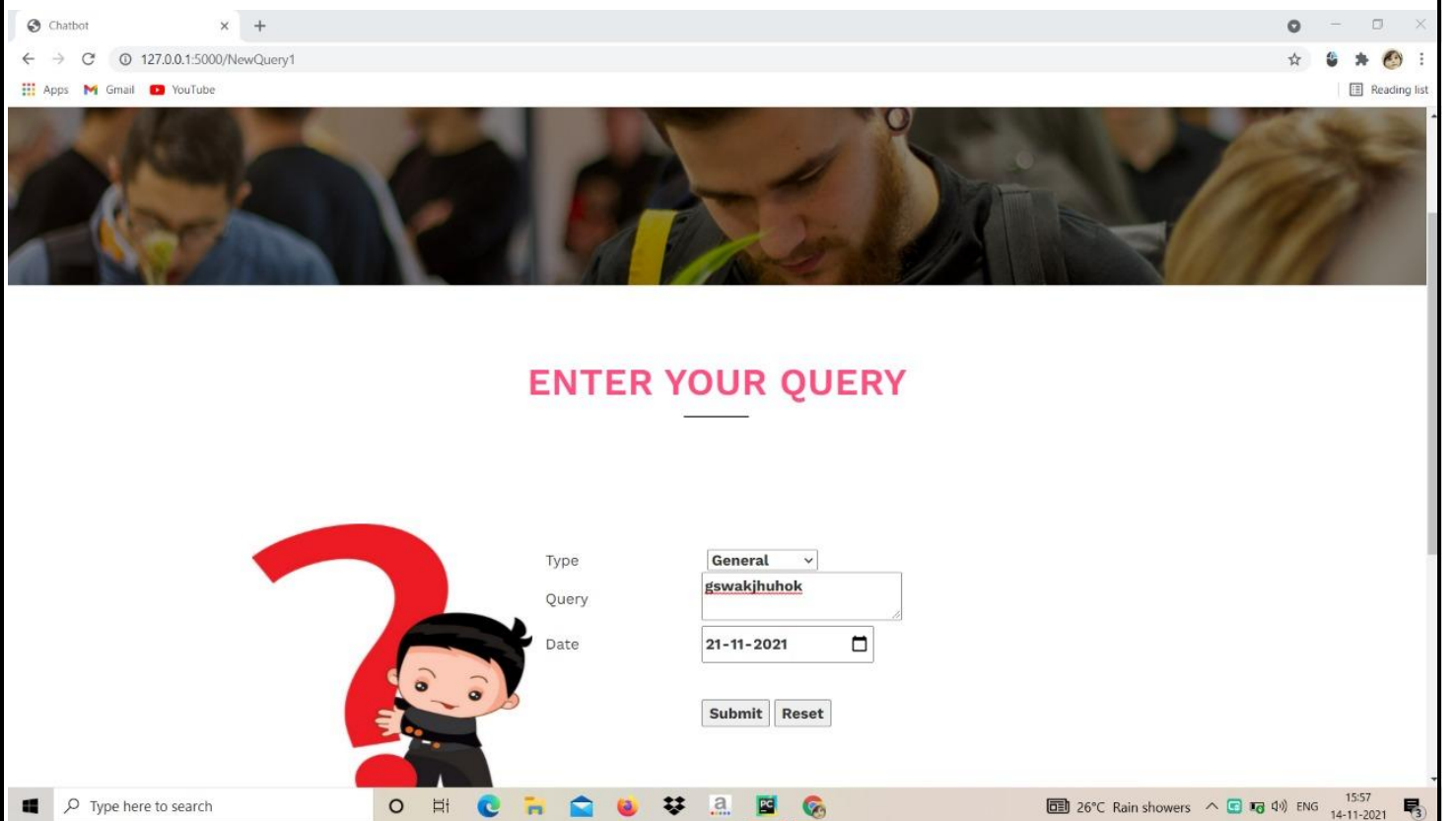
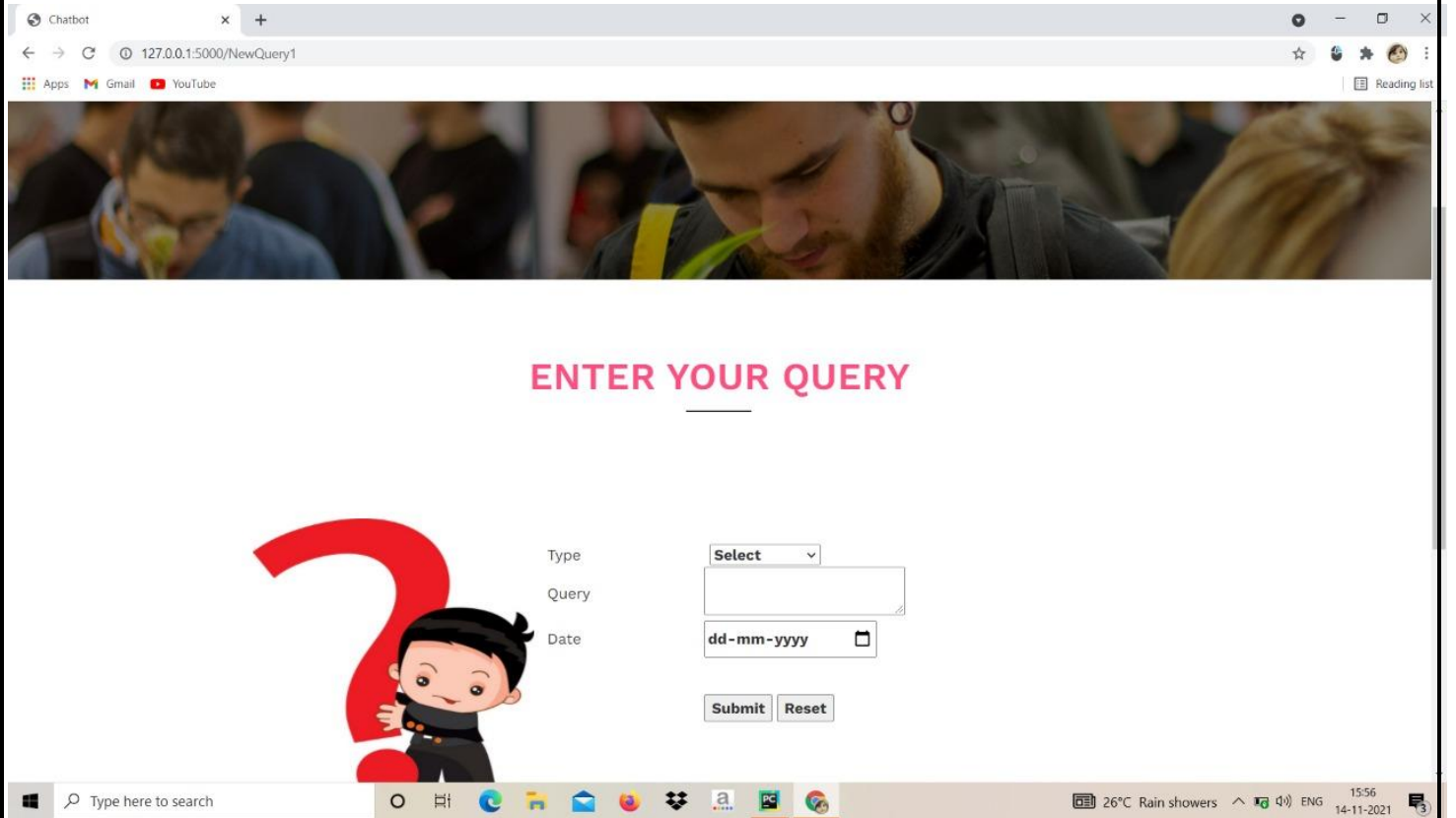
User Name

Password

Type here to search

25°C Light rain 20:54 11-11-2021






Chatbot


127.0.0.1:5000/newquery

AppsGmailYouTube

Reading list




HOME NEWQUERY QUERY&ANSWERINFO LOGOUT



YOUR'S QUERY INFORMATION

UserName	QType	Query	QDate	Status
yogi	General	gswakjhuhok	2021-11-21	waiting

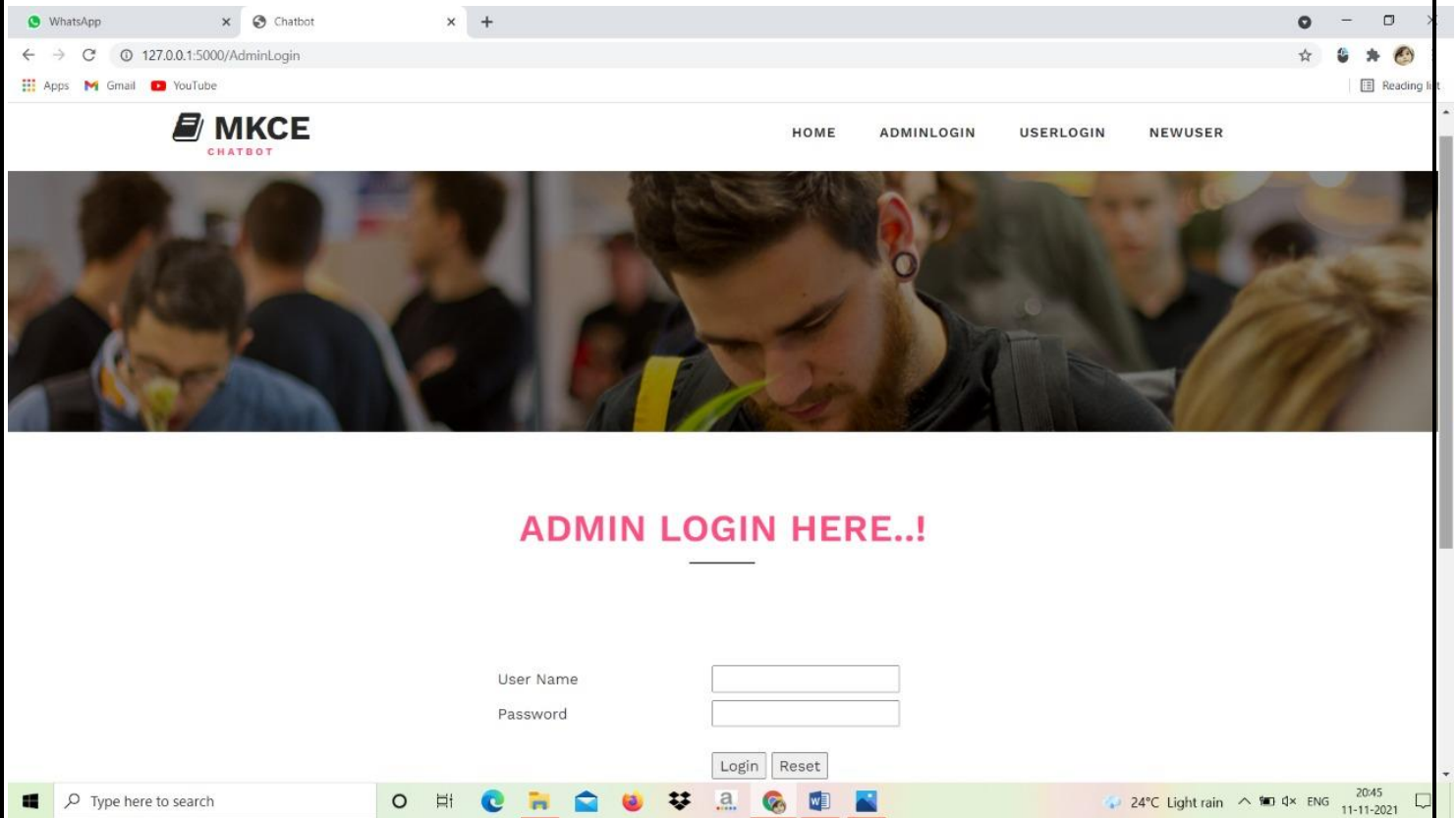
Type here to search

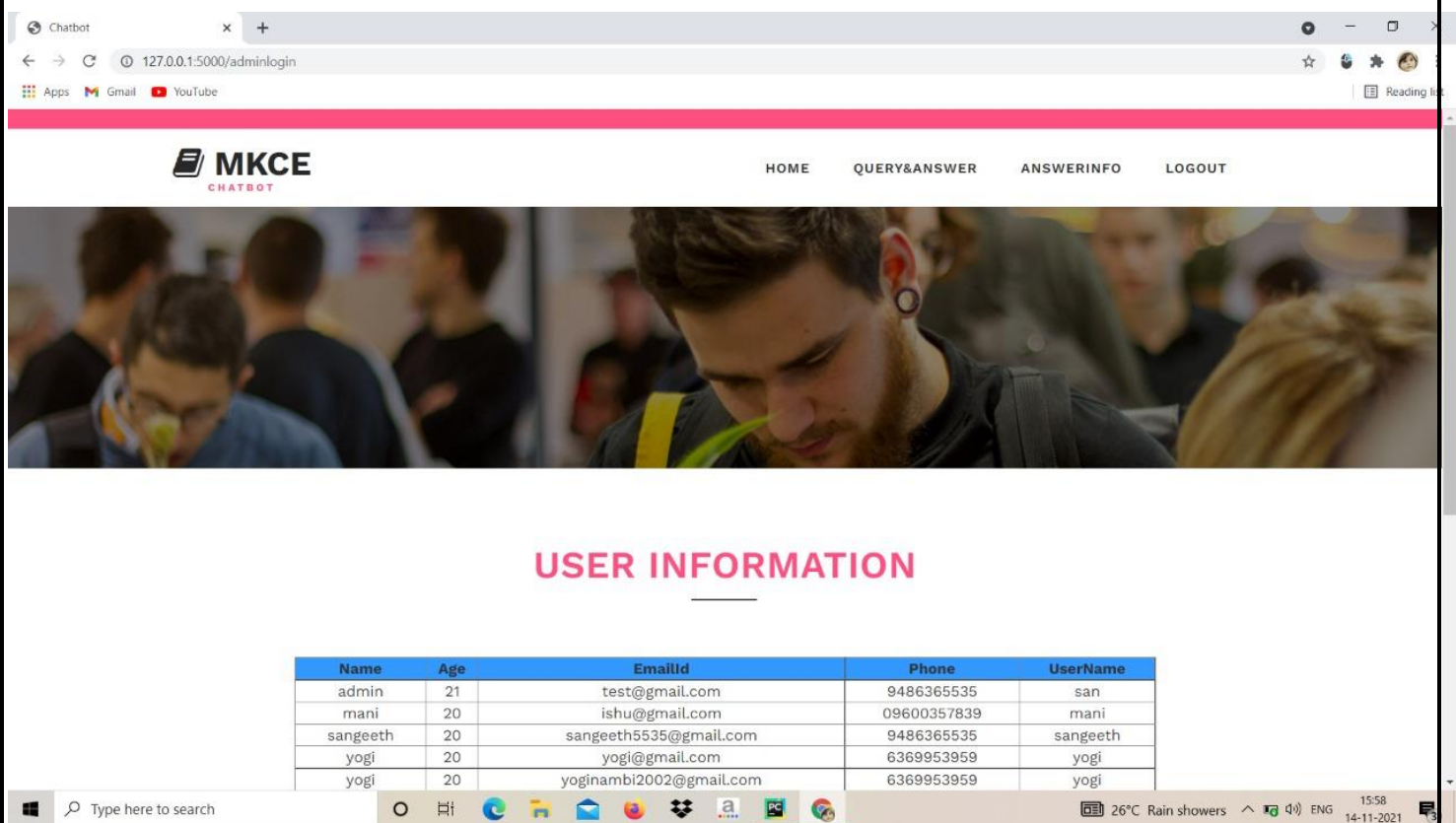


26°C Rain showers

ENG

15:57
14-11-2021






Chatbot

127.0.0.1:5000/AdminQInfo

Apps Gmail YouTube

CHATBOT



QUERY INFORMATION

Enter Answer

Id	UserName	QType	Query	QDate	Status	Answer
7	yogi	General	gswakjhuhok	2021-11-21	waiting	Reply

Type here to search

26°C Rain showers

ENG

15:59


14-11-2021

Chatbot


127.0.0.1:5000/newquery

Apps Gmail YouTube

Reading list




HOME NEWQUERY QUERY&ANSWERINFO LOGOUT



YOUR'S QUERY INFORMATION

UserName	QType	Query	QDate	Status
yogi	General	gswakjhuhok	2021-11-21	waiting

Type here to search



26°C Rain showers

ENG

15:57
14-11-2021

Chatbot


127.0.0.1:5000/answer

Apps


Gmail

YouTube

Reading list




HOME QUERY&ANSWER ANSWERINFO LOGOUT



ANSWER INFORMATION

Id	UserName	QType	Query	QDate	Status	Answer
3	yogi	Subject	what is java	2021-10-26	Answer	program language
4	yogi	General	leave	2021-11-20	Answer	
7	yogi	General	gswakjhuhok	2021-11-21	Answer	kjhsadjsfmj

Type here to search



26°C Rain showers

ENG

15:59

14-11-2021

5. CONCLUSION

This project entitled as “MKCE Chatbot” has been developed to satisfy all the proposed requirements. The process of recording details about college related user query details is more simple and easy. The system reduces the possibility of errors to a great extent and maintains the data in an efficient manner. User friendliness is the unique feature of this system. The system generates the reports as and when required. The system is highly interactive and flexible for further enhancement. The coding is done in a simplified and easy to understandable manner so that other team trying to enhance the project can do so without facing much difficulty. The documentation will also assist in the process as it has also been carried out in a simplified and concise way.

FUTURE ENHANCEMENT

In future we can develop this project in android application with extra features like online admission process.

REFERENCES

BOOK REFERENCES

- Van Rossum, Guido, and Fred L. Drake. The python language reference manual. Network Theory Ltd., 2011.
- Dierbach, Charles. Introduction to Computer Science using Python: A Computational Problem-Solving Focus. Wiley Publishing, 2012.
- James, Mike. Programmer's Python: Everything is an Object Something Completely Different. I/O Press 2018.
- Reges, Stuart, Marty Stepp, and Allison Obourn. Building Python Programs. Pearson, 2018.

WEBSITE REFERENCES

- <https://docs.python.org/3/tutorial/>
- <https://www.w3schools.com/python/>
- <https://www.tutorialspoint.com/python/index.htm>
- <https://www.programiz.com/python-programming>
- <https://www.learnpython.org/>

APPENDIX

CODING

```
from flask import Flask, render_template, flash, request, session
from flask import render_template, redirect, url_for, request
#from wtforms import Form, TextField, TextAreaField, validators, StringField, SubmitField
#from werkzeug.utils import secure_filename

import mysql.connector
import smtplib

app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
app.config['DEBUG']

@app.route("/")
def homepage():
    return render_template('index.html')

@app.route("/Home")
def Home():
    return render_template('index.html')

@app.route("/AdminLogin")
def AdminLogin():
    return render_template('AdminLogin.html')

@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')

@app.route("/UserLogin")
```

```
def UserLogin():
    return render_template('UserLogin.html')
@app.route("/UserHome")
def UserHome():
    return render_template('UserHome.html')
@app.route("/AdminHome")
def AdminHome():
    return render_template('AdminHome.html')
@app.route("/NewQuery1")
def NewQuery1():
    return render_template('NewQueryReg.html')
@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
        if request.form['uname'] == 'admin' or request.form['password'] == 'admin':
            conn = mysql.connector.connect(user='root', password="", host='localhost', database='l chatbotdb')
            cursor = conn.cursor()
            cur = conn.cursor()
            cur.execute("SELECT * FROM register")
            data = cur.fetchall()
            return render_template('AdminHome.html', data=data)
        else:
            return render_template('index.html', error=error)
```

```
@app.route("/reg", methods=['GET', 'POST'])
def reg():
    if request.method == 'POST':
        n = request.form['name']
        address = request.form['address']
        age = request.form['age']
        pnumber = request.form['phone']
        email = request.form['email']
        zip = request.form['zip']
        uname = request.form['uname']
        password = request.form['psw']
        conn = mysql.connector.connect(user='root', password='', host='localhost', database='1chatbotdb')
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO register VALUES ('" + n + "','" + age + "','" + email + "','" + pnumber + "','" + zip + "','" +
address + "','" + uname + "','" + password + "')"
        )
        conn.commit()
        conn.close()
        # return 'file register successfully'
        return render_template('UserLogin.html')
@app.route("/userlogin", methods=['GET', 'POST'])
```

```

password = request.form['password']

session['uname'] = request.form['uname']


conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
cursor = conn.cursor()
cursor.execute("SELECT * from register where uname='" + username + "' and psw='" + password + "'")
data = cursor.fetchone()

if data is None:
    return render_template('index.html')
    return 'Username or Password is wrong'
else:
    print(data[0])
    session['uid'] = data[0]

    conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM register where uname='" + username + "' and psw='" + password + "'")
    data = cur.fetchall()

    return render_template('UserHome.html', data=data )


@app.route("/newquery", methods=['GET', 'POST'])
def newquery():

```



```
if request.method == 'POST':
    uname = session['uname']
    type = request.form['Qtype']

    query = request.form['query']
    date = request.form['date']

    conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
    cursor = conn.cursor()
    cursor.execute(
        "INSERT INTO Querytb VALUES ('" + uname + "','" + type + "','" + query + "','" + date + "','waiting')")
    conn.commit()
    conn.close()
    # return 'file register successfully'

    conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and status='waiting'")
    data = cur.fetchall()
    return render_template('UserQueryInfo.html', data=data)

@app.route("/UQueryandAns")
def UQueryandAns():
```

```
uname = session['uname']
```

```
conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
```

```
# cursor = conn.cursor()
```

```
cur = conn.cursor()
```

```
cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and status='waiting'")
```

```
data = cur.fetchall()
```

```
conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
```

```
# cursor = conn.cursor()
```

```
cur = conn.cursor()
```

```
cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and status='Answer'")
```

```
data1 = cur.fetchall()
```

```
return render_template('UserQueryAnswerinfo.html', wait=data, answ=data1 )
```

```
@app.route("/AdminQinfo")
```

```
def AdminQinfo():
```

```
    #uname = session['uname']
```

```
    conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
```

```
    # cursor = conn.cursor()
```

```
    cur = conn.cursor()
```

```
    cur.execute("SELECT * FROM Querytb where status='waiting'")
```

```
    data = cur.fetchall()
```

```

return render_template('AdminQueryInfo.html', data=data )

@app.route("/answer", methods=['GET', 'POST'])
def answer():
    if request.method == 'POST':
        Answer = request.form['AAnswer']
        id = request.form['id']
        uname = request.form['uname']
        print(Answer)
        print(id)
        conn = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
        cursor = conn.cursor()
        cursor.execute(
            "update Querytb set status='Answer',Answer='"+Answer+" where id='"+ str(id) + "' ")
        conn.commit()
        conn.close()
        conn3 = mysql.connector.connect(user='root', password="", host='localhost', database='lchatbotdb')
        cur3 = conn3.cursor()
        cur3.execute("SELECT * FROM register where  uname='"+ str(uname) + "'")
        data3 = cur3.fetchone()
        if data3:
            phnumber = data3[4]
            print(phnumber)
            sendmsg(phnumber, "Your Query Answer updated!")

```

```

# return 'file register successfully'

conn = mysql.connector.connect(user='root', password="", host='localhost', database='1chatbotdb')

# cursor = conn.cursor()

cur = conn.cursor()

cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and status !='waiting'")

data = cur.fetchall()

return render_template('AdminAnswer.html', data=data)

@app.route("/AdminAinfo")

def AdminAinfo():

    conn = mysql.connector.connect(user='root', password="", host='localhost', database='1chatbotdb')

    # cursor = conn.cursor()

    cur = conn.cursor()

    cur.execute("SELECT * FROM Querytb where status !='waiting'")

    data = cur.fetchall()

    return render_template('AdminAnswer.html', data=data )

def sendmsg(targetno,message ):

    import requests    requests.post("http://smsserver9.creativepoint.in/api.php?username=fantasy&password=596692&to="
+ targetno + "&from=FSSMSS&message=Dear user your msg is " + message + " Sent By FSMSG
FSSMSS&PEID=1501563800000030506&templateid=1507162882948811640")

if __name__ == '__main__':

    app.run(debug=True, use_reloader=True)

```