

# **Chatbot Management System**

**A Project - III**

**Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of  
Technology in Computer Science & Information Technology**

**Submitted to:**



**RAJIV GANDHI PRODYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)**

**Submitted by:**

**Jaydeep Sahu – 0808CI191059**

**Harsh Gupta– 0808CI191051**

**Under the Supervision of:**

**Mr. Somil Neema  
Assistant Professor  
Department of CSE**



**IPS ACADEMY INDORE**

**INSTITUTE OF ENGINEERING & SCIENCE**

**(A UGC Autonomous Institute Affiliated to RGPV)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COMPUTER SCIENCE & INFORMATION TECHNOLOGY BRANCH**

**SESSION: 2022-23**

**IPS Academy, Indore**  
**Institute of Engineering and Science**  
(A UGC Autonomous Institute Affiliated to RGPV)

**Department of Computer Science & Engineering**  
**Computer Science & Information Technology Branch**

**2022-23**



*Project-III entitled*

*“Chatbot Management System”*

*For the partial fulfillment for the award of the Bachelor of Technology (Computer Science & Information Technology) Degree by Rajiv Gandhi Proudhyogiki Vishwavidhyalaya, Bhopal.*

**Guided by: -**

Mr. Somil Neema  
Assistant Professor  
Department of CSE

**Submitted by: -**

Jaydeep Sahu (0808CI191059)  
Harsh Gupta (0808CI191051)

**IPS Academy, Indore**  
**Institute of Engineering and Science**  
(A UGC Autonomous Institute Affiliated to RGPV)  
**Department of Computer Science & Engineering**  
**Computer Science & Information Technology Branch**  
**2022-23**



*CERTIFICATE*

This is to certify that Project - III entitled

*“CHATBOT MANAGEMENT SYSTEM”*

*has been successfully completed by the following student*

*Harsh Gupta (0808CI191051)*

*in partial fulfillment for the award of the Bachelor of Technology (Computer Science & Information Technology) Degree by Rajiv Gandhi Pradyogiki Vishwavidhyalaya, Bhopal during the academic year 2022-23 under our guidance.*

Mr. Somil Neema  
Assistant Professor, CSE

Dr. Vaishali Gupta  
Branch Coordinator

Dr. Neeraj Shrivastava  
HOD CSE

Dr. Archana Keerti Chowdhary  
Principal

## **Acknowledgement**

I would like to express my heartfelt thanks to my guide **Mr. Somil Neema, Assistant Professor CSIT**, for his guidance, support, and encouragement during the course of my study for B.Tech (CSIT) at IPS Academy, Institute of Engineering & Science Indore. Without his endless effort, knowledge, patience, and answers to my numerous questions, this Project would have never been possible. It has been great honor and pleasure for me to do Project under his supervision.

My gratitude will not be complete without mention of **Dr. Archana Keerti Chowdhary, Principal, IPS Academy, Institute of Engineering & Science, Dr. Neeraj Shrivastava, HOD CSE, IPS Academy, Institute of Engineering & Science, and Dr. Vaishali Gupta, Branch Coordinator CSIT, IPS Academy, Institute of Engineering & Science** for the encouragement and giving me the opportunity for this project work.

I also thank my friends who have spread their valuable time for discussion/suggestion on the critical aspects of this report. I want to acknowledge the contribution of my parents and my family members, for their constant motivation and inspiration.

Finally I thank the almighty God who has been my guardian and a source of strength and hope in this period.

**Jaydeep Sahu (0808CI191059)**

**Harsh Gupta (0808CI191051)**

# CONTENTS

<b>List of Figures</b>	<b>i</b>
<b>List of Tables</b>	<b>ii</b>
<b>List of Abbreviation</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Overview	2
1.2 Literature Survey	5
<b>CHAPTER 2: PROBLEM IDENTIFICATION &amp; FEASIBILITY ANALYSIS</b>	<b>7</b>
2.1 Problem Domain	8
2.2 Solution Domain	8
2.3 Need & Scope	9
2.4 Feasibility Analysis	9
<b>CHAPTER 3: SOFTWARE ENGINEERING APPROACH</b>	<b>12</b>
3.1 Software model used	13
3.1.1 Description	
3.1.2 Reason for use	
3.2 Platform Specification	15
3.2.1 Hardware Specification	
3.2.2 Software Specifications	
<b>CHAPTER 4: DESIGNS</b>	<b>16</b>
4.1 Use Case Diagram	17
4.2 Sequence Diagram	18
4.2.1 Sequence Diagram for User Authentication	
4.2.2 Sequence Diagram for Dashboard	
4.3 Activity Diagram	20
4.4 Database Design	21
	<b>25</b>

<b>CHAPTER 5 : IMPLEMENTATION PHASE</b>	
5.1 Language Used & its Characteristics	26
5.2 GUI (Snapshots)	28
 <b>CHAPTER 6: TESTING METHOD</b>	 <b>35</b>
6.1 Testing Method	36
6.2 Test Cases	36
 <b>CHAPTER 7 : CONCLUSION</b>	 <b>42</b>
 <b>CHAPTER 8: LIMITATIONS &amp; FUTURE ENHANCEMENTS</b>	 <b>44</b>
 <b>REFERENCES</b>	 <b>47</b>

## LIST OF FIGURES

<b>Figure No</b>	<b>Title</b>	<b>Page No.</b>
Figure 1.1	Basic Model of Chatbot	2
Figure 1.2	Chatbot Management System	4
Figure 1.3	HR Platform	5
Figure 1.4	Salesforce Dashboard	6
Figure 2.1	Time Line Chart	10
Figure 3.1	Incremental Model	13
Figure 4.1	Use Case Diagram	17
Figure 4.2.1	Sequence Diagram for user authentication	18
Figure 4.2.2	Sequence Diagram for Dashboard	19
Figure 4.3	Activity Diagram	20
Figure 5.1	Login Page	28
Figure 5.2	Home Page	29
Figure 5.3	Faculty Data Visualization	30
Figure 5.4	Notice Data Visualization	30
Figure 5.5	Subscribed Data Visualization	31
Figure 5.6	Students Details	32
Figure 5.7	Campus Guide	32
Figure 5.8	Add Users	33
Figure 5.9	Alert System	33
Figure 5.10	Campus Guide – Add Faculty Details	34
Figure 5.11	Campus Guide – Edit Faculty Details	35

## LIST OF TABLES

<b>Table No.</b>	<b>Table Name</b>	<b>Page No.</b>
Table 4.1	User Table	21
Table 4.2	Students Table	21
Table 4.3	Syllabus	22
Table 4.4	RGPV Notes	22
Table 4.5	Notice Board	23
Table 4.6	Faculty Details	23
Table 4.7	Users	24
Table 4.8	Login	24
Table 6.1	Test Case – Login Page	37
Table 6.2	Test Case – Home Page	38
Table 6.3	Test Case – Chatbot User Module	39
Table 6.4	Test Case – Students Detail Module	39
Table 6.5	Test Case – Campus Guide Module	40
Table 6.6	Test Case – Add User Module	41
Table 6.7	Test Case - Alert System Module	41



## **LIST OF ABBREVIATION**

<b>AI</b>	<b>Artificial Intelligence</b>
<b>APK</b>	<b>Application</b>
<b>DB</b>	<b>Database</b>
<b>DBMS</b>	<b>Database Management System</b>
<b>GUI</b>	<b>Graphical User Interface</b>
<b>JSON</b>	<b>Javascript Object Notation</b>
<b>NLP</b>	<b>Natural Language Processing</b>
<b>SQL</b>	<b>Structured Query Language</b>
<b>UI</b>	<b>User Interface</b>
<b>RAM</b>	<b>Random Access Memory</b>
<b>HDD</b>	<b>Hard Disk Drive</b>
<b>SSD</b>	<b>Solid State Drive</b>
<b>IDE</b>	<b>Integrated Development Environment</b>

# **ABSTRACT**

The major project entitled “Chatbot Management System” is an extension to the minor project entitled “Institutional Information Chatbot”. It is a web application focusing on creating an Chatbot Management System that enables user to manage, monitor and maintain the chatbot data, along with analytical abilities with the help of different kind of Graphs, Charts etc. using the dashboard. It also provides few unique features that complements the chatbot and helps in establishing college information chatbot system. It will enable chatbot admins/faculty members to user the chatbot more conveniently, effectively and efficiently. To build this project we have used HTML, CSS, Bootstrap, JavaScript (ES6), JQuery (Datatable plugin), chart.js, for frontend development and for backend development Python programming language, Flask-API, fast-API, Telegram API and for database PostgreSQL is used with the help of SupaBase-Cloud Platform and the project is fully deployed on Heroku.

# **CHAPTER - 1**

## **INTRODUCTION**

## 1.1 OVERVIEW

The project “Chatbot Management System” mainly focuses on creating a web application that can be used by faculties of college to provide an interactive and eye pleasing user interface to manage institution’s Chatbot. It enables an admin to maintain, manage and visualize the data of chatbot and helps in extension of features of chatbot.

### 1.1.1 Chatbot

A chatbot is an intelligent piece of software that is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites and instantly messaging the client.

There are two basic types of chatbot models based on how they are built; Retrieval based and Generative based models.

**1. Retrieval based Chat bots:** A retrieval-based chatbot uses predefined input patterns and responses. It then uses some type of heuristic approach to select the appropriate response. It is widely used in the industry to make goal-oriented chatbots where we can customize the tone and flow of the chatbot to drive our customers with the best experience. Retrieval-based chatbots use techniques like keywords matching, machine learning or deep learning to identify the most appropriate response.

**2. Generative based Chat bots:** Generative models are not based on some predefined responses. They are based on seq2 neural networks. It is the same idea as machine translation. In machine translation, we translate the source code from one language to another language but here, we are going to transform input into an output. It needs a large amount of data and it is based on Deep Neural networks.

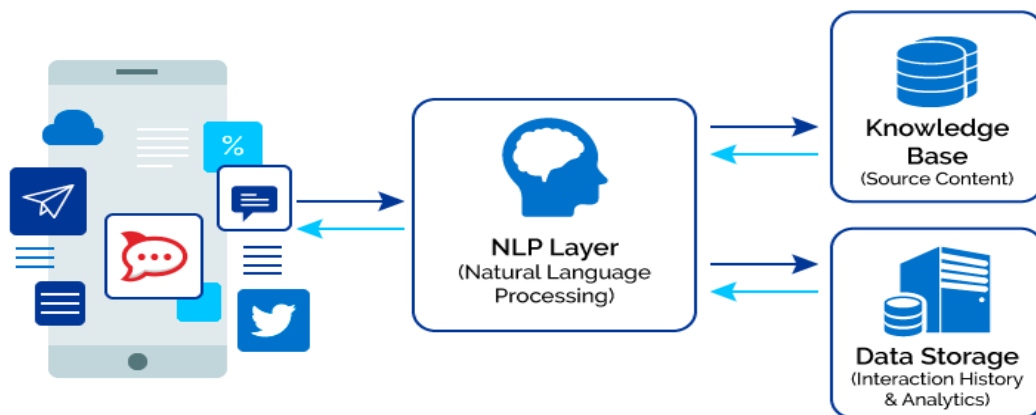


Fig 1.1 “Basic Model of a chatbot”

### **1.1.2 Minor Project : ‘Institutional Information Chatbot’**

This minor project entitled “Institutional Information Chatbot” is a Retrieval based chatbot on one of the most popular messaging platforms- ‘Telegram’, implemented using Telegram’s public API, on top of web-scraping done on Institute’s website (ies.ipsacademy.org). This is Retrieval Based Chat-Bot for a Messaging Platform (Telegram) for Services offered by Institute’s Website.

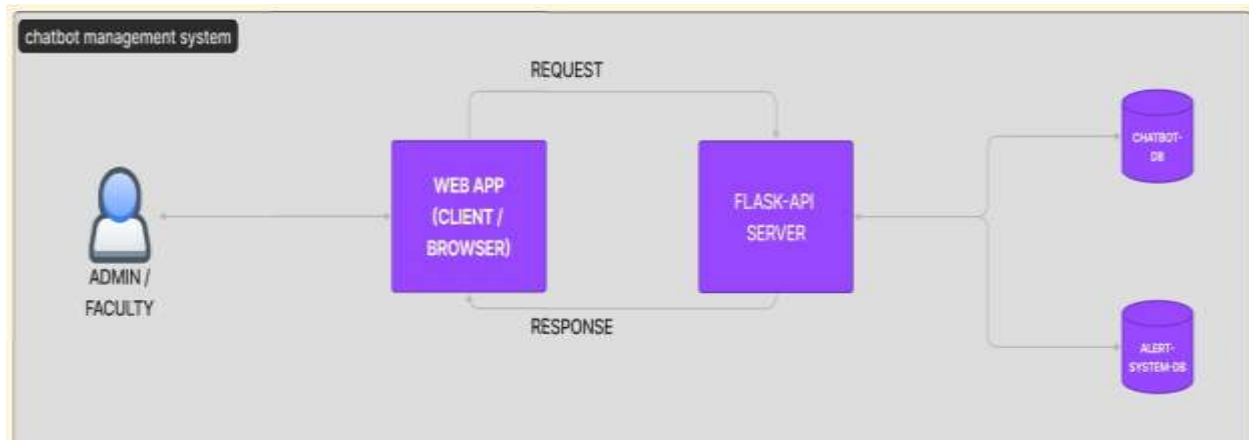
The basic functionality of chatbot is to interact with the user to answer various queries related to the college like notice-board, schemes/syllabus, academic calendar etc. Some of the unique features added in our minor project is campus guide, personalized notices & syllabus, notes & alert system.

In minor project we use Postgres database for storing details of students and used later to provide personalized result to end user. The application is deployed on Heroku platform which can be easily accessed by user on Telegram App.

The main idea behind building the chatbot is to make it easier for the student to get useful information, news and documents with a simple command/chat without having to worry about going from one tab to another tab or getting stuck on one page due to slow network speed. The major advantage of chatbot is the ease of using it. The chat-bot is built on top of an api that provides useful information extracted from institute’s website with web-scraping techniques

### **1.1.3 Extension in Minor Project : Chatbot Management System**

The project entitled “Chatbot Management System” is a extension in our minor project (“Institutional Information Chatbot”). Chatbot Management System is a web application that facilitates the easy maintenance of our chatbot by authorized user along with data visualization and scrapping management modules. Admin/Faculty interacts with web app (Chatbot Management System) and application interacts with Chatbot Database and Alert System Database using a Flask API. The web app is designed using a language python for creating API, some libraries of Javascript i.e chart.js use to show various graphs & charts. Chatbot Management System is a web Application enabled with various features like Data Visualization, Notes System, Campus Guide, User Management, Scraper Management, User Authentication, Customized Alert Bot which makes the easy maintainability of Chatbot and easy availability of user’s data. Web Application is based on request/response model.



**Fig. 1.2 Chatbot Management System**

#### **1.1.4 What does our project provides?**

The major project is a web application build using request/response model which provides the effective management of institution's chatbot build on one of the most popular messaging platform – 'Telegram', implemented using Telegram's public API using Flask API. The Chatbot Management System provide the various feature to manage data such as notes, campus guide, faculty & user details for Chatbot. Our platform provide the user interface so that without using SQL queries by user, he/she can easily update/add/delete the notes, faculty details.

Chatbot Management System embedded with unique feature of alert system, Registered User on institute's Chatbot on Telegram will get alerts message if the new notice is arrived. Also the faculty can directly send customized alert to the registered students within in single click.

Another important use of this web app is that it use to maintain the correct information in database which create the trust of end user of Chatbot as he/she get correct information. The admin/faculty can effectively update details of faculty, update/add notes using Flask API so that they do not need to have a knowledge of SQL Queries.

This web also provides chats and graph to analyze the data generated on Chatbot & present on database so that admin can analyze and take effective steps to add new feature on Chatbot.

#### **1.1.5 Why Chatbot Management System?**

When we have to retrieve data from database or update data in Chatbot database, one have to run SQL queries which need a technical knowledge of SQL and difficult for most to its users. Chatbot Management System is a solution of that problem which provide an efficient, effective and eye

pleasant interface for performing such operation and any person with non-technical background can also perform this operations easily. It provide the layer of abstraction in accessing and manipulation data from database where API is used to perform operations.

## 1.2 LITERATURE SURVEY

The continuous increase of data leads to information overloading which is a known phenomenon and thus becomes essential to deal with. Dashboards are used to manage this overload since they are considered very useful tools in Business Intelligence (BI) as they use concepts like scorecards to assist employees and stakeholders to make appropriate and quick decisions and to improve performance. This study defines dashboard as “a display or visualization of the highly important information which is used to fulfill necessary objectives, which is consolidated and displayed on a single screen so that all the necessary information can be monitored at once”.

Some Projects related to dashboard to study:

- **HR Platform**

This minimalistic dashboard displays all the data an HR needs on a single screen. Using a pie chart, an HR specialist can understand the ratio of candidates and quickly react if there is a shortage of candidates in one of the application categories. In the bottom right there is a calendar with scheduled interviews, exact times, and desired positions. Here users have everything they need at arm’s length: from indicators of the company’s HR situation to their current tasks. To get more information in each section, a specialist can click the “view all” button and open the full data.

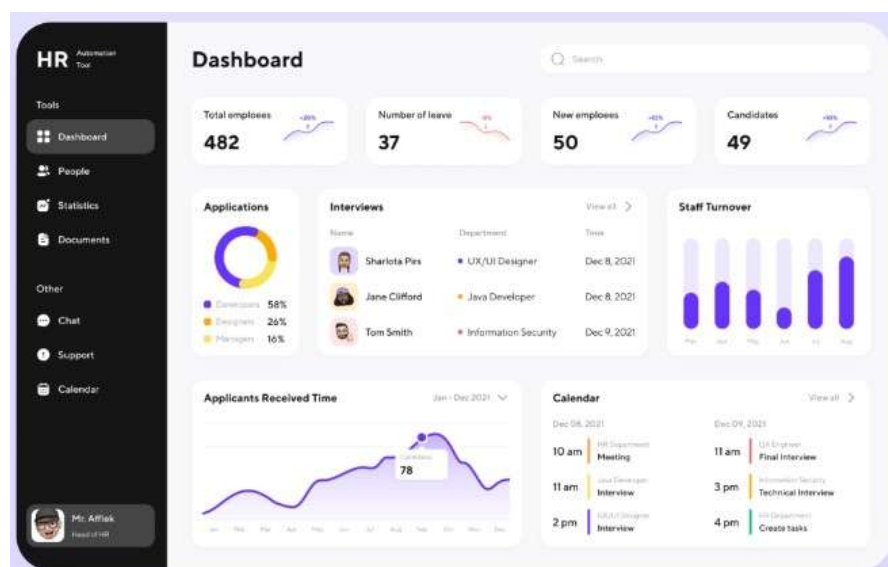


Fig 1.3 “HR Platform”

- **Restaurant Management App**

The dashboard provides the restaurant manager with real-time information, so they can quickly assess the situation and adapt. For example, the number of waiters in the restaurant depends on how many tables are reserved and how many guests are in line. These indicators are emphasized with the help of UI UX design. If it is a chain, the manager can also easily switch between different locations and see the key indicators of each of them.

- **Call Center Dashboard**

A call center dashboard (or contact center dashboard) visualizes live metrics such as number of calls, wait time, and Customer Satisfaction (CSAT). This enables call center teams to create a better customer experience by quickly reacting to changing situations. In physical call centers, these dashboards are usually displayed on TVs where everyone can see them. However, they are also used by remote and hybrid teams. They provide a focal point for performance measurement, helping motivate the team to achieve their goals.

- **Salesforce Dashboard**

This dashboard is designed to provide a clear overview of your company's business KPIs, from revenue figures to activity in the sales pipeline. It pulls data from several Salesforce reports, and visualizes those metrics in a way where it's easy to see your company's progress. It's the perfect dashboard to displayed on a TV in the office where all members of the team can see it.



**Fig. 1.4 “Salesforce Dashboard”**



**CHAPTER – 2**

**PROBLEM IDENTIFICATION &  
FEASIBILITY ANALYSIS**

## **2.1 PROBLEM DOMAIN**

Chatbot is developed on telegram platform for college information but there is no a such a special platform form where we can monitor or update details in database. We have to enter hardcoded data in database. Every Chatbot must have its own platform where it can manage and monitor effectively. Current Existing System has various drawbacks –

- We have to run SQL query for retrieving data from database, There is not such API which get data for us from Database so retrieving a data is very difficult task and non-technical person can not perform retrieval of data.
- We have to run scrapper manually to update data in database from college website, running a scrapper manually means running a code but it can not managed by everyone easily, only a developer can manage it easily.
- There is not any facility for faculty of college to send a customized alert message to the targeted audience with in the single click.
- There is no such platform which provide the visualization of data generated from Chatbot user graphically with the help various different graphs and charts.
- There is no graphical user interface from where Chatbot feature is managed.
- It is difficult to understand the need of Chatbot user without analyze the data, so we need a web app where all data numbers are available at same time on same platform.

## **2.2 SOLUTION DOMAIN**

The Chatbot Management System eliminates almost all of the limitations of existing system. Main objective of proposed system are as follow-

- It is a web application having eye pleasant Graphical User Interface so that working on dashboard is easy for new person also.
- To add/update in database or delete from database, an end user of dashboard/web app has not to run SQL Queries, a person having zero knowledge of SQL can also perform deletion/updation/insertion in database easily.
- We do not need to run code of scrapper, instead of that any dashboard user can run it on a single click.
- Our dashboard provide a platform where admin/faculty see various graphs and charts regarding data present in database and data generated by Chatbot user, which can used for data analytics so that new feature can be added in Chatbot as per user need. Pre-defined templates for alert is

provided so that faculty does not have to type the complete message, along with it customized alerts can also be sent to targeted audience

## **2.3 NEED & SCOPE**

### **2.3.1 Need**

Our project is based on Chatbot Dashboard. Dashboard gives us a full overview of what's happening with out chatbots and ongoing chats. We'd like to focus on what matters, our bot, that's why ChatBot is a user-friendly and easy to understand the environment. The bot dashboard shows key performance indicators (KPIs) and charts that we can use to understand how bots are performing in a support organizations. Dashboards offer a method of consolidating company data into one unified location with secure data storage. Dashboards are designed to offer a comprehensive overview of company performance, and do so through the use of data visualization tools like charts and graphs.

### **2.3.2 Scope**

Chatbot Dashboard can play a vital role in reducing the dependency on developers/technical person to update the content provided in Chatbot. Anyone can easily retrieve, insert, update data easily.

It gives the facility to add notes of respective subjects/add notes of new subjects easily using eye pleasant user interface provided in dashboard.

Dashboard aims to provide the facility to send alerts to specific students if new notice is updated in their respective department.






Dashboard will also be provided with predefined alerts which can directly send to targeted students and they receive it on alert bot on telegram app.

Faculty/admin having proper login credentials can only access dashboard, data present in database, is secured and can be updated only by authorized user.

## **2.4 FEASIBILITY ANALYSIS : -**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization.

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

Development Phase	60 Days Timeline Chart				Duration
	0 to 15 Days	16 to 30 Days	31 to 45 Days	46 to 60 Days	
Requirement Gathering					12 Days
Analysis					15 Days
Design					28 Days
Coding					25 Days
Documentation					48 Days
Total					60 Days

**Table (2.1) - Time line Chart**

#### **2.4.1 Technical Feasibility: -**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?
- The current system developed is technically feasible. Thus it provides an easy access to the users.

#### **2.4.2 Economical feasibility: -**

In economic feasibility, the most important is cost-benefit analysis. For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

#### **2.4.3 Operational feasibility: -**

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?
- Does management support the project?

# **CHAPTER – 3**

## **SOFTWARE ENGINEERING APPROACH**

## 3.1 SOFTWARE MODEL USED

### 3.1.1 Description

#### Incremental Model

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

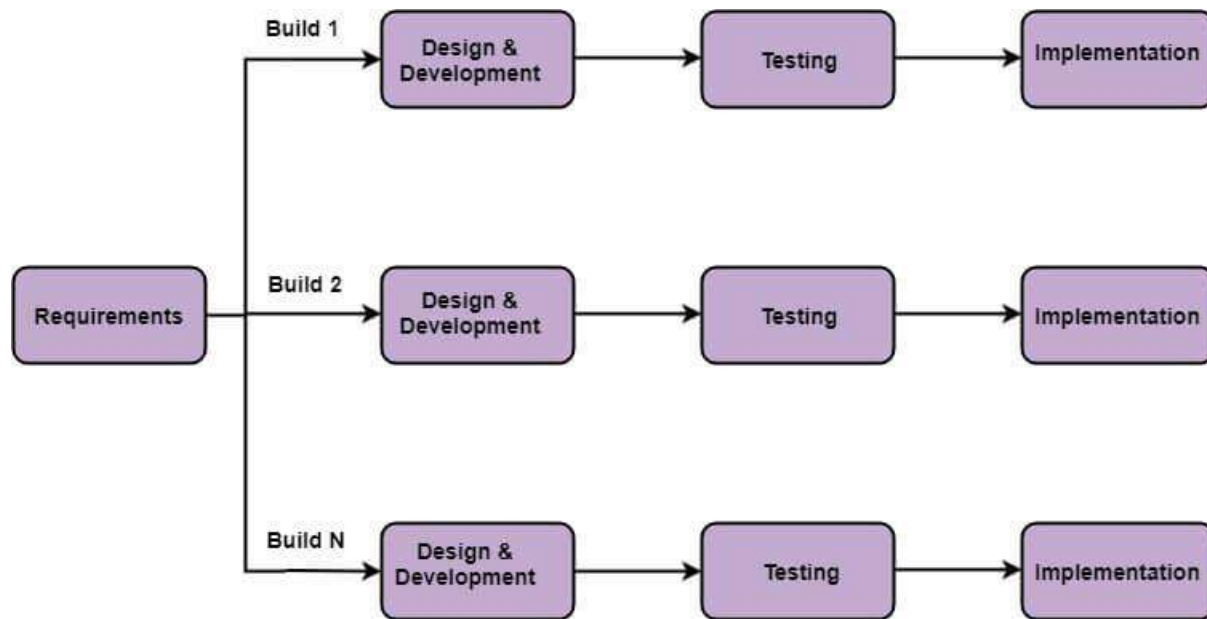


Fig: Incremental Model

Fig. 3.1 “Incremental Model”

The various phases of incremental model are as follows:

- 1. Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
- 2. Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

**3. Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

**4. Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

### **Advantages of Incremental Model**

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

### **Disadvantages of Incremental Model**

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.
- Well defined module interfaces are needed.

#### **3.1.2 Reasons for Use**

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.



## 3.2 PLATFORM SPECIFICATION

### 3.2.1 Hardware Specification

The hardware Requirement needed for implementing our project are as follows –

<b>Memory</b>	: 4 GB RAM, 256GB HDD/SDD
<b>Processor</b>	: Intel i3 or Equivalent
<b>Devices</b>	: Android/IOS Based smartphone, PC/Laptop

### 3.2.2 Software Requirement

The software Requirement needed for implementing our project are as follows-

<b>Operating System</b>	: Microsoft Windows (8 or higher)/MacOS/Android/IOS
<b>Programming Language</b>	: Python 3.8 +, JavaScript
<b>Frontend Design</b>	: HTML, CSS, Bootstrap, JQuery (Datatable plugin), chart.js
<b>Database</b>	: PostgresSql, Supabase
<b>Applications</b>	: Web Browser (Mozilla Firefox / Google Chrome), Telegram Application/ Telegram Desktop Application
<b>IDE/Tools</b>	: Pycharm IDE, DataGrip, Heroku

# **CHAPTER – 4**

## **DESIGNS**

## 4.1 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system the shows the relationship between the user and the different use cases in which user is involved. A use case diagram can identify the different types of a system and the use cases and will often be accompanied by other types of diagram as well.

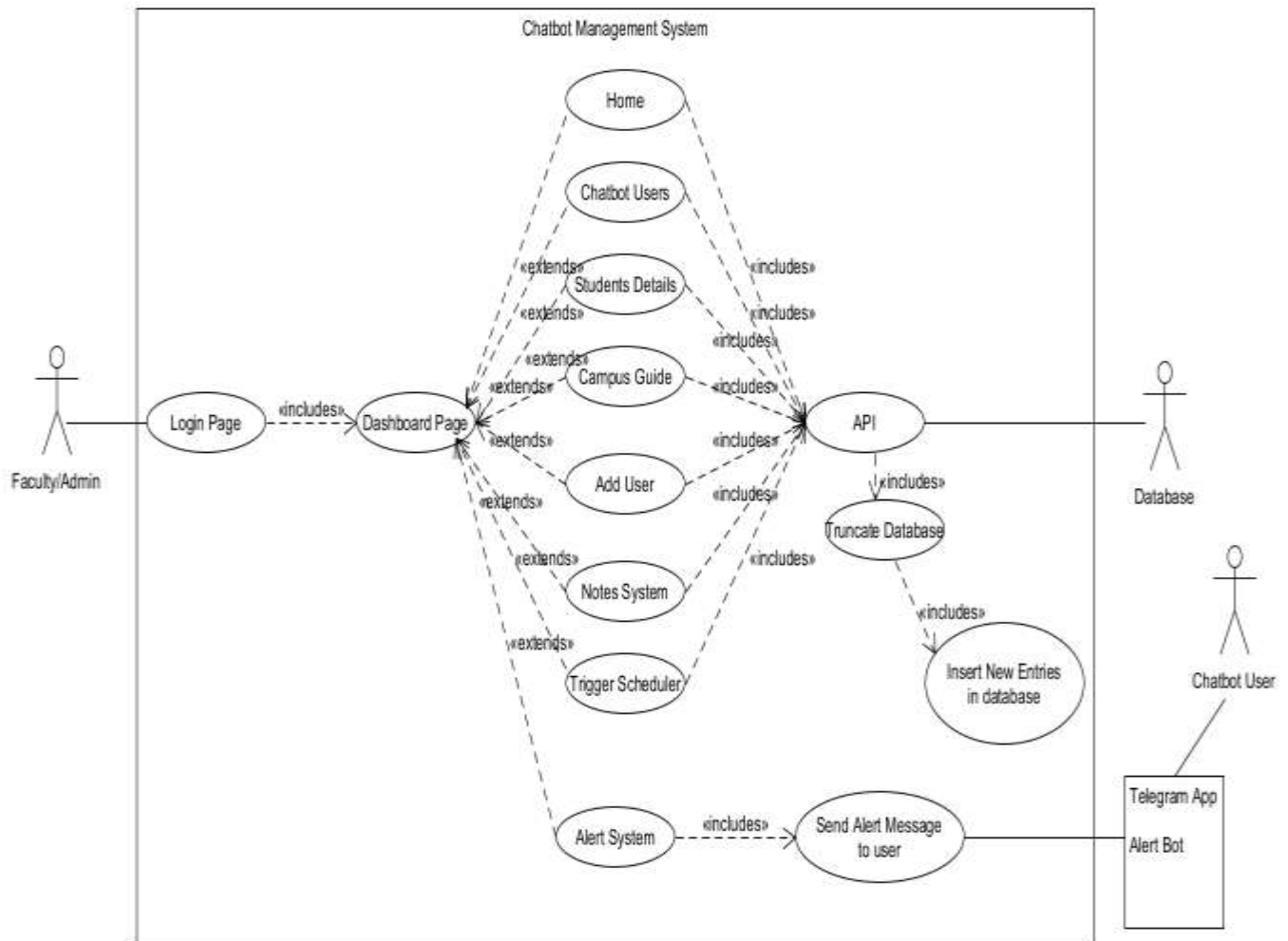


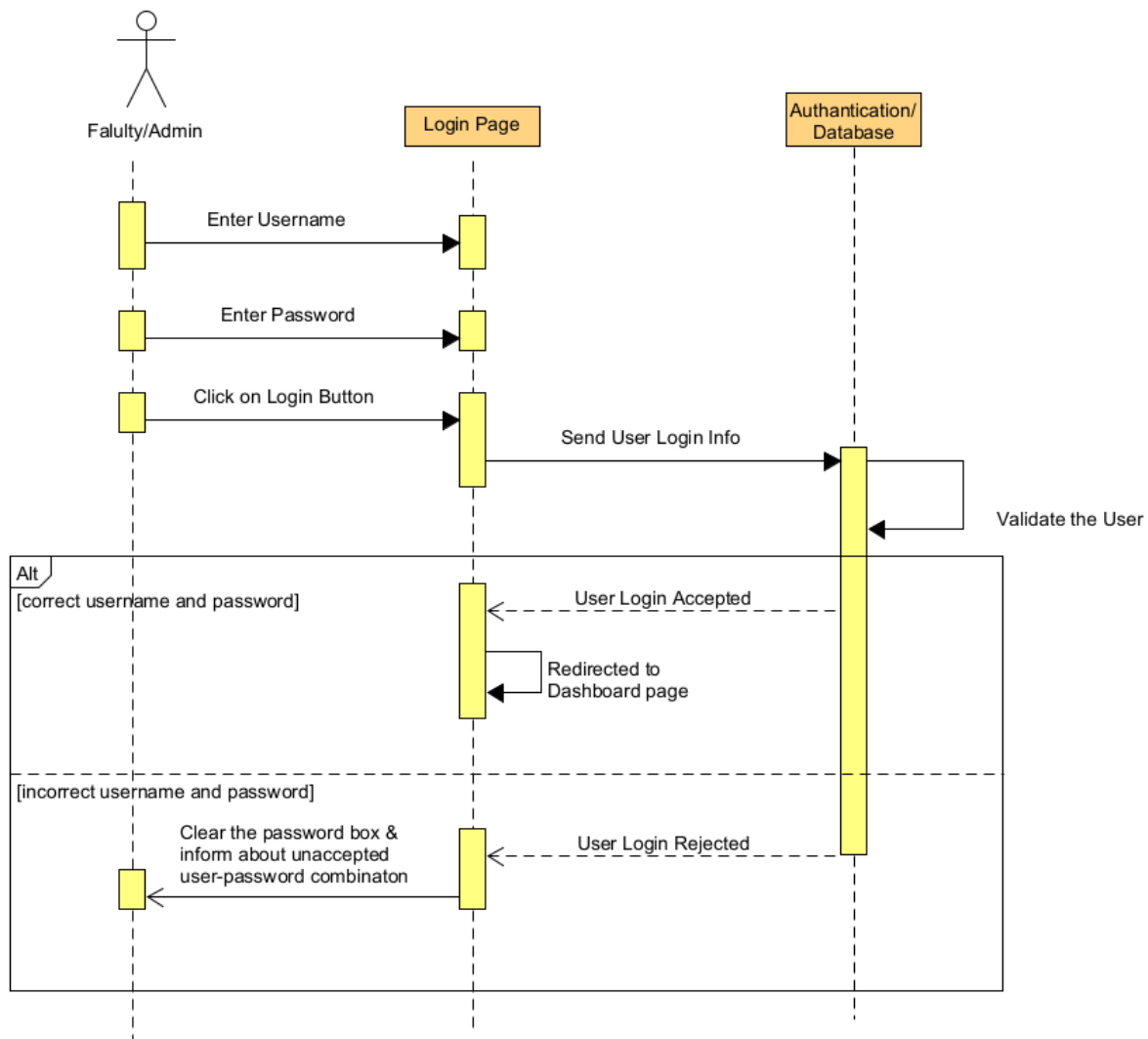
Fig 4.1 "Use Case Diagram"

## 4.2 SEQUENCE DIAGRAM

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. Sequence Diagrams captures:

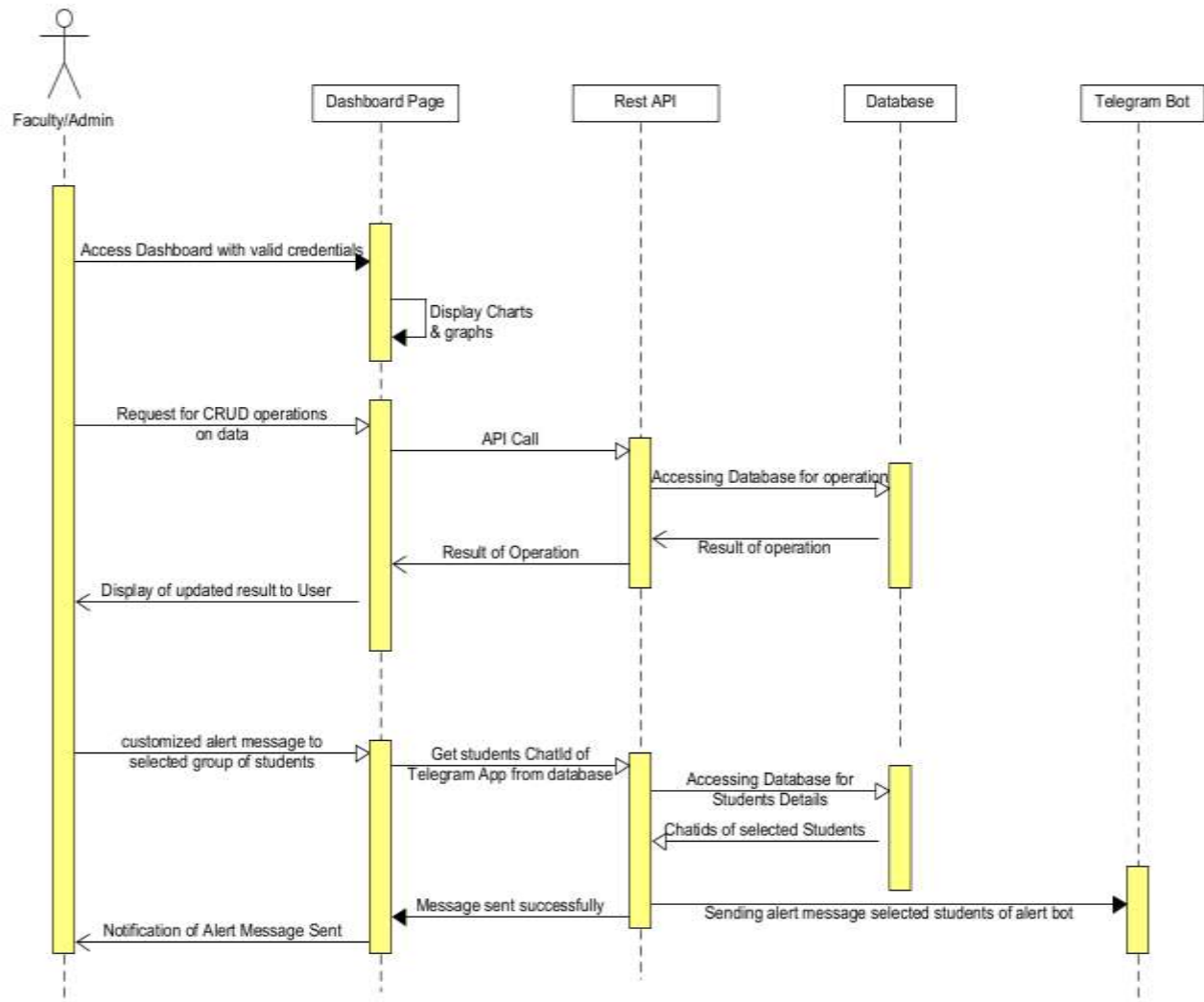
- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams).
- High-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams).

### 4.2.1 Sequence Diagram for User Authentication Module



**Fig. 4.2.1 “Sequence Diagram for user authentication”**

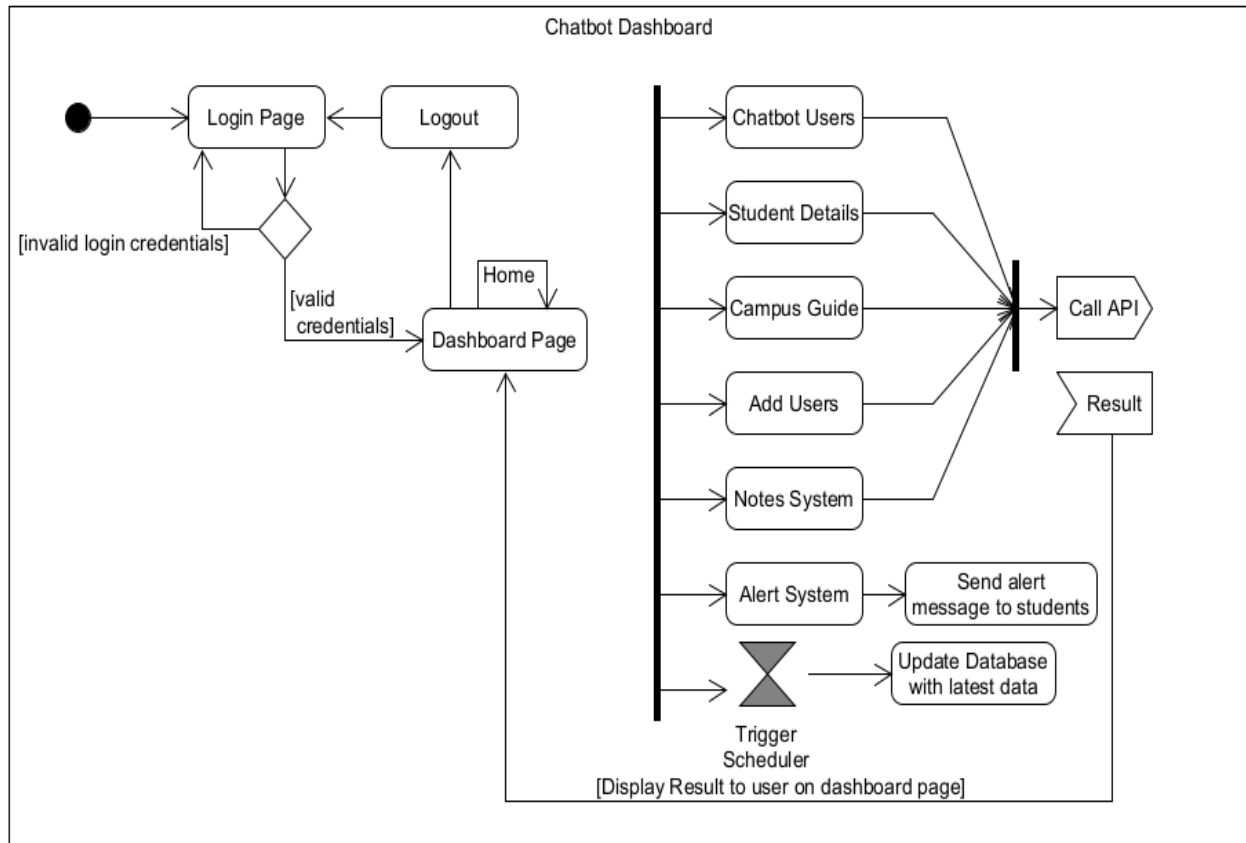
## 4.2.2 Sequence diagram for dashboard



**Fig 4.2.2 “Sequence diagram for dashboard”**

### 4.3 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



**Fig 4.3 “Activity Diagram”**

## 4.4 DATABASE DESIGN

### Database: ieschatbot\_db

Different tables in database are as follow-

**Schema Name** – user\_details\_schema

**Table Name** – users

**Primary Key** – chat\_id

Columns	Type	Constraints	Description
chat_id	varchar	Primary_key	Stores the chat_id of user
first_name	varchar	Not NULL	Stores the first name of user
last_name	varchar	No constraint	Stores the last name of user
created_at	timestamp	Not NULL	Stores time and date of first use of bot by user

**Table (4.1) – “Table structure for user”**

**Schema Name** – user\_details\_schema

**Table Name** – students

**Foreign Key** – chat\_id references users.chat\_id

Columns	Type	Constraints	Description
chat_id	varchar	Primary_key	Stores the chat_id of user
branch	varchar	Not NULL	Stores the branch of user
year	integer	Not NULL	Stores the year of user

**Table (4.2) – “Table structure for students”**

**Schema Name** – syllabus\_schema

**Table Name** – syllabus

**Primary Key** – branch + semester

Columns	Type	Constraints	Description
branch	varchar	Not NULL	Stores the branch for specified syllabus
semester	varchar	Not NULL	Stores the semester for specified syllabus
link	text	Not NULL	Stores the link from where user can download syllabus
scaaped_on	timestamp	Not NULL	Stores time and date of scrapping link for the syllabus
title	varchar	Not NULL	Store the title heading for the syllabus

**Table (4.3) – “Table structure for syllabus”**

**Schema Name** – rgpv\_notes\_schema

**Table Name** – rgpv\_notes

**Primary Key** – subject + unit\_no

Columns	Type	Constraints	Description
unit_no	Integer	Not NULL	Stores the unit number
subject	text	Not NULL	Stores the subject name
url	text	Not NULL	Stores the link from where user can download notes
abbr	text	Not NULL	Stores the abbreviations for subject name

**Table (4.4) – “Table structure for RGPV Notes”**



**Schema Name** – notice\_board\_schema

**Table Name** – notice\_board

**Primary Key** – notice\_title + notice\_link

Columns	Type	Constraints	Description
notice_title	text	Not NULL	Stores the title of notice
notice_link	text	Not NULL	Stores the link of notice
notice_type	text	Not NULL	Stores the type of notice
scrapped_on	timestamp	Not NULL	Stores the date & time of scrapping

**Table (4.5) – “Table structure for Notice Board”**

**Schema Name** – campus\_guide\_schema

**Table Name** – faculty\_details

**Primary Key** – first\_name + contact

Columns	Type	Constraints	Description
Title	text	Not NULL	Stores the title for teacher
first_name	text	Not NULL	Stores the first name of faculty
last_name	text	Not NULL	Stores the last name of faculty
department	text	Not NULL	Stores the department name of faculty
designation	text	Not NULL	Stores the designation name of faculty
contact	varchar	Not NULL	Stores the contact details of faculty
email	text	Not NULL	Stores the email address of faculty
cabin	text	Not NULL	Stores the cabin number of faculty

**Table (4.6) – “Table structure for Campus Guide”**

**Schema Name** – alert\_system\_schema

**Table Name** – users

**Primary Key** – chat\_id

Columns	Type	Constraints	Description
chat_id	int4	Primary Key	Stores the chat_id
first_name	varchar	No constraint	Stores the first name
last_name	varchar	No constraint	Stores the last name
subscribed_on	timestamp	Not NULL	Stores the time & data of first subscribed
user_name	varchar	No constraint	Stores the user name
branch	varchar	No constraint	Stores the branch of user
year	Int4	No constraint	Stores the year of user

**Table (4.7) – “Table structure for user”**

**Schema Name** – dashboard\_sch

**Table Name** – login

**Primary Key** – id

Columns	Type	Constraints	Description
id	Int4	Primary Key	Stores the id
username	varchar	No constraint	Stores the user name
password	varchar	No constraint	Stores the password
first_name	varchar	No constraint	Store the first name
last_name	varchar	No constraint	Stores the last name
designation	varchar	No constraint	Store the designation
image_url	varchar	No constraint	Stores the image url

**Table (4.8) – “Table structure for login”**

# **CHAPTER – 5**

## **IMPLEMENTATION PHASE**

## 5.1 LANGUAGE USED & ITS CHARACTERISTICS

### 5.1.1 Technology used for development

#### 1. Python:-

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

- **Why python for Development:** - Python is one of the best language for chatbot. The presence of AIML (Artificial Intelligence Markup Language) makes it easier to compose syntax. It also reduces the margin of errors for beginners and professionals. Python makes it easy for developers to define strategic roadmaps for the chatbots to take under different scenarios. It's used to make the conversational components hosted on the webserver.

#### 2. HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages.

#### 3. CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications.

#### 4. FLASK API

**Flask** is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.<sup>[2]</sup> It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

#### 5. JavaScript Library (Chart.js)

**Chart.js** is a free, open-source JavaScript library for data visualization, which supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. Created by London-based web developer Nick Downie in 2013, now it is maintained by the community and is the second most popular JavaScript charting library on GitHub by the number of stars after D3.js,

considered significantly easier to use though less customizable than the latter. Chart.js renders in HTML5 canvas and is widely covered as one of the best data visualization libraries. It is available under the MIT license.

### 5.1.2 Technology used for database

**Database:** Database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

#### **PostgreSQL**

PostgreSQL is an enterprise-class open source database management system. It supports both SQL and JSON for relational and non-relational queries for extensibility and SQL compliance. PostgreSQL supports advanced data types and performance optimization features, which are only available in expensive commercial databases, like Oracle and SQL Server. It is also known as Postgres.

#### **SUPABASE:**

Supabase is a hosted platform. You may sign up and begin using Supabase right away, without having to install anything on your server. You can also locally host and develop anything.

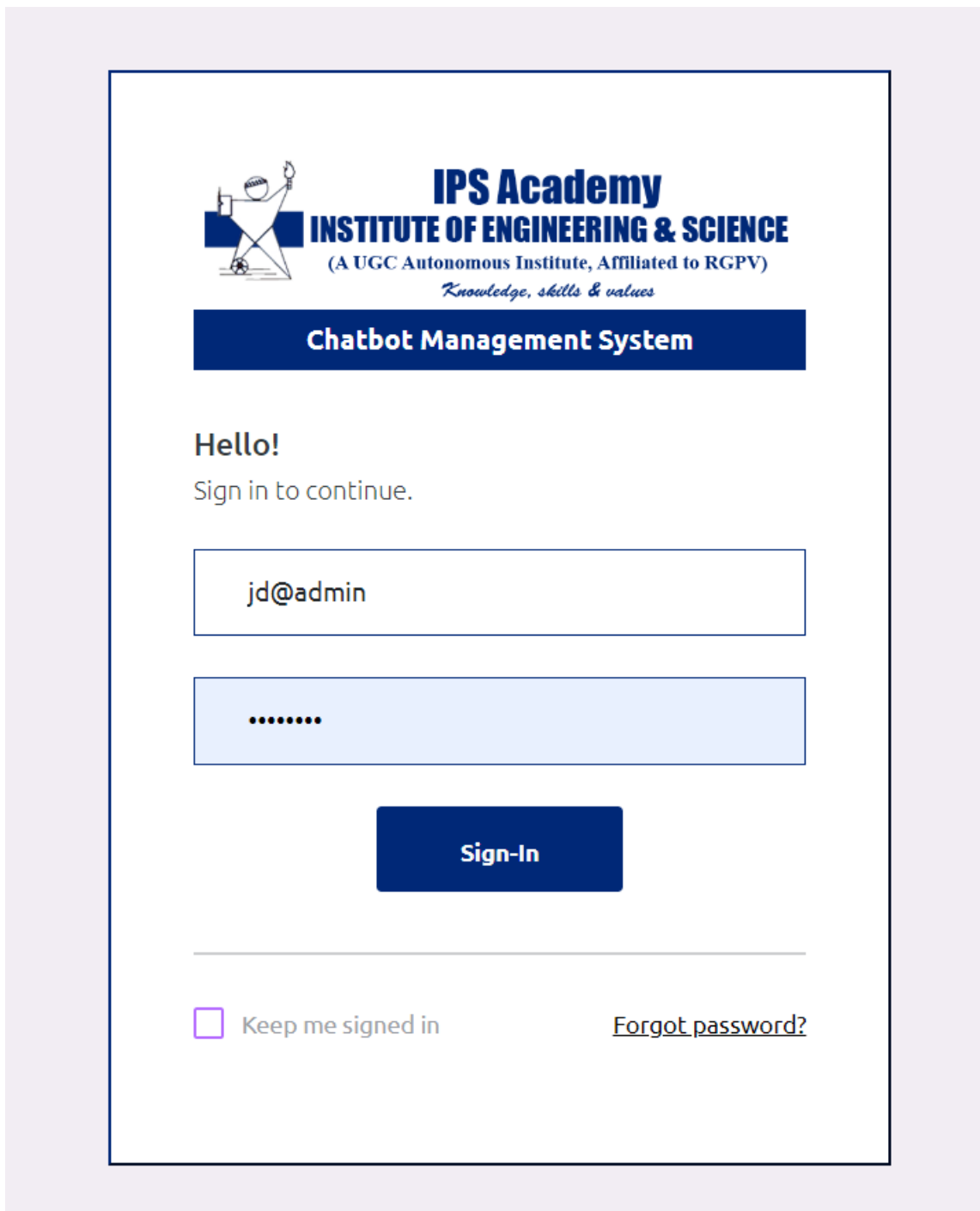
They divided each library into small pieces that may be used separately. Each sub-library is a standalone implementation of one external system. This is one of the ways they keep current technologies relevant.

It contains all of the backend features you'll need to build a product. You may use all the tools or just the ones you need the most, such as:

- **Database:** A dedicated, scalable Postgres database.
- **Auth:** User management with Row Level Security.
- **File Storage:** Store, organize, and serve large files.
- **Auto-generated APIs:** Instantly generate APIs for your database

## 5.2 GUI (SNAPSHOTS)

### 1. Login Page



The screenshot displays the login interface for the Chatbot Management System of IPS Academy. At the top, the academy's logo and name are shown, followed by its affiliation and motto. Below this, a dark blue header bar contains the text 'Chatbot Management System'. The main content area begins with a 'Hello!' greeting and a prompt to 'Sign in to continue.'. Two input fields are provided: one for the email address, which contains 'jd@admin', and another for the password, represented by dots. A prominent dark blue 'Sign-In' button is centered below the password field. At the bottom, there is a horizontal line separating the login section from the footer. The footer includes a checkbox labeled 'Keep me signed in' and a link for 'Forgot password?'.

**IPS Academy**  
**INSTITUTE OF ENGINEERING & SCIENCE**  
(A UGC Autonomous Institute, Affiliated to RGPV)  
*Knowledge, skills & values*

**Chatbot Management System**

**Hello!**  
Sign in to continue.

jd@admin

.....

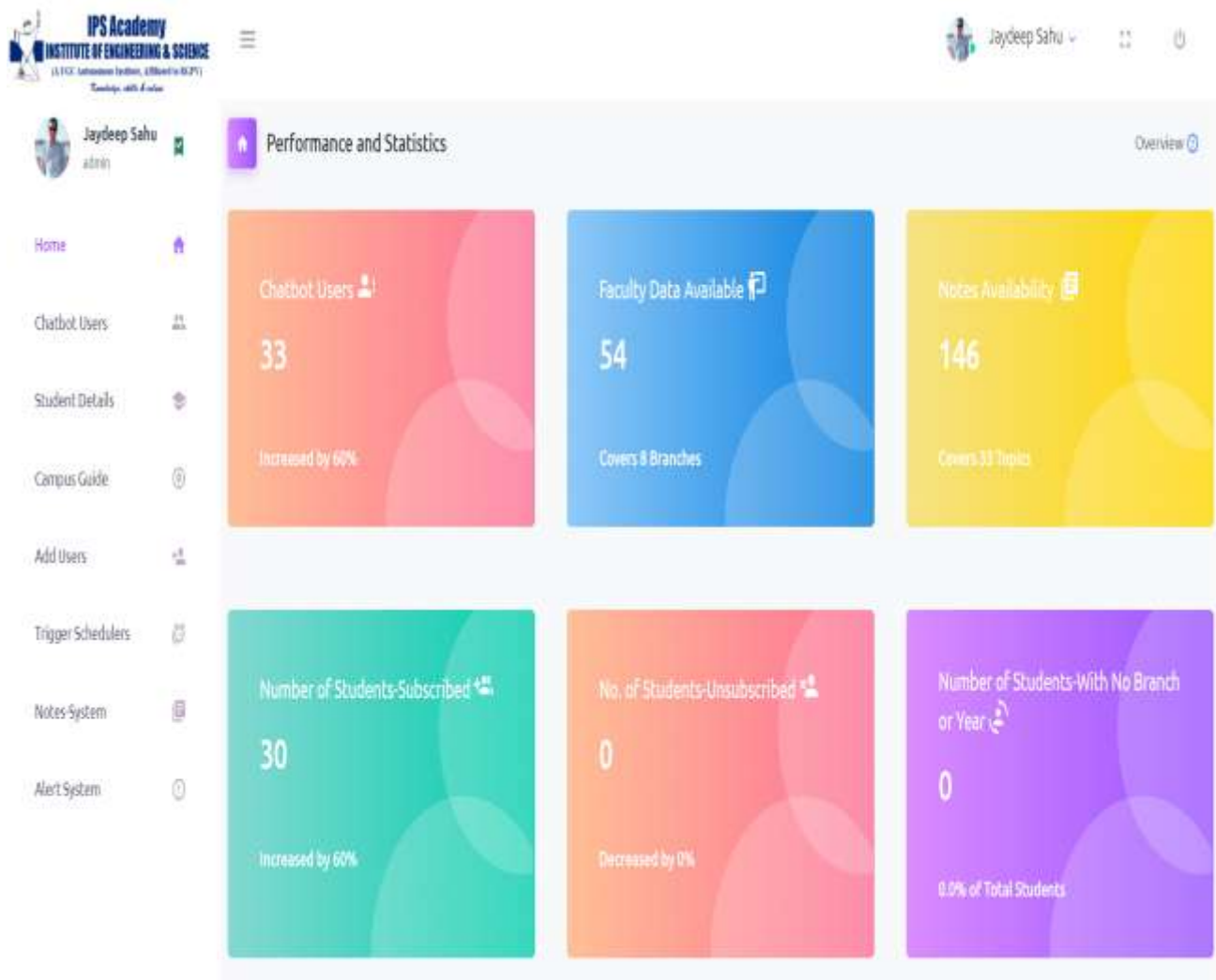
**Sign-In**

---

☐ Keep me signed in      [Forgot password?](#)

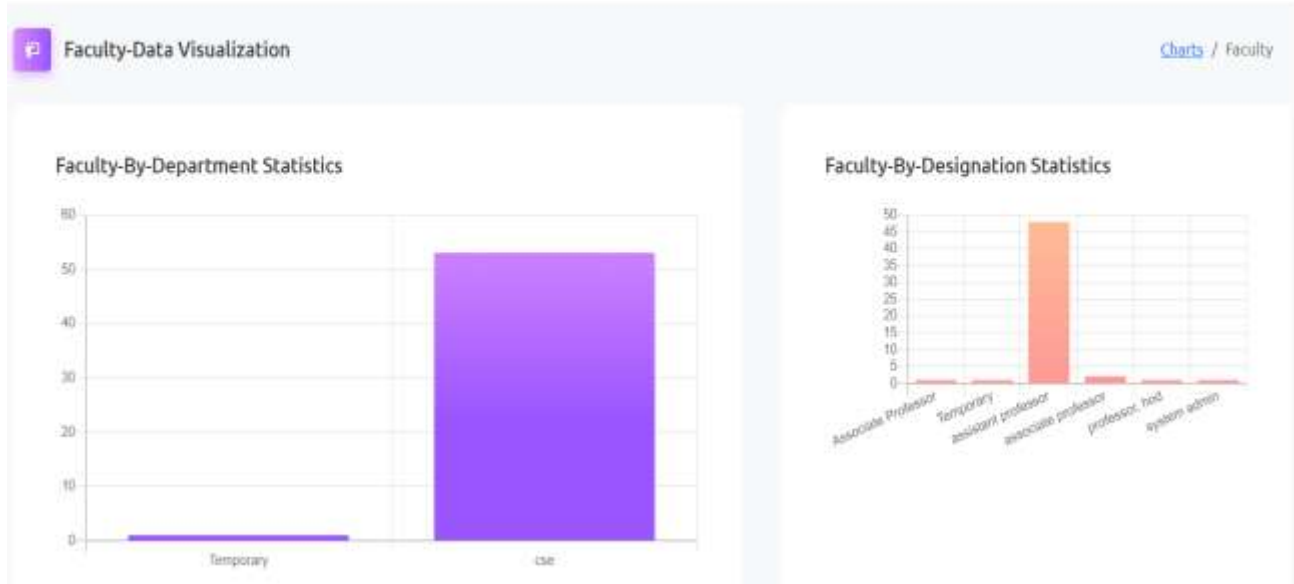
Fig 5.1 “Login Page”

## 2. Home Page



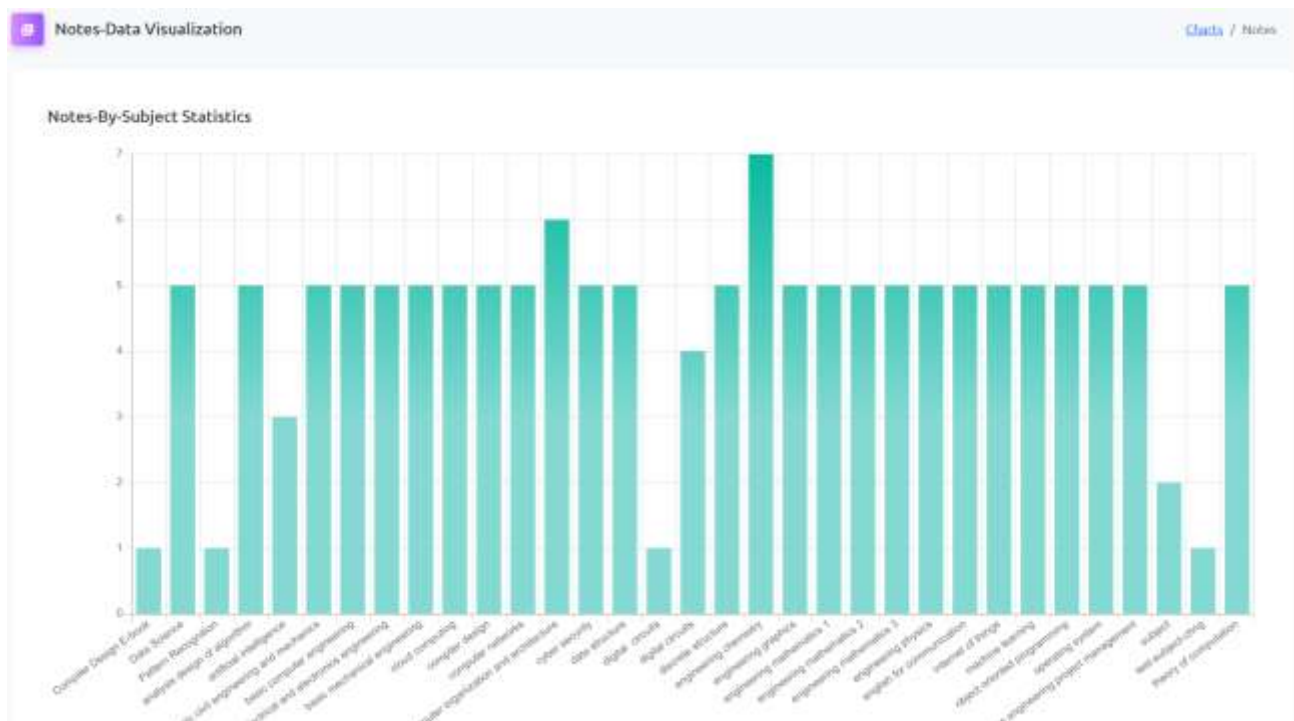
**Fig 5.2 “Home Page”**

### 3.1 Faculty Data Visualization



### Fig 5.3 “Faculty Data Visualization”

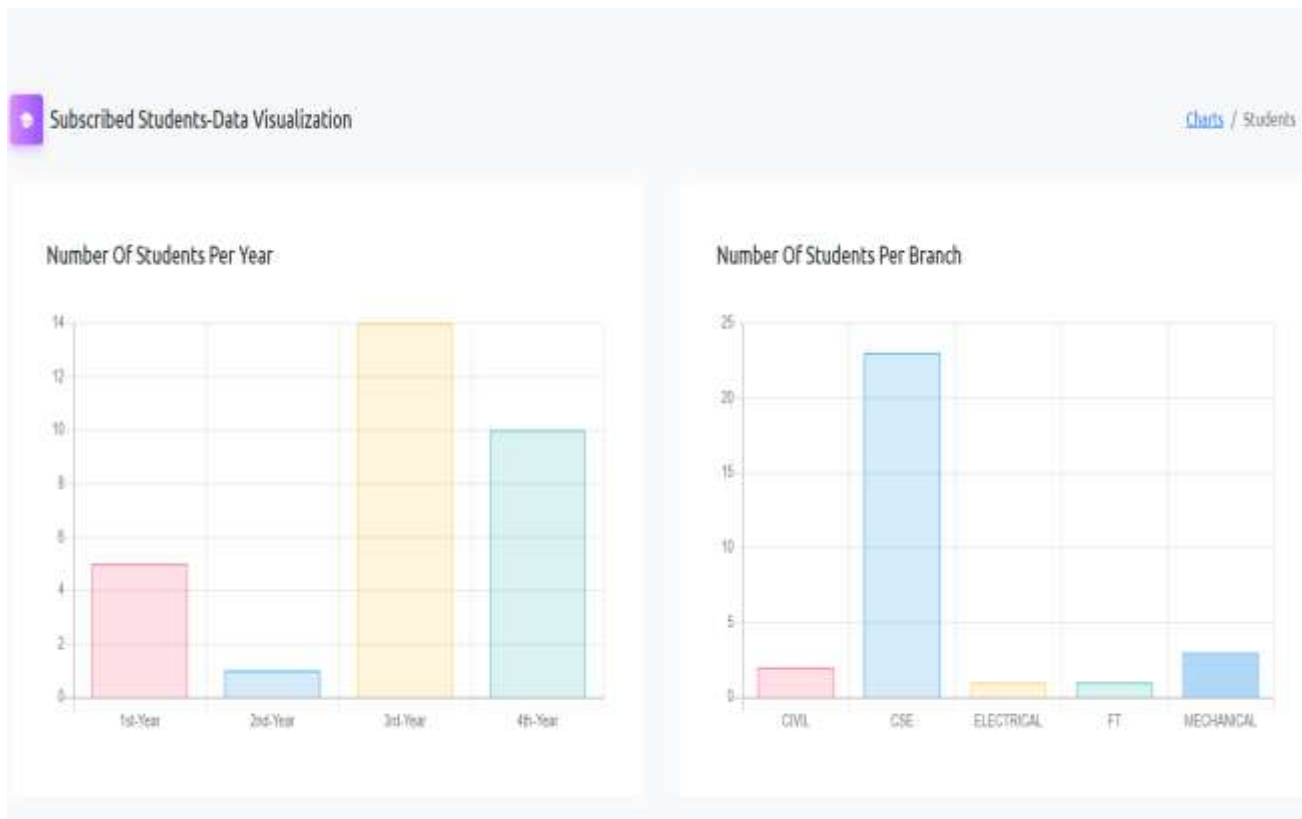
### 3.2 Notes data Visualization



### Fig 5.4 “Notes Data Visualization”



### 3.3 Subscribed Students – Data Visualization



**Fig 5.5 “Subscribed Students- Data Visualization”**

## 4. Modules

### 4.1 Students Details

No.	Chat-ID	First Name	Last Name	Branch	Year	Subscribed On (UTC)
1	1802958203	Nayan	Praspat	MECHANICAL	3	2022-04-11 15:26:52.77216
2	830710314	Aryan	Pegwar	MECHANICAL	3	2022-06-22 11:04:01.566727
3	1308853626	Narendra	Patidar	ELECTRICAL	4	2022-04-11 15:53:28.089875
4	849787669	KULDEEP	SAHU	CSE	4	2022-07-12 15:51:43.234213
5	746820016	Parth	Wanik	CSE	3	2022-04-12 15:58:13.154286
6	1211843654	Harsh	Dhakad	CSE	1	2022-04-16 14:21:40.780878
7	978592976	Akshat	Udeeniya	CSE	3	2022-04-17 09:58:06.262312
8	1228746979	S	M	Civil	3	2022-04-17 13:38:30.744651
9	949543474	Vedant	None	MECHANICAL	3	2022-04-22 06:31:24.680592
10	1843819190	Nikhil	Rathod	CSE	3	2022-04-25 05:55:34.954449

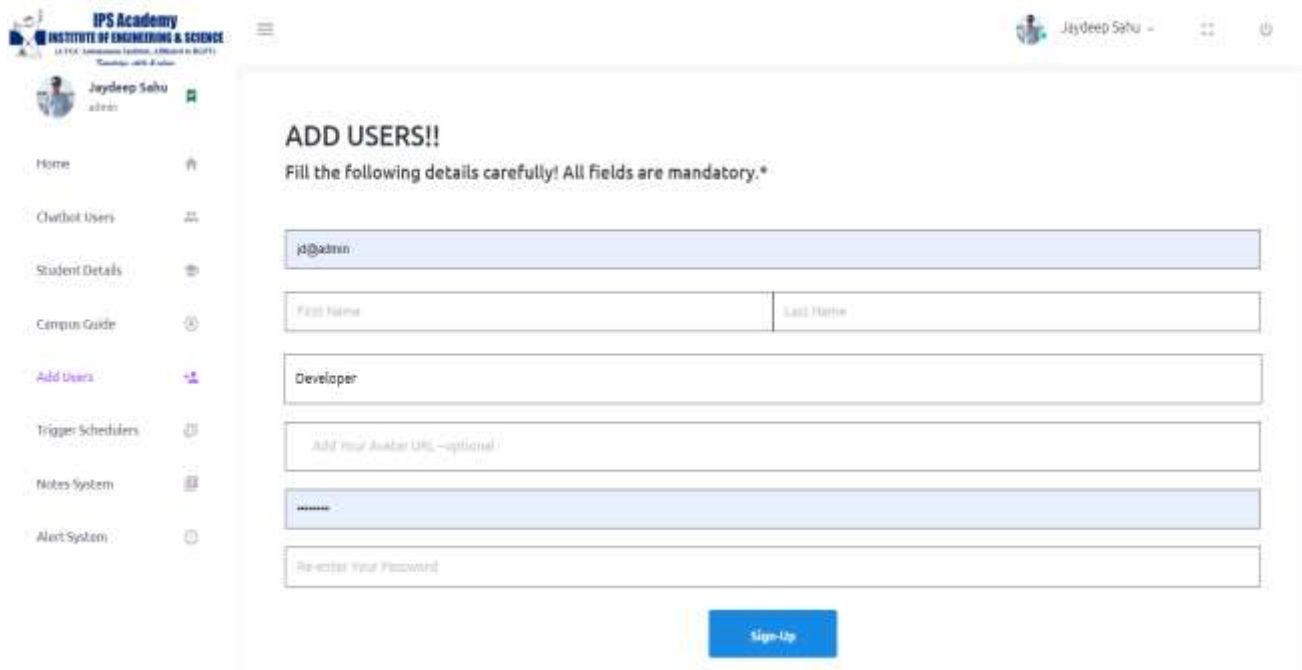
Fig 5.6 “Students Details”

### 4.2 Campus Guide

No.	Title	First Name	Last Name	Department	Designation	Contact	Email	Cabin	Action
1	Dr.	Arvind	Upadhyay	Cse	Associate Professor	91813556635	Abc@ipsacademy.Org	Department Of Cse, 2nd Floor	<a href="#">EDIT</a> <a href="#">DELETE</a>
2	Mr.	Neeraj	Shrivastha	Cse	Associate Professor	91813556636	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
3	Mr.	Sunil	Nimawat	Cse	Assistant Professor	91813556637	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
4	Mr.	Yagyopal	Yadav	Cse	Assistant Professor	91813556660	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
5	Mrs.	Barkha	Sahu	Cse	Assistant Professor	91813556661	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
6	Mrs.	Sourabh	Jain	Cse	Assistant Professor	91813556662	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
7	Mrs.	Ved	Verma	Cse	Assistant Professor	91813556663	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
8	Mrs.	Anjali	Gupta	Cse	Assistant Professor	91813556664	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
9	Mrs.	Vaishali	Verma	Cse	Assistant Professor	91813556665	Abc@ipsacademy.Org	No	<a href="#">EDIT</a> <a href="#">DELETE</a>
10	Mrs.	Sudhar	Patidar	Cse	Assistant Professor	91813556667	Abc@ipsacademy.Org	207	<a href="#">EDIT</a> <a href="#">DELETE</a>

Fig 5.7 “Campus Guide”

## 4.3 Add Users



The screenshot shows the 'Add Users!!' form in the IPS Academy system. The left sidebar contains a menu with options: Home, Chatbot Users, Student Details, Campus Guide, Add Users (highlighted), Trigger Schedulers, Notes System, and Alert System. The main content area has a header 'ADD USERS!!' and a note 'Fill the following details carefully! All fields are mandatory.\*'. The form fields are: Email (j@admin), First Name, Last Name, Role (Developer), Add Your Avatar URL (optional), Password, and Re-enter Your Password. A 'Sign-Up' button is at the bottom.

**ADD USERS!!**  
Fill the following details carefully! All fields are mandatory.\*

Email: j@admin

First Name: Last Name:

Role: Developer

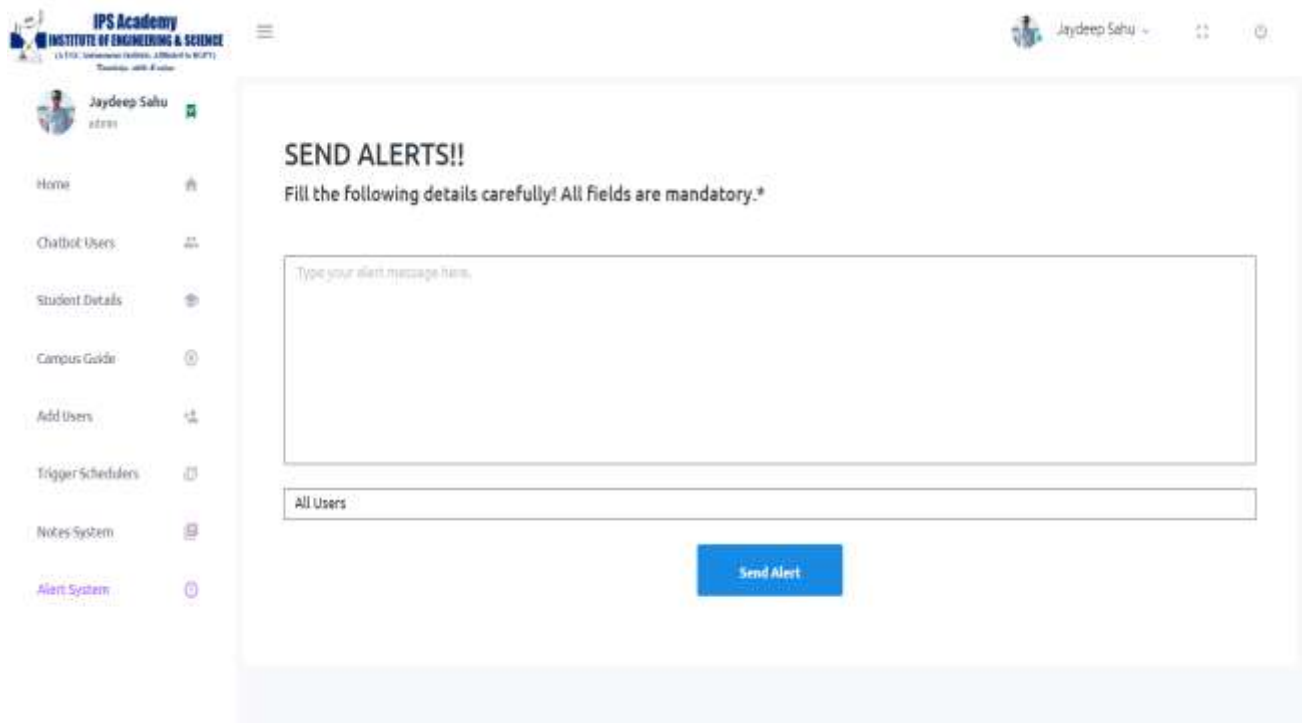
Add Your Avatar URL (optional):

Password: Re-enter Your Password:

**Sign-Up**

Fig 5.8 “Add Users”

## 4.4 Alert System



The screenshot shows the 'Alert System' form in the IPS Academy system. The left sidebar contains a menu with options: Home, Chatbot Users, Student Details, Campus Guide, Add Users, Trigger Schedulers, Notes System, and Alert System (highlighted). The main content area has a header 'SEND ALERTS!!' and a note 'Fill the following details carefully! All fields are mandatory.\*'. The form fields are: Type your alert message here (a large text area) and a dropdown menu for 'All Users'. A 'Send Alert' button is at the bottom.

**SEND ALERTS!!**  
Fill the following details carefully! All fields are mandatory.\*

Type your alert message here:

All Users

**Send Alert**

Fig 5.9 “Alert System”

#### 4.5 Campus Guide “Add Faculty Details”

Add Faculty Detail in Campus-Guide

Title

Mr.

Name

First Name

Last Name

Contact Details

+91

Email Details

Cabin Details

Room No. / Lab No. / Floor / Block

Designation

Head of Department

Department

CSE

Close

Save changes

**Fig 5.10 “Campus Guide- Add Faculty Details”**

## 4.6 Campus Guide “Edit Faculty Details”

Edit Faculty Detail in Campus-Guide

Updated Title

New Name

anvird

upadhyay

New Contact Details

+91

91813556655

New Email Details

anv@psacademy.org

New Cabin Details

department of cse, 2nd floor

New Designation

New Department

CSE

Close

Save changes

**Fig 5.11 “Campus Guide – Edit Faculty Details”**

# **CHAPTER – 6**

## **TESTING METHODS**

## 6.1 TESTING :-

Testing is a process of executing a program with the intent of finding an error and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

### 6.1.1 Testing Objectives: -

The objectives of testing are:

- Testing is done with intent of finding an error.
- Testing is done in order to check the various control flow of system.
- The scope of testing is the extensiveness of the test process

## 6.2 Test Cases

### Test 01 – Test Case “Login Page”

Case No.	Action	Expected Result	Success	Comment
01	Press sign-in button without filling username & password	User must be asked to fill the details.	Yes	Action Successfully Tested.
02	Press sign-in button with incorrect username and incorrect password	Sign-in Fail. User must be asked to fill correct details.	Yes	Action Successfully Tested.
03	Press sign-in button with incorrect username but correct password	Sign-in Fail. User must be asked to fill correct details.	Yes	Action Successfully Tested.
04	Press sign-in button with correct username & incorrect password	Sign-in Fail. User must be asked to fill correct details.	Yes	Action Successfully Tested.
05	Press sign-in button with correct username and correct password	Sign-in Success. User directed to home page.	Yes	Action Successfully Tested.

**Table (6.1) – “Test Case – Login Page”**

**Test 02 – Test Case “Home Page”**

<b>Case No.</b>	<b>Action</b>	<b>Expected Result</b>	<b>Success</b>	<b>Comment</b>
01	Data fetch related to number of Chatbot users, notes, faculty data etc.	Must show fetched data on the home page in performance and statistics data.	Yes	Action Successfully Tested.
02	Faculty Data Visualization	Must show graph of Faculty-By-Department Statistics and Faculty-By-Designation Statistics using chart.js library of JavaScript & based on data fetched using API “Charts/faculty”	Yes	Action Successfully Tested.
03	Notes Data Visualization	Must show graph of Notes-By-Subject Statistics using chart.js library of JavaScript & based on data fetched using API “Charts/Notes”	Yes	Action Successfully Tested.
04	Subscribed students data visualization	Must show graph of students data year wise and branch wise using chart.js library of JavaScript & based on data fetched using API “Charts/Students”	Yes	Action Successfully Tested.

**Table (6.2) – “Test Case – Home Page”**



**Test 03 – Test Case “Chatbot User Module”**

<b>Case No.</b>	<b>Action</b>	<b>Expected Result</b>	<b>Success</b>	<b>Comment</b>
01	Click on “Chatbot User” from sidebar on home page.	User must be directed to “user” page and table related to Chatbot user data is shown on page with eye pleasant GUI to perform actions and maintain details.	Yes	Action Successfully Tested.
02	User click on “Delete User” button to delete user data form database	User data must be deleted from database.	Yes	Action Successfully Tested.
03	User type name in “Search bar” to search for specific name	All entries related to user of that specific name only must be shown in table.	Yes	Action Successfully Tested.

**Table (6.3) – “Test Case –Chatbot User Module”****Test 04 – Test Case “Students Detail Module”**

<b>Case No.</b>	<b>Action</b>	<b>Expected Result</b>	<b>Success</b>	<b>Comment</b>
01	Click on “Students Detail” from sidebar on home page.	User must be directed to “students” page and table related to Students detail is shown on page with eye pleasant GUI to perform actions and maintain details.	Yes	Action Successfully Tested.
02	User type name in “Search bar” to search for specific name	All entries related to user of that specific name only must be shown in table.	Yes	Action Successfully Tested.

**Table (6.4) – “Test Case – Students Details Module”**

### Test 05 – Test Case “Campus Guide Module”

Case No.	Action	Expected Result	Success	Comment
01	Click on “Campus Guide” from sidebar on home page.	User must be directed to “campus_guide” and table related to Faculties is shown on page with eye pleasant GUI to perform actions and maintain details.	Yes	Action Successfully Tested.
02	User click on “Delete” button to delete user data from database	User data must be deleted from database.	Yes	Action Successfully Tested.
03	User type name in “Search bar” to search for specific name	All entries related to user of that specific name only must be shown in table.	Yes	Action Successfully Tested.
04	User click on “Edit” button to edit details of specific faculty.	Form asking for new details of faculty must be open and after filling details, it must be updated and changes commit in database.	Yes	Action Successfully Tested.
05	User click on “Add Faculty Details” button to add new faculty in database.	Form asking for various details of faculty must be open and after filling details, it must be added to database.	Yes	Action Successfully Tested.

**Table (6.5) – “Test Case –Campus Guide Module”**

**Test 06 – Test Case “Add Users Module”**

<b>Case No.</b>	<b>Action</b>	<b>Expected Result</b>	<b>Success</b>	<b>Comment</b>
01	Click on “Add Users” from sidebar on home page.	User must be directed to “add_users” page, and form asking details of user.	Yes	Action Successfully Tested.
02	User click on “sign up” button and user with filled details is added, and using that credentials, new user can login	User with filled details is added, and using that credentials, new user can login to their account.	Yes	Action Successfully Tested.

**Table (6.6) – “Test Case – Add User Module”****Test 07 – Test Case “Alert System Module”**

<b>Case No.</b>	<b>Action</b>	<b>Expected Result</b>	<b>Success</b>	<b>Comment</b>
01	Click on “Alert System” from sidebar on home page.	User must be directed to “alerts” page.	Yes	Action Successfully Tested.
02	User click on “sent alert” button after select target users and type alert message in textarea.	All targeted audience, which are selected from list given, receive a “alert message” on telegram app from alert bot.	Yes	Action Successfully Tested.

**Table (6.7) – “Test Case – Alert System Module”**

# **CHAPTER – 5**

## **CONCLUSION**

## 5.1 CONCLUSION

- The major project will not only help us manage the chatbot, by creating a stand alone "College Information Management System with Chatbot", but also help the in understanding and managing data with the help of data visualization using various graphs and charts.
- Features like alert system will help the students get personalized alerts of the information such as notice board, new assignments, new feature updates etc.
- Using Web Application, anyone can access and manage the chatbot data with eye pleasing UI, with no prior technical knowledge required.
- Major project aims to provides an easy and effective way for faculties to provide the notes, assignment to their students.
- College Management System aims to provides the layer of abstraction with which we can update, read, manage and visualize data without directly interacting with database.
- The project will inbuilt with user authentication system, using this feature only a authorized person with valid credential can manipulate the data.

# **CHAPTER- 6**

## **LIMITATIONS & FUTURE ENHANCEMENTS**

## 6.1 LIMITATIONS

- **Performance**

A web app is directly linked to a web browser. Due to this, the app size tends to get increased. The impact of this can be seen in the performance of a web application. A large web app performs considerably slower than a native desktop app.

- **Internet Dependence**

An internet connection is compulsory when running a web application. Still there are many parts of the world where internet is not accessible. Without a reliable internet connection you cannot either browse the web or run the web application.

- **Reduced Speed**

Often a web app operates relatively at a slower speed than the application hosted on a local server, and for those reasons can't completely replace mobile apps. It is also directly linked to our browser, because of which its app size tends to increase. A large app, therefore, is considerably slower than a native desktop one. Plus, as a web app runs completely on the internet, it frequently can feel slower due to the internet connection quality.

- **Deployed on Heroku**

Application is deployed on heroku and heroku is going to stop the free tier from Dec 2022 so in future we might have to search for its alternative that is cheaper and effective.

## 6.2 FUTURE ENHANCEMENTS

- **More data visualization features**

In our current dashboard, many graphs and charts are provided data visualizations which can be used for easy data analysis, but more data visualizations can be provided.

- **Targeted Alerts**

Enabling faculty member to send targeted alerts to particular branch and particular year students. Also providing facility of sending message to specific group of student at one click.

- **Question Paper**

**Adding an API** (Application programming interface) and user interface to add previous year question paper.

- **Scheduler**

Enable schedulers to run through controls provided on Web Interface to perform web scrapping related operations

- **Campus Guide Enhancement**

Collecting more relevant data of All the faculty members so that Chatbot user can get more accurate details about faculty members. Collecting details of faculty from another department is also a primary goal.

- **Query Resolution**

Adding interface in chatbot to send their queries/ review/feedbacks to admin and hence enabling the web interface to View those queries and resolve them in efficient manner



## **REFERENCES**

## REFERENCES

### Online Sources

- [1] <https://fastapi.tiangolo.com/>
- [2] <https://www.toptal.com/python/build-high-performing-apps-with-the-pythonfastapi-framework>
- [3] <https://tobiasahlin.com/blog/introduction-to-chartjs/>
- [4] <https://www.chatbot.com/help/overview/meet-your-dashboard/>
- [5] <https://docs.pyrogram.org/>
- [6] <https://www.chartjs.org/docs/latest/>
- [7] <https://botbot.ai/product/data-analytics-dashboard/>

### Research Papers

- [1] Gangrade, J., Surme, S. S., Somu, S., Raskonda, S., Gupta, P. (2019). A review on college enquiry chatbot. *International Journal of Engineering Science and Computing*, 9(3).
- [2] J. Bang, H. Noh, Y. Kim and G. G. Lee, "Example-based chat-oriented dialogue system with personalized long-term memory," 2015 International Conference on Big Data and Smart Computing (BIGCOMP), Jeju, 2015.