**VIDYA JYOTHI INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

**Aziz Nagar, Hyderabad -500075**

A Project Report

on

**MY COLLEGE FINDER**

Submitted for partial fulfilment of the requirements for the award of the degree

of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**

**BY**

**KANUMURI BHANU TEJA (16911A0584)**

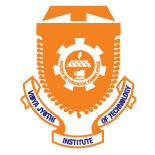
Under the guidance of

**MUNGA KAVYA**

Asst.Prof

Department of Computer Science Engineering

VJIT, Hyderabad.

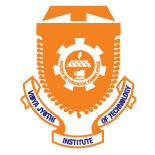


**Department of Computer Science Engineering**

**VIDYA JYOTHI INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

**Himayathnagar (vi), C.B. Post, R.R Dist. 500075**



**CERTIFICATE**

This is to certify that the project work entitled “**MY COLLEGE FINDER”** is a bonafide work carried out by **KANUMURI BHANU TEJA (16911A0584)**  in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE ENGINEERING to be awarded by** the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY**, Hyderabad.

The content in this report has not been submitted to any other university or institute for the award of any degree or diploma.

Internal Supervisor Head of the Department

Department of Computer Science Engineering Department of Computer Science Engineering

Hyderabad Hyderabad

**External Examiner**

**DECLARATION**

This is to certify that the work reported in the present project entitled as “**MY COLLEGE FINDER**” is a record of work done by me in the Department of Computer Science Engineering, Vidya Jyothi Institute of Technology(Autonomous), Jawaharlal Nehru Technological University, Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source.

**KANUMURI BHANU TEJA (16911A0584)**

**ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude and indebtedness to my project supervisor Munga Kavya for her valuable suggestions and interest throughout the course of this project.

I am also thankful to Head of the Department Computer Science for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

I convey my heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, I would like to take this opportunity to thank our family for their support through the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**ABSTRACT**

In this application, students can find various colleges that they want based on their fields of interest. This system is developed for students looking out for admission to the best colleges with their own fields of interest. It is very important to know about college into which we are getting through. In any field after higher secondary education/under-graduation, there is an entrance examination for getting admission to graduation college. It may be for any department in the engineering field. It happens that for searching the colleges the students have to search on the internet extensively. My college Finder is an application that will help these students to get the most appropriate college for them in whatever field they are aiming to develop their careers. In this application, the system will provide a list of colleges to students from which they can select their required college based on choices like hostel facility, locations, etc. On submitting the form the system sends the result along with the college's details like the number of departments/courses available and its location.

**INDEX**

**CONTENTS PAGE NO.**

Abstract iv

Index v

List of figures vi

List of screens vii

**CHAPTER-1: INTRODUCTION**

* 1. MOTIVATION
  2. PROBLEM DEFINITION
  3. LIMITATIONS OF PROJECT
  4. OBJECTIVE OF PROJECT

**CHAPTER-2: LITERATURE SURVEY**

1. INTRODUCTION
2. EXISTING SYSTEM
3. PROPOSED SYSTEM

**CHAPTER-3: ANALYSIS**

* 1. REQUIREMENT SPECIFICATION
     1. FUNCTIONAL REQUIREMENTS
     2. NON-FUNCTIONAL REQUIREMENTS
  2. SOFTWARE REQUIREMENTS

**CHAPTER-4: SOFTWARE DESIGN**

1. INSTALLATION OF XAAMP
   1. INSTALLATION SCREEN
   2. CHOOSE INSTALLATION OPTIONS
   3. CONTINUE INSTALLATION
   4. INSTALLATION FINISHED
   5. XAAMP CONTROL
   6. XAAMP SOFTWARE TOOLS
   7. APACHE SERIAL PORT
   8. MY\_SQL SERIAL PORT

**CHAPTER-5: TECHNOLOGY USED**

1. INTRODUCTION
2. EXPLANATION AND OUTPUT DESIGN
3. SOFTWARE ARCHETECTURE
4. UML DIAGRAMS
5. CONCLUSION

**CHAPTER-6: IMPLEMENTATION AND RESULT**

1. INTRODUCTION
2. EXPLANATION OF KEY FUNCTIONS
3. SAMPLE CODE
4. SCREENSHOTS

**CHAPTER-7: CONCLUSION**

**CHAPTER-8: FUTURE ENHANCEMENT**

**CHAPTER-9:**

**LIST OF FIGURES PAGENO.**

Figure 1.1.1: Arduino Board Diagram 1

Figure 1.4.1: Pin Diagram 3

Figure 3.1.1: DC Motor 16

Figure 4.2.1: Embedded System Architecture 23

Figure 4.2.2: Building Block Diagram 24

Figure 5.1.1: GSM/GPRS Module 27

Figure 5.2.1: Application of Wireless Modem 28

Figure 8.2.1: Use Case Diagram 56

Figure 8.2.2: Sequence Diagram 57

Figure 8.2.3: State Chart Diagram58

Figure 8.2.6: Collaboration Diagram 61

Figure 8.2.7: Class Diagram 62

Figure 8.2.8: Deployment Diagram 63

Figure 8.3 : Dataflow Diagram 64

**CHAPTER 1**

**INTRODUCTION**

1. **MOTIVATION**

In this project, aiming at manually finding the colleges based on different privileges/fields of interest of students. Unlike the manual process of searching, colleges make the students more stressed. Such a manual process of visiting each and every college makes students as well as parents uncomfortable. So to make it an easy process we use this web application.

1. **PROBLEM DEFINITION**

Without this application, there are many issues and wastage of sources like fuel, energy, time, etc. By keeping in mind about all these problems we created this application to make the task simpler in very few steps.

1. **LIMITATIONS OF PROJECT**

Although my college finder brings a lot of benefits to the users. There are few limitations such as data in the database that needs to be updated manually by the admin. Input provided by the user can't be modified/changed. Specifically usage of primary keys the data in the database will not allow repetition/duplication of data.

1. **OBJECTIVE OF PROJECT**

Security analysis demonstrates that our system is secure in terms of the data specified in the proposed security model. Finally, we implement a prototype of the proposed authorized duplicate check and conduct testbed experiments to evaluate the overhead of the prototype. In this project, aiming at the wastage of resources, we consider an application based on HTML, CSS and PHP.

**CHAPTER 2**

**LITERATURE SURVEY**

1. **INTRODUCTION**

A literature survey is the most important step in the software development process. Before developing the application it is necessary to search colleges manually i.e. student needs to go to the particular college and verify all the details manually. Nowadays we have various tools to create an application that makes everything simpler. Once the programmers start building the application the programmers need a lot of external support. This support can be obtained from senior programmers, from a book or from websites. Before building the system the above consideration is taken into account for developing the proposed system.

1. **EXISTING SYSTEM**

We have college selector applications that are similar but these applications doesn’t show the hostel facilities, no of departments available.

 Such architecture is practical and has attracted much attention from researchers.

 The data owners only outsourcing their data storage by utilizing limited databases.

 This process consumes more time

1. **PROPOSED SYSTEM**

In this project, we show the data in a web application by retrieving it from the database. Specifically, we present an advanced scheme to display all the details of colleges by using differential privilege keys. In this way, the users can access the details through this application without any time consumption. Accessing the data is very easy in this application.

**CHAPTER 3**

**ANALYSIS**

1. **REQUIREMENT SPECIFICATION** 
   * 1. **FUNCTIONAL REQUIREMENTS**

Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.

* + 1. **NON-FUNCTIONAL REQUIREMENTS**

Describe user-visible aspects of the system that are not directly related with the functional behaviour of the system. Non-Functional requirements include quantitative constraints, such as query (i.e. what is the query being executed).

**Usability**

Prioritize the important functions of the system based on usage patterns. Frequently this application should be uploaded based on changes in colleges for better usability. Be sure to create a requirement for this.

**Reliability**

Users have to trust the system, even after using it for a long time. Your goal should be a long MTBF (mean time between failures). Create a requirement that database created in the system will be retained for a number of years by changed the data by the admin. It’s a good idea to also include requirements that make it easier to monitor system performance.

1. **SOFTWARE REQUIREMENTS**

Coding Language : HTML, CSS, JSP.

IDE : Sublime Text 3

Server: XAAMP

Database : MYSQL

**CHAPTER 4**

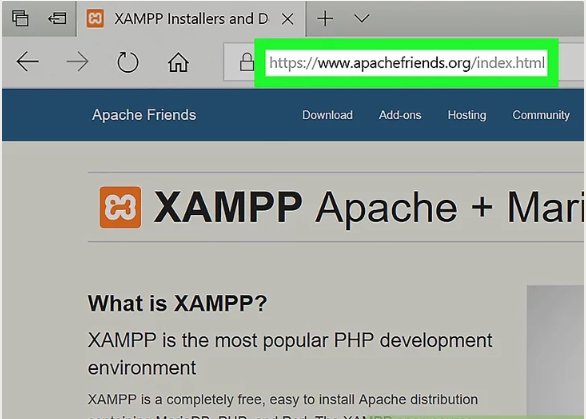
**SOFTWARE DESIGN**

1. **Xaamp software and drivers installation:**

This describes the installation of the XAAMP IDE Development software and drivers for the Windows Operating System. The images and description is based on installation under Windows XP, but the process should be similar for Vista and Windows 7.First we need to get the latest version of the XAAMP 64bit software this can be downloaded from the apachefriends website

1. **STEP 1: Installation Screen**

**Open the XAMPP website.** Go to <https://www.apachefriends.org/index.html> in your computer's web browser.

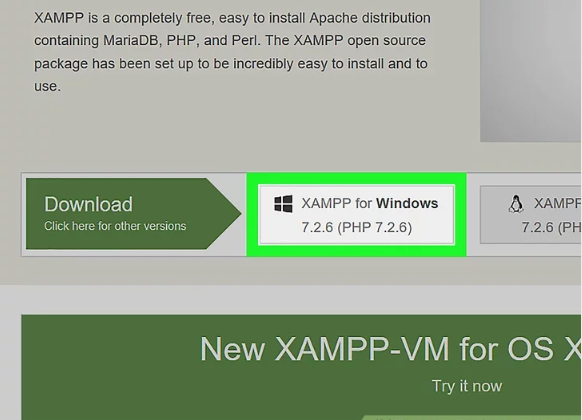


Screen 4.1. Web source

1. **STEP 2: Choose Installation Option**

**Click XAMPP for Windows.** It's a grey button near the bottom of the page.

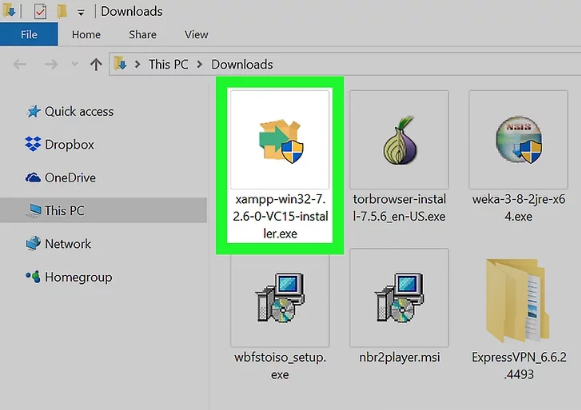
* Depending on your browser, you may first have to select a save location or verify the download

****

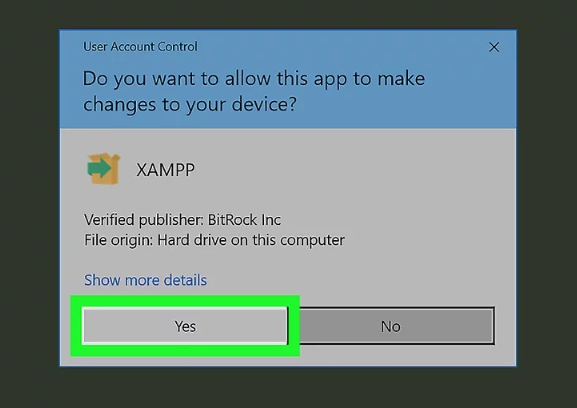
**Screen 4.2. Choose installation options**

1. **STEP3: Continue Installation**

**Double-click the downloaded file.** This file should be named something like **xampp-win32-7.2.4-0-VC15-installer**, and you'll find it in the default downloads location (e.g., the "Downloads" folder or the desktop).

****

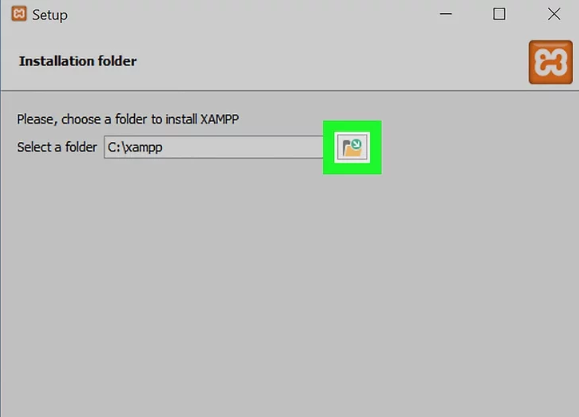
Screen 4.3. Setup.exe file



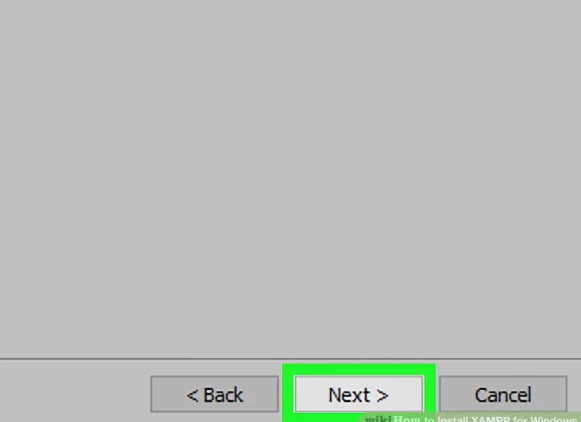
Screen 4.4. Run the setup

1. **STEP 4: Installation Finished**

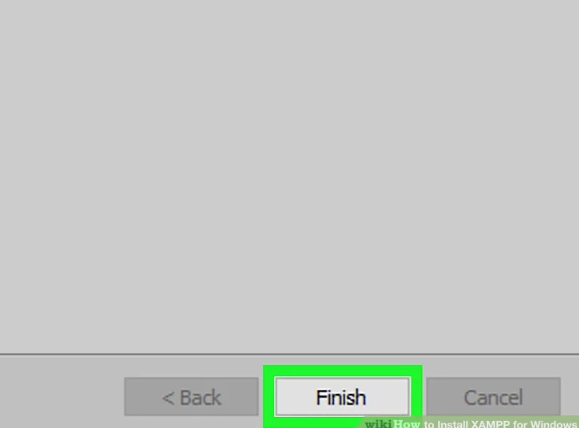
**Select an installation location.** Click the folder-shaped icon to the right of the current installation destination, then click a folder on your computer and click on next.



Screen 4.5. set the path

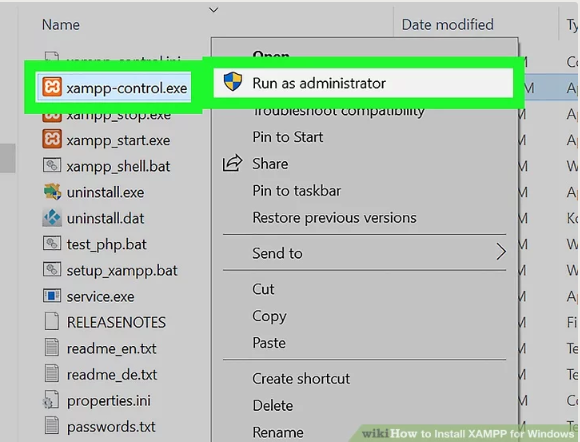
****

Screen 4.6. Click on next

****

Screen 4.7. Installation Finished

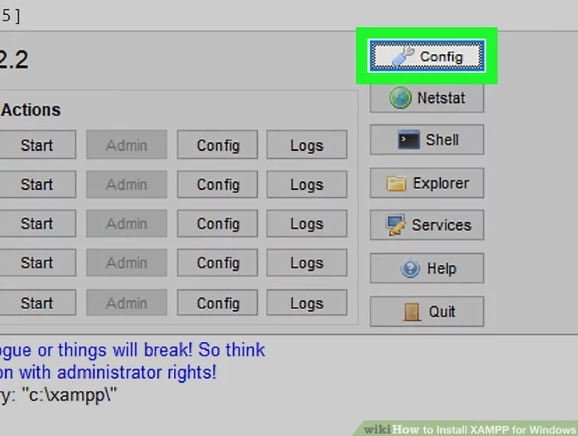
1. **STEP 5: Device Manager**

The computer communicates with the XAAMP server via ports. 

Screen 4.8. Run as Administrator

1. **STEP 6: Config Serial Ports**

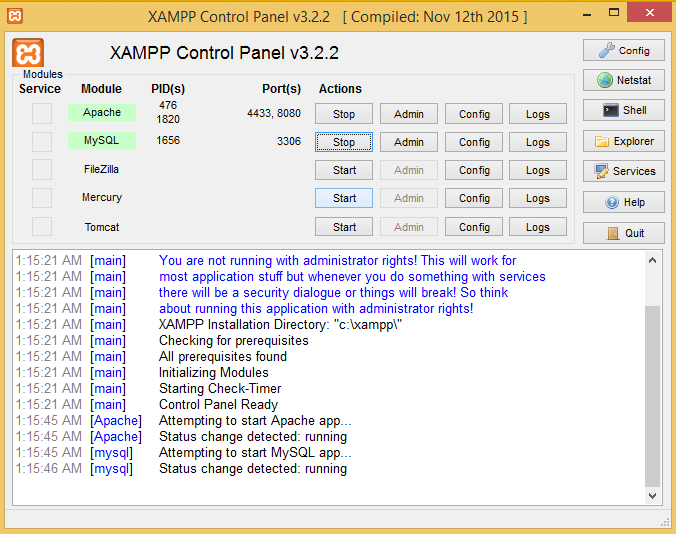
Next click on Config to set the port numbers



Screen 4.9. Serial Ports

1. **STEP 7: XAAMP Main Screen**

Finally we get the XAAMP control .exe file and we go to main screen. Then we start Apache and MYSQL.



Screen 4.10. XAAMP Main Screen

**CHAPTER 5**

**TECHNOLOGY USED**

1. **INTRODUCTION**

The purpose of the design phase is to create the application for the given databases. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blue print or plan for the solution, and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

1. **INPUT DESIGN AND OUTPUT DESIGN**
2. **INPUT DESIGN**

The input design is the link between the web application and the user. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What college should be taken?

How the application should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

1. **OUTPUT DESIGN**

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

Find the colleges based on the given search options.

Retrive the data from the database.

Print the database values.

1. **SYSTEM ARCHETECTURE**

Fig 4.3.1 – System Architecture 11

1. **UML DIAGRAMS**

**Object Oriented Analysis**

An object-oriented system is composed of objects. The behaviour of the system is achieved through collaboration between these objects, and the state of the system is the combined state of all the objects in it. Object Oriented Analysis aims to model the problem domain, the problem we want to solve by developing an object-oriented (OO) System. The source of the analysis is a written requirement statement, and/or written use cases, UML diagrams can be used to illustrate the statements.

**Object Oriented Design**

Object-Oriented Design (OOD) is an activity where the designers are looking for logical solutions to solve a problem, using Objects Object-oriented design takes the conceptual model that is the result of object-oriented analysis, and adds implementation constraints imposed by the environment, the programming language and the chosen tools, as well as architectural assumptions chosen as basis of Design.

The concepts in the conceptual model are mapped to concrete classes, to abstract interfaces in APIs and to roles that the objects take in various situations. The interfaces and their implementation for stable concepts can be made available as reusable services. Concepts identified as unstable in object-oriented analysis will form basis for policy classes that make decisions, implement environment-specific or situation specific logic or algorithms.

UML is the international standard notation for object-oriented analysis and design. The Object Management Group defines it. The heart of object-oriented problem solving is the construction of the model. The model abstracts the essential details of underline problem from its usually complicated real world. Several modeling tools are wrapped under the heading of the UML, which stands for Unified Modeling Language.

 Use case diagram

 Class diagram

 Object diagram

 Sequence diagram

 Collaboration diagram

 State chart diagram

 Activity diagram

 Component diagram

 Deployment diagram

**Use Cases** - Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view. Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer etc., or another system like central database.

**Sequence diagram** - Is an interaction diagram that details how operations are carried out -- what messages are sent and when Sequence diagrams are organized according to time? The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

**Class Diagram** - A Class diagram gives an overview of a system by showing its classes and the relationships among them. Class diagrams are static -- they display what interacts but not what happens when they do interact. Our class diagram has three kinds of relationships.

**Association** - A relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes.

**Aggregation** - An association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole.

**Generalization** - An inheritance link indicating one class is a super class of the other. A generalization has a triangle pointing to the super class.

**Activity Diagram** - An activity diagram is essentially a fancy flowchart. Activity diagrams and state chart diagrams are related. While a state chart diagram focuses attention on an object undergoing a process (or on a process as an object), an activity diagram focuses on the flow of activities involved in a single process.

**Use Case Diagram**

Fig 4.4.1 – Use Case Diagram 14

**Class Diagram**

Fig 4.4.2 – Class Diagram 15

**Sequence Diagram**

Fig 4.4.3 – Sequence Diagram 16

**Activity Diagram**

Fig 4.4.4 – Activity Diagram 17

1. **CONCLUSION**

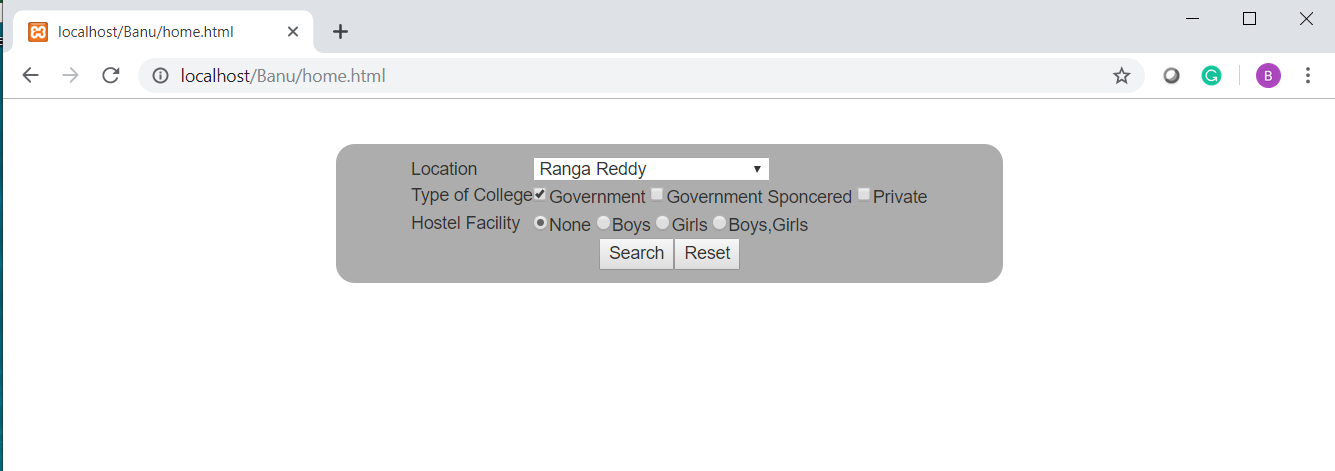
Web designing has reached a maturity that leads it into a productive phase. This means that most of the main issues with web designing have been addressed to a degree that these applications have become interesting for full commercial exploitation. This however does not mean that all the problems listed above have actually been solved, only that the data present in the database can be retrieved. Web designing is therefore still as much a research topic, as it is a market offering. For better confidentiality in web applications there are many open source software’s, in which we get different types of applications. Proposed system includes proof of data owner so it will help to implement better database queries in web designing.

**CHAPTER 6**

**IMPLEMENTATION AND RESULT**

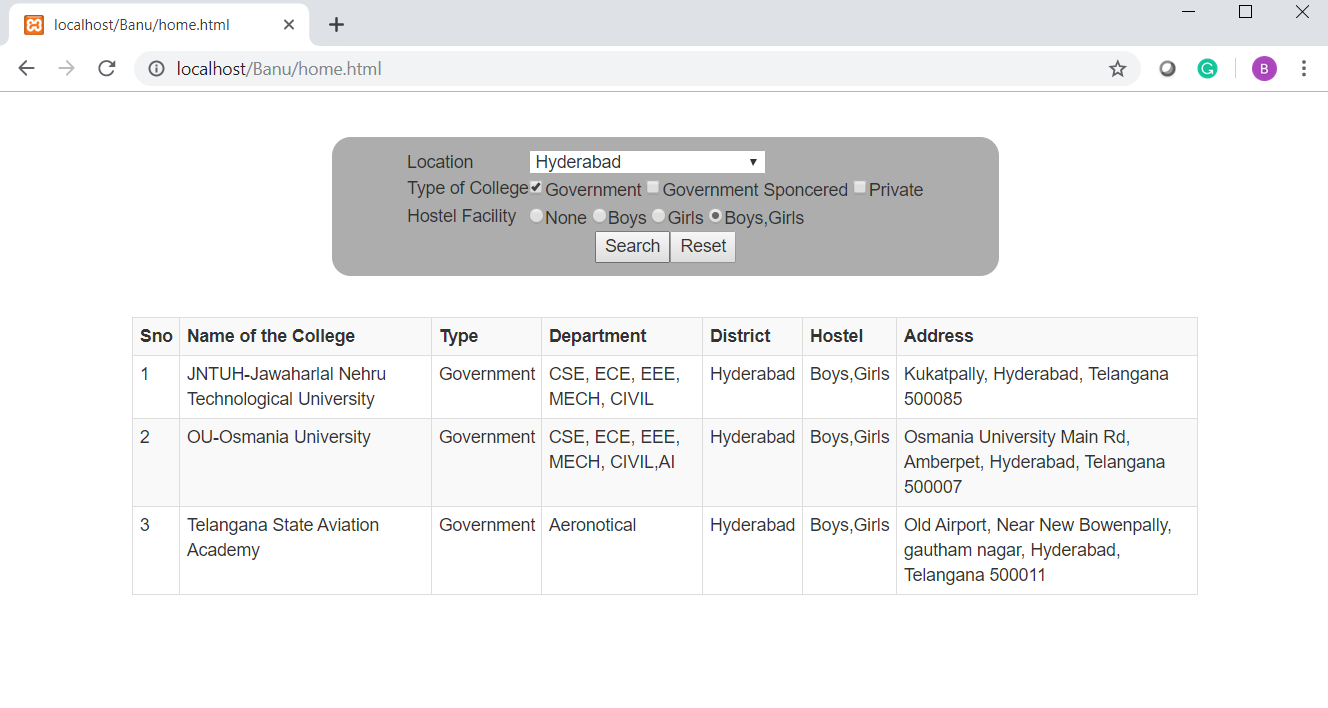
1. **IMPLEMENTATION**

We select the values into the application based on our interest and then retrieve the data from the database

****

1. **RESULT**

Once the data is retrieved we get the retrieved information of data from the MY\_SQL database server.

****

**CHAPTER 7**

**CONCLUSION**

This project has an aim to control the speed of any vehicles automatically in cities and also in restricted areas such schools, parks, hospitals and in speed limited areas etc. Nowadays in a fast moving world all the peoples are not have self-control. Such peoples are driving vehicles in a high speed. so the police are not able to monitor all those things. This paper provides a way for how to control the speed without harming others. Driver does not control anything during such places controls are taken automatically by the use of electronic system.

By using this system we can effectively control the speed of the vehicle according to the specified zones. This system can be implemented in any large vehicles.By using this system we can control the over speed and the rash driving of the drivers at restricted areas will be controlled. We can use this system in highly populated regions and thus we can decrease the effects of accidents.

**CHAPTER 8**

**FUTURE ENHANCEMENT**

The future enhancement of this project is though the above solution supports the differential privilege duplicate, it can be updated by using various fields by uploading the databases. Which can recover ﬁles falling into a known set. So we have to find an even better solution to it, by using this we can make a better application by saving time.

**CHAPTER 9**

**FILE NAME: “home.html”**

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

#one

{

top:5%;

width:50%;

left:25%;

background-color: #adadad;

padding: 10px 10px;

border-radius: 15px;

position: absolute;

}

#two

{

top:25%;

left:10%;

width:80%;

position: absolute;

}

</style>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

<script type="text/javascript">

loc();

function search()

{

var location=document.getElementById('location').value;

var type\_res=document.getElementsByClassName('type');

var hostel="";

var type="";

for(var i=0; i<type\_res.length;i++)

if(type\_res[i].checked)

type+=type\_res[i].value+",";

var hostel\_res=document.getElementsByClassName('hostel');

for(var i = 0; i <hostel\_res.length; i++)

if(hostel\_res[i].checked)

{

hostel=hostel\_res[i].value;

break;

}

// alert(location+":"+type+":"+hostel);

var str="location="+location+"&type="+type+"&hostel="+hostel;

if(window.XMLHttpRequest)

xmlhttp=new XMLHttpRequest();

else

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

xmlhttp.onreadystatechange=function()

{

if(this.readyState==4 && this.status==200)

document.getElementById('two').innerHTML=this.responseText;

};

xmlhttp.open("get","DataRetrive.php?"+str,true);

xmlhttp.send();

}

function loc()

{

if(window.XMLHttpRequest)

xmlhttp=new XMLHttpRequest();

else

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

xmlhttp.onreadystatechange=function()

{

if(this.readyState==4 && this.status==200)

document.getElementById('location').innerHTML=this.responseText;

};

xmlhttp.open("get","DataRetrive.php?empty=empty",true);

xmlhttp.send();

}

</script>

</head>

<body>

<center>

<div id="one">

<form >

<table>

<tr>

<td>Location</td>

<td>

<select style='width:60%;' id='location'>

<option>------------Select-----------</option>

<option>Hyderabad</option>

<option>Ranga Reddy</option>

<option>Sanga Reddy</option>

<option>---Other---</option>

</select>

</td>

</tr>

<tr>

<td>Type of College</td>

<td>

<input type="checkbox" class="type" checked value="Government">Government

<input type="checkbox" class="type" value="Government Sponcered">Government Sponcered

<input type="checkbox" class="type" value="Private">Private

</td>

</tr>

<tr>

<td>Hostel Facility</td>

<td>

<input type="radio" class="hostel" name="gender" value="None" checked>None

<input type="radio" class="hostel" name="gender" value="Boys">Boys

<input type="radio" class="hostel" name="gender" value="Girls">Girls

<input type="radio" class="hostel" name="gender" value="Boys,Girls">Boys,Girls

</td>

</tr>

<tr><td colspan="2"></td></tr>

<tr><td colspan="2"></td></tr>

<tr>

<td colspan="2" align="center"><input type="button" name="" value="Search" onclick="search()"><input type="reset" name="" value="Reset"></td>

</tr>

</table>

</form>

</div>

<div id="two" width="100%">

</div>

</center>

</body>

</html>

**FILE NAME: “dbconnect.php”**

<?php

$conn=mysqli\_connect("localhost","root","","search\_colleges");

?>

**FILE NAME: “dataretrive.php”**

<?php

include 'dbconnect.php';

if(isset($\_GET['empty']))

{

$options="";

$query="select DISTINCT district from colleges";

$clg\_res=mysqli\_query($conn,$query);

while($clg\_row=mysqli\_fetch\_assoc($clg\_res))

$options=$options."<option>".$clg\_row['district']."</option>";

echo $options;

}

if(isset($\_GET['type']))

{

$location=$\_GET['location'];

$type="";

$temp=explode(",",$\_GET['type']);

$hostel=$\_GET['hostel'];

foreach($temp as $x)

$type=$type."type='".$x."' or ";

$type=substr($type,0,-14);

$query="select \* from colleges where district='$location' and hostel='$hostel' and ".$type;

$clg\_res=mysqli\_query($conn,$query);

echo "<table class='table table-striped table-bordered table-hover table-condensed'>

<tr>

<th>Sno</th>

<th>Name of the College</th>

<th>Type</th>

<th>Department</th>

<th>District</th>

<th>Hostel</th>

<th>Address</th>

</tr>

";

$rows="";

for($i=1;$clg\_row=mysqli\_fetch\_assoc($clg\_res);$i++)

$rows=$rows."<tr><td>$i</td><td>".$clg\_row['college\_name']."</td><td>".$clg\_row['type']."</td><td>".$clg\_row['department']."</td><td>".$clg\_row['district']."</td><td>".$clg\_row['hostel']."</td><td>".$clg\_row['address']."</td></tr>";

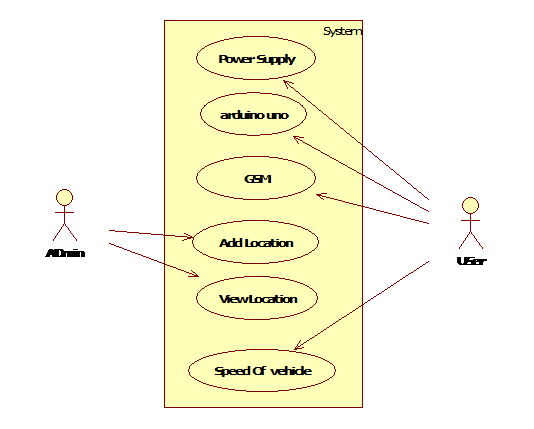
echo $rows."</table>";

}

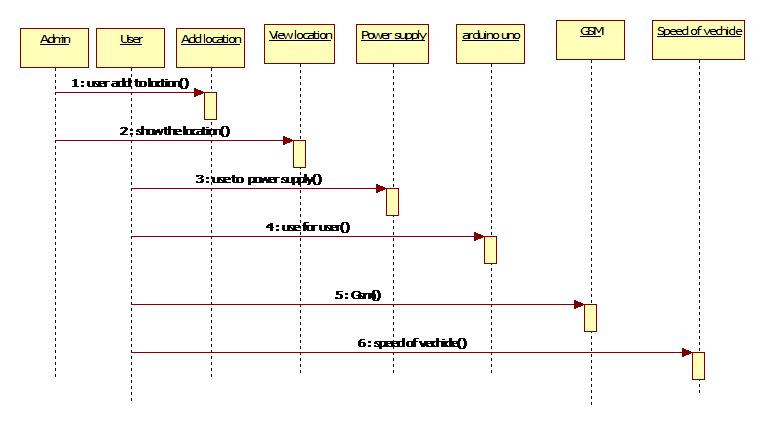
?>

**8.2. UML Diagram**

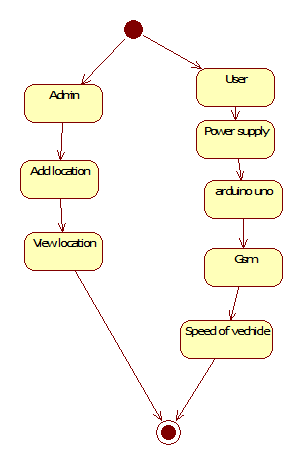
**8.2.1. Use Case Diagram**



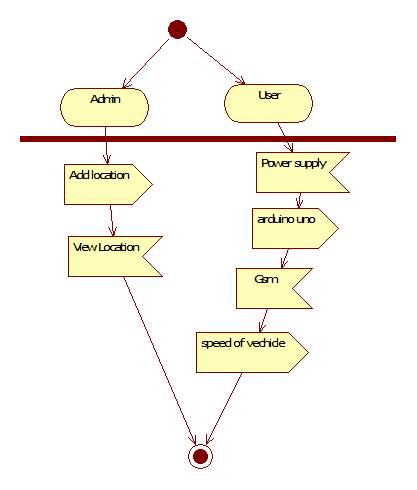
**8.2.2. Sequence Diagram**



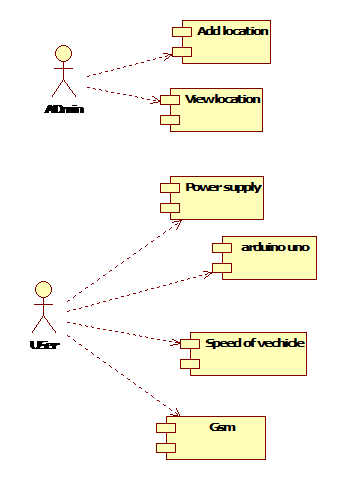
**8.2.3. State Chart Diagram**



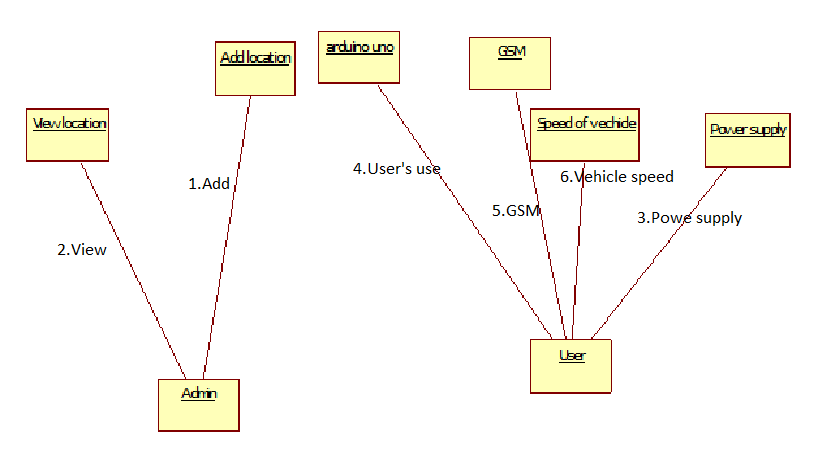
**8.2.4. Activity Diagram**



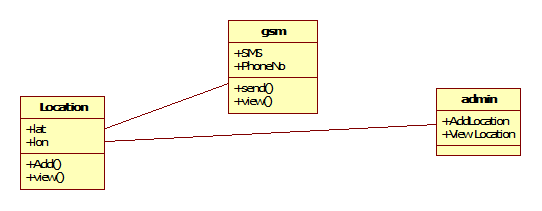
**8.2.5. Component Diagram**



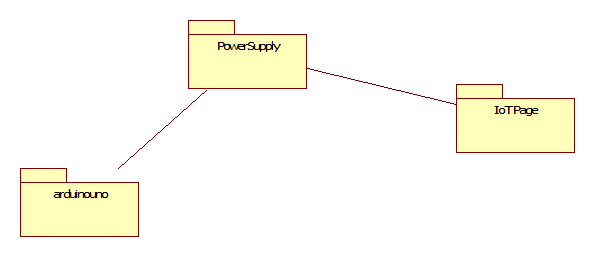
**8.2.6. Collaboration Diagram**



**8.2.7. Class Diagram**



**8.2.8. Deployment Diagram**

**8.3. Dataflow Diagram**

